

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : EDSJ - Ecole Doctorale Sciences Juridiques

Spécialité : Droit Privé

Unité de recherche : Centre de Recherches juridiques

Ecriture de contrats intelligents : essai de méthodologie en droit et en informatique

Writing of Smart legal contracts : proposition of a methodology in law and computer science

Présentée par :

Abdoulaye DIALLO

Direction de thèse :

Amélie FAVREAU

MAITRE DE CONFERENCES, Université Grenoble Alpes

Directrice de thèse

Rapporteurs :

Caroline LE GOFFIC

PROFESSEURE DES UNIVERSITES, Université de Lille

Davide FREY

DIRECTEUR DE RECHERCHE, UNIVERSITE DE RENNES

Thèse soutenue publiquement le **11 décembre 2023**, devant le jury composé de :

Amélie FAVREAU

MAITRESSE DE CONFERENCES, Université Grenoble Alpes

Directrice de thèse

Sihem AMER-YAHIA

DIRECTRICE DE RECHERCHE, Université Grenoble Alpes

Co-directrice de thèse

Caroline LE GOFFIC

PROFESSEURE DES UNIVERSITES, Université de Lille

Rapporteuse

Davide FREY

DIRECTEUR DE RECHERCHE, UNIVERSITE DE RENNES

Rapporteur

Noel DE PALMA

DIRECTEUR DE RECHERCHE, INRIA Grenoble

Examineur

Mouna MOUNCIF-MOUNGACHE

MAITRESSE DE CONFERENCES, Université de Saint-Etienne

Examinatrice



ECRITURE DE CONTRATS INTELLIGENTS – ESSAI DE METHODOLOGIE EN DROIT ET EN INFORMATIQUE

SOMMAIRE

Une table détaillée figure à la fin de l'ouvrage

PREMIERE PARTIE : DELIMITATION DU CONTRAT INTELLIGENT

Titre I – Le domaine substantiel du contrat intelligent

Chapitre I – Les processus sujets à une exécution dans la blockchain

Chapitre II – Les contrats sujets à une exécution dans la blockchain

Titre II – Le domaine formel du contrat intelligent

Chapitre I – Les différents *instrumentum* des contrats intelligents

Chapitre II – Sélection de *l'instrumentum* du contrat intelligent

SECONDE PARTIE : ELABORATION DU CONTRAT INTELLIGENT

Titre I – Rédaction des clauses indispensables du contrat intelligent

Chapitre I – Stipulations initiales du contrat

Chapitre II – Stipulations relatives à l'exécution du contrat

Chapitre II – Stipulations terminales du contrat

Titre II – Développement technique du contrat intelligent

Chapitre I - La détermination de la blockchain comme infrastructure d'exécution

Chapitre II – L'écriture des smart contracts

INTRODUCTION

*To be clear, at this point I quite regret adopting the term "smart contracts". I should have called them something more boring and technical, perhaps something like "persistent scripts"*¹. Vitalik Buterin

1. **Persistent scripts.** Ce sont en ces mots que le créateur d'Ethereum² exprimait son regret d'avoir adopté l'expression de *smart contract*. Il estimait qu'un terme plus austère comme celui de "scripts persistants" aurait évité de nombreuses confusions, et décrit plus justement ce que sont réellement des smart contracts : de simples programmes informatiques très résilients. Nous pensons cependant que c'est précisément l'usage de cette formule qui a suscité un tel intérêt de la part des juristes pour la *blockchain* et ses applications juridiques. Sans celle-ci, il est probable qu'ils ne se seraient pas autant intéressés à cette technologie, au point de réfléchir à l'adopter afin d'exécuter des contrats. Pour cette raison, nous saluons cette appellation, sans laquelle nous n'aurions peut-être jamais eu l'opportunité de mener ces présents travaux de recherches³.

I – Définition du sujet de recherche

2. **Définition de la *blockchain*.** Notre travail appelle qu'un soin particulier soit donné à la définition de certains termes dès l'introduction⁴. Nous commencerons par celui de *blockchain*⁵, avant les smart contracts, car la *blockchain* constitue l'infrastructure où ils évoluent. Pour la définir,

¹ Twitter. « vitalik.eth sur Twitter », 11 octobre 2018. <https://twitter.com/VitalikButerin/status/1051160932699770882>.

Pour être clair, à ce stade, je regrette vraiment d'avoir adopté le terme "smart contract". J'aurais dû les appeler par quelque chose de plus ennuyeux et technique, peut-être quelque chose comme "scripts persistants".

² Ethereum est le nom de la seconde cryptomonnaie et *blockchain* (que nous définirons) la plus populaire après Bitcoin. Contrairement à cette dernière, le fondateur derrière ce protocole n'est pas anonyme : il s'appelle Vitalik Buterin et est une figure très éminente de notre milieu.

³ Twitter. « michael rice, legal engineer sur Twitter », 2 novembre 2018. <https://twitter.com/michaelriceLE/status/1058477183872581637>.

I'm glad @VitalikButerin used the term smart contracts. Doing so captured the attention (and, for some, imagination) of the legal industry in a way that blockchains might not have otherwise.

⁴ Un lexique figurera en annexe pour compléter ce travail de définition.

⁵ A ce stade de notre thèse, nous considérerons les blockchains dans leur sens le plus ordinaire, c'est-à-dire celles dites publiques. Nous ferons la distinction avec les blockchains privées dans des développements plus tard.

nous nous appuyerons sur l'explication volontairement simple mais claire de Vitalik Buterin⁶ :

Une blockchain est un ordinateur magique dans lequel n'importe qui peut téléverser des programmes et les laisser s'exécuter automatiquement. Les états actuels et antérieurs de chaque programme sont toujours visibles publiquement, et elle offre une garantie très solide, sécurisée par des mécanismes cryptoéconomiques, que les programmes s'exécutant en son sein continueront à fonctionner exactement conformément aux spécifications du protocole⁷.

Ainsi, une *blockchain* est d'abord une infrastructure numérique capable d'accueillir des programmes⁸. Elle se caractérise, en plus, par sa transparence, permettant à quiconque disposant des outils appropriés de suivre librement l'exécution des programmes qu'elle abrite⁹. Enfin, cet « ordinateur magique » est décentralisé : ce n'est pas une entité en particulier, mais un large réseau d'individus qui travaillent et se synchronisent ensemble pour le faire fonctionner ; ce qui procure une garantie très solide que les opérations qui y sont prescrites seront bien réalisées.

Plus précisément, ce sont des mécanismes économiques incitatifs, cadrés par de la cryptographie, qui assurent le maintien décentralisé d'une *blockchain*. Lorsqu'un individu souhaite interagir avec elle, pour utiliser ou téléverser un programme, il soumet sa requête à des nœuds¹⁰ : les personnes membres

⁶ Nous avons fait le choix de ne pas assommer nos lecteurs par une description trop exhaustive de la *blockchain* et de son fonctionnement. Notre sujet ne nécessite pas de la comprendre au-delà de ses fonctionnalités essentielles qui seront présentées.

⁷ Vitalik Buterin. « Visions, Part 1: The Value of Blockchain Technology ». Ethereum Foundation Blog (blog), 13 avril 2015. <https://blog.ethereum.org/2015/04/13/visions-part-1-the-value-of-blockchain-technology>.

A blockchain is a magic computer that anyone can upload programs to and leave the programs to self-execute, where the current and all previous states of every program are always publicly visible, and which carries a very strong cryptoeconomically secured guarantee that programs running on the chain will continue to execute in exactly the way that the blockchain protocol specifies.

⁸ A tout le moins, la définition actuelle de la *blockchain*. A l'époque où Ethereum n'existait pas encore, il pourrait être argué que le terme de *blockchain* désignait simplement le procédé utilisé pour valider les blocs de transaction du protocole Bitcoin. Aujourd'hui, une *blockchain* est employée de façon métonymique pour désigner un registre distribué.

Larousse, Éditions. « Définitions : programme - Dictionnaire de français Larousse ». Consulté le 15 août 2023. <https://www.larousse.fr/dictionnaires/francais/programme/64207>.

Ensemble d'instructions et de données représentant un algorithme et susceptible d'être exécuté par un ordinateur.

⁹ Nous verrons par la suite qu'il existe des blockchainss dans lesquelles il est possible de limiter la visibilité des opérations.

¹⁰ « Running A Full Node - Bitcoin ». Consulté le 19 août 2023. <https://bitcoin.org/en/full-node>.

du réseau maintenant la *blockchain*. Un seul nœud alors est sélectionné, selon un procédé pouvant varier¹¹, pour vérifier que les règles de « l'ordinateur magique » ont bien été respectées et proposer sa modification subséquente à la communauté des nœuds¹². Si cette dernière le valide, la *blockchain* sera modifiée et le nœud sera rémunéré pour son travail par une monnaie numérique, une cryptomonnaie, spécialement générée afin de récompenser cette tâche¹³.

3. Définition des smart contracts. La première *blockchain*, *Bitcoin*, est une infrastructure garantie par ces mécanismes, mais qui permet presque uniquement le transfert, entre ses utilisateurs, de la cryptomonnaie *bitcoin*¹⁴. Face à ses limitations, des individus entreprirent de créer des *blockchainss* offrant davantage de fonctionnalités ; capables, par exemple, d'accueillir de véritables programmes pouvant contrôler les modalités de transfert des cryptomonnaies (exemple : si X évènement se passe, alors le transfert de Y somme peut avoir lieu). Ethereum fut la première d'entre

A full node is a program that fully validates transactions and blocks. Almost all full nodes also help the network by accepting transactions and blocks from other full nodes, validating those transactions and blocks, and then relaying them to further full nodes.

¹¹ Ce procédé fait référence au mécanisme de consensus. Un nœud peut être choisi au cours d'une « compétition » lors de laquelle il doit résoudre en premier une équation cryptographique (preuve de travail), ou encore à l'occasion d'une sorte de tirage au sort après avoir mis en jeu un certain type de cryptoactif (preuve d'enjeu).

¹² Valéria Faure-Muntia, Claude De Ganay et Ronan Le Gleut. « Comprendre les *blockchainss* : fonctionnement et enjeux de ces nouvelles technologies », p.33, 20 juin 2018. https://www.senat.fr/rap/r17-584/r17-584_mono.html.

Lorsqu'un nœud crée ou reçoit un nouveau bloc, il l'ajoute à sa copie du registre puis le transmet à ses nœuds pairs. Quand ceux-ci le reçoivent, ils vérifient que ce nouveau bloc est valide, c'est-à-dire qu'ils veillent en particulier à ce que la somme des transactions soit égale en entrée et en sortie. Si le bloc est valide, ils l'intègrent alors à leur registre et le transmettent à leur tour à leurs pairs.

¹³ Joelle Toledano et Lionel Janin. « Les enjeux des *blockchainss* ». Chapitre 3 – Des promesses à la chaîne – 1. Deux champs principaux, p.42. France Stratégie, juin 2018.

*De fait, les protocoles de consensus, qui sont aujourd'hui au cœur des *blockchainss* publiques, reposent tous sur des mécanismes d'incitation économique qui requièrent l'émission d'un actif numérique. Cet actif numérique permet d'inciter les différents acteurs à participer à la sécurisation du réseau – le protocole attribuant automatiquement un certain nombre d'actifs aux validateurs des nouveaux blocs. Ce fonctionnement fait des actifs numériques l'une des pierres angulaires des *blockchainss* publiques.*

¹⁴ Il pourra être avancé, surtout par les supporters invétérés de Bitcoin, que ce dernier protocole permet beaucoup plus d'usages. La réalité est qu'il était et est essentiellement utilisé à dessein de transférer la crypto-monnaie bitcoin et qu'il devient rapidement impraticable de l'utiliser pour d'autres buts que celui-ci.

Nakamoto, Satoshi. « Bitcoin: A Peer-to-Peer Electronic Cash System », 2008. bitcoin.org.

A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution.

elles à supporter cette évolution, et les programmes déployables en son sein furent nommés *smart contract*¹⁵.

Le choix de ce terme est lié aux travaux de Nick Szabo, une figure éminente dans le monde de la *blockchain*¹⁶. Dans les années 1990, ce dernier proposait dans un article l'idée de créer des logiciels cryptographiques garantissant l'exécution de véritables contrats, qu'il appelait *smart contract*¹⁷. Bien que son concept n'ait jamais été implémenté, les créateurs d'Ethereum choisirent de reprendre cette appellation pour désigner les programmes déployables dans leur *blockchain*. La caractéristique de contrôler les règles de délivrance de cryptomonnaies en s'appuyant sur la cryptographie, semble effectivement être la concrétisation des outils qu'avaient imaginés Nick Szabo¹⁸.

Aujourd'hui toutefois, il existe un consensus sur la signification des smart contracts : ils ne sont que de simples programmes, qui ont pour unique singularité de fonctionner à l'intérieur d'une *blockchain*. Ils héritent ainsi de ses propriétés : persistance et transparence. Avec leur aide, il est possible de créer des mécanismes de levées de fonds¹⁹ ou des monnaies privées comme les *stablecoin*, ces jetons à la

¹⁵ Primavera De Filippi et Aaron Wright. *Blockchain and the Law: The Rule of Code*. Smart contracts and Legal contracts. Harvard University Press, 2018. <https://www.jstor.org/stable/j.ctv2867sp>.

With the growing adoption of Bitcoin and other blockchain-based systems, there has been a renewed interest, and increased experimentation, in transforming legal agreements into code. Advanced blockchain-based protocols like Ethereum provide the necessary technology to implement some of the ideas described by Nick Szabo over twenty years ago.

¹⁶ Stefan Stankovic. « Who Is Nick Szabo, The Mysterious Blockchain Titan ». Unblock.Net (blog), 11 janvier 2018. <https://unblock.net/nick-szabo/>.

Nick Szabo is a living legend in both the cryptocurrency and cryptography worlds. Although he's not a household name, and if you're new to the crypto games, you probably would have never heard of him. However, Nick Szabo has godlike status amongst the sincere crypto enthusiasts. (...) Both a computer scientist and a legal scholar, he married his two central interests and, in 1996, gave birth to the concept of "smart contracts." Many years later, "smart contracts" became the central "feature" of the Ethereum blockchain protocol and gave rise to a whole new way of commerce on the Internet.

¹⁷ Nick Szabo. « Formalizing and Securing Relationships on Public Networks ». First Monday 2, n° 9 (1 septembre 1997). <https://doi.org/10.5210/fm.v2i9.548>.

Smart contracts combine protocols with user interfaces to formalize and secure relationships over computer networks. (...). By using cryptographic and other security mechanisms, we can secure many algorithmically specifiable relationships from breach by principals...

¹⁸ X (formerly Twitter). « vitalik.eth sur X », 11 octobre 2018. <https://twitter.com/VitalikButerin/status/1051161357104635906>.

I do think that persistent scripts controlling assets compete with the legal system on some margins, but so do locks on doors...

¹⁹ V., *supra*, §16

valeur stable qui ont suscité l'intérêt des Etats et des grandes entreprises²⁰. Ils rendent aussi possible la création de jetons non fongibles, les fameux *NFT*, qui ont, eux aussi, su capturer intensément mais quoique brièvement l'attention du monde entier²¹. Ils peuvent encore être utilisés pour enregistrer des informations dans la *blockchain*, créer des applications décentralisées, des jeux vidéo...

4. Les contrats exécutés à l'aide de *smart contract*. Comme tout programme, ils peuvent aussi être utilisés dans un contexte contractuel : afin d'automatiser l'exécution de contrats²². Il est alors survenu une difficulté sémantique pour distinguer les véritables contrats exécutés à l'aide de *smart contract*, des smart contracts en eux-mêmes. Si ces derniers ne désignent que des programmes fonctionnant dans une *blockchain*, quelle expression doit être utilisée pour nommer ce qu'ils évoquent réellement dans l'imaginaire collectif : à savoir des véritables contrats automatisés dans la *blockchain* ? La communauté juridique anglo-saxonne a, en première, avancé le terme de *smart legal contract*, pour désigner un accord contractuel dont les processus sont exécutés à l'aide de *smart contract*²³. Sa traduction française donne la très inélégante expression de « contrat juridique intelligent »²⁴. Les américains ont proposé le terme de « contrat computationnel », premièrement employé dans un article de Harry Surden²⁵, afin de désigner tout contrat automatisé par des programmes informatiques : ce qui inclut les contrats exécutés par des scripts dans la *blockchain*, et ceux exécutés par des scripts « ordinaires » (sur des serveurs centralisés). Récemment, le terme de contrat ricardien a pu être convoqué pour nommer ces conventions, en référence aux travaux de Ian Grigg²⁶. Dans sa thèse, proposant une étude en droit des contrats des smart contracts, Madame

²⁰ V., *infra*, §54

²¹ V., *infra*, §57

²² Ludovic Mounoussamy. « Le Smart contract, acte ou hack juridique ? », Petites affiches, n° 037 (20 février 2020).

Le smart contract n'est donc pas à proprement parler un contrat mais au sens juridique un accessoire au contrat principal. En effet, il ne contient pas les éléments substantiels à sa validité mais constitue un mode d'exécution de celui-ci.

²³ Nous attribuons l'origine du terme à Peter Hunn, un entrepreneur britannique de *legaltech* qui a créé l'entreprise *Clause* (dont nous aurons l'occasion de parler) qui proposait la création de tels contrats.

Allen Jason, et Peter Hunn, éd. *Smart Legal Contracts: Computable Law in Theory and Practice*. Oxford, New York: Oxford University Press, 2022.

²⁴ Fabien Gillioz. « Du contrat intelligent au contrat juridique intelligent », *Dalloz IP/IT*, 2019, 16.

La question se pose donc de savoir comment utiliser la technologie de la blockchain pour compléter ou remplacer les contrats juridiques existants pour devenir un contrat juridique intelligent (« smart legal contract »).

²⁵ Harry Surden. « Computable Contracts ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 2012. <https://papers.ssrn.com/abstract=2216866>.

²⁶ Une autre entreprise spécialisée dans la création de contrats exécutés par des smart contracts, *OpenLaw* (sur laquelle

Leveneur suggère celui de « contrats informatisables », visant spécifiquement les contrats composés de *smart contract* et de stipulations en langage naturel²⁷. Force est de constater qu'en dépit du consensus sur la nécessité de distinguer sémantiquement ces contrats des smart contracts, il n'en existe toujours pas sur le terme le plus convenable à choisir. L'exercice est d'autant plus difficile qu'il existe des qualifications pouvant séduire, comme celle de contrats électroniques, mais que le droit français réserve déjà pour désigner les conventions qui n'ont pour caractéristique que d'être conclues par voie électronique²⁸ ; ce qui recouvre imparfaitement les contrats dont nous discutons.

5. Choix du terme de contrat intelligent. Dans cette étude, nous choisissons l'expression de contrats intelligents pour nommer les contrats exécutés à l'aide de *smart contract*. Bien qu'elle soit la traduction littérale de *smart contract*, nous estimons que le terme anglais s'est solidement ancré dans notre vocabulaire pour désigner clairement des programmes informatiques déployés dans une *blockchain*. Tandis que celui de contrat intelligent, tout en évoquant les smart contracts, peut être utilisé de façon distinctive afin de désigner spécifiquement des véritables contrats qui utilisent ces smart contracts pour leur exécution.

6. Distinction entre exécution et automatisation. Nous distinguons l'exécution et l'automatisation par *smart contract*. L'automatisation signifie la suppression de l'intervention humaine dans l'accomplissement d'une tâche, afin qu'elle soit réalisée par un outil²⁹. Dans notre

nous aurons également l'occasion de revenir), utilise cette expression pour désigner ces derniers contrats.

OpenLaw. « The Smart contract Stack ». Medium (blog), 24 septembre 2019.
<https://medium.com/@OpenLawOfficial/the-smart-contract-stack-5566ea368a74>.

OpenLaw brings the vision of ricardian contracts to life, providing the tools to fulfill the initial vision that Ian Grigg set out several decades ago (...). When combined with Ethereum-based smart contracts and third-party oracle providers, like ChainLink, we truly have dynamic legal agreements that are understandable and executable, but responsive to any present, past or future real-world events.

²⁷ Claire Leveneur. « Les smart contracts : étude de droit des contrats à l'aune de la blockchain ». §742, p. 522. Thèse de doctorat, Université Paris-Panthéon-Assas, 2022. <https://www.theses.fr/2022ASSA0063>.

Ainsi, les smart contracts peuvent être soit une simple modalité d'exécution de certaines clauses d'un contrat préexistant, soit un véritable contrat à part entière. Dans ce second cas, le contrat, appelé informatisable, réunit dans un support numérique unique tant le negotium que son mode d'exécution, ce qui constitue une véritable innovation : l'instrument est alors à la fois qualifié de contrat et de paiement, lorsque les obligations informatisables qu'il contient sont exécutées conformément à ce qui était prévu

²⁸ Articles 1369-1 et suivants du code civil.

²⁹ Larousse, Éditions. « Définitions : automatisation - Dictionnaire de français Larousse ». Consulté le 15 août 2023.
<https://www.larousse.fr/dictionnaires/francais/automatisation/6753>.

Exécution totale ou partielle de tâches techniques par des machines fonctionnant sans intervention humaine.

contexte, un contrat automatisé serait donc un contrat dont tout ou partie des obligations qui incombent aux parties seraient réalisées par des programmes. Prenons l'exemple d'une police d'assurance stipulant le versement d'une indemnité en cas de sinistre spécifique. Si ce contrat était automatisé, la détection du sinistre et le versement de l'indemnité seraient entièrement gérés par des programmes : un logiciel informerait de l'incident et un autre déclencherait le paiement³⁰. À l'inverse, s'il n'était pas automatisé, le bénéficiaire devrait lui-même informer et prouver le sinistre à l'assureur, qui ensuite effectuerait le paiement.

Or, nous verrons que des smart contracts peuvent être utilisés dans des contrats, sans pour autant qu'ils aient la tâche de réaliser à la place des parties leurs obligations. Ces conventions ne sont alors pas automatisées, au sens propre du terme, mais restent exécutées à l'aide de *smart contract*. Dans un contrat de vente d'un bien numérique, si le bien est matérialisable dans une *blockchain*³¹, les parties peuvent utiliser un *smart contract* afin de le séquestrer et ainsi sécuriser la transaction³². Le vendeur placerait le bien numérique dans un *smart contract*, qui le tiendrait en séquestre le temps que l'acheteur verse le prix à ce même *smart contract*. Lorsque la somme serait déposée, l'acheteur pourrait récupérer le bien numérique et le vendeur le prix d'achat. Bien que les opérations se soient déroulées numériquement, à l'aide d'un *smart contract*, elles n'ont nécessité que des actions quasi-manuelles des parties : il n'y a pas eu de versement ni de livraison automatique des biens et de la somme à payer. Nous définissons donc un contrat intelligent comme un contrat exécuté à l'aide de *smart contract*, ce qui inclut les contrats automatisés par des smart contracts, sans y être limités.

³⁰ Luc Mayaux. « L'assurance 3.0 », Revue générale du droit des assurances, n° 12, p.1.

Quant à l'automaticité dans le règlement des sinistres, elle jette aux oubliettes, pêle-mêle, la déclaration du sinistre (que l'article L. 113-2, 4°, du Code des assurances érigeait pourtant au rang d'obligation : l'imparfait est de rigueur), l'expertise, et naturellement le principe indemnitaire que l'assurance paramétrique avait déjà mis à mal.

³¹ V., *supra*, §47

³² Ici le mot séquestre est employé comme le mot anglais *escrow* (qui est sa traduction en français). Son sens est légèrement différent du séquestre visé aux articles 1955 et suivants du code civil :

« Escrow », 22 mars 2023. <https://dictionary.cambridge.org/fr/dictionnaire/anglais/escrow>.

an agreement between two people or organizations in which money or property is kept by a third person or organization until a particular condition is met.

II – Cadres de la recherche

7. **Inscription de la recherche dans un cadre de référence théorique.** Le sujet de recherche défini, nous pouvons procéder à plusieurs mises en contextes. Tout d'abord, le cadre historique du contrat intelligent permet de mettre en lumière le fait que la démarche d'élaboration de ces contrats est l'aboutissement d'un effort ancien (a). De plus, le cadre législatif est un élément essentiel pour établir l'environnement légal dans lequel évoluent les contrats intelligents (b). Enfin le cadre doctrinal donne une assise scientifique à notre sujet de recherche (c).

a) Cadre historique

8. **La formalisation, retour sur une méthodologie ancienne.** Lorsqu'on recourt à des programmes informatiques pour exécuter un contrat, et que celui-ci est préalablement matérialisé par un texte rédigé en langage naturel, il est exercé une sorte de traduction des stipulations juridiques vers un langage informatique. Par exemple, si un contrat de prestation de service stipule le versement d'une somme d'argent après l'accomplissement d'une tâche par un prestataire, l'exécution de cette obligation par un programme nécessitera de la retranscrire en code informatique. Cette démarche s'appelle la *formalisation*³³ en informatique, et elle constitue un effort ancien de la communauté juridique dont l'écriture de contrats intelligents est l'itération la plus récente.

9. **Layman E. Allen.** L'universitaire américain Layman E. Allen figure parmi les tous premiers juristes à avoir proposé une méthode pour formaliser le langage juridique³⁴. Il choisit d'utiliser la logique symbolique, l'idiome pour exprimer les expressions mathématiques³⁵, afin d'« écrire » le droit ; cherchant ainsi à éliminer les ambiguïtés non voulues dans les contrats³⁶, tout

³³ Frédéric Muhindo Muyisa. « Formalisation informatique de la comptabilité hospitalière en RDC. Cas des Centres de santé ». II- Nouvelles technologies et informatique -- II.4 Formalisation. Université adventiste de Lukanga, 2008.

Formaliser un problème, c'est le décomposer en opérations très élémentaires en vue de faciliter leur automatisation (utilisation de la machine pour la réalisation d'un programme de travail, l'intervention humaine étant moins réduite.) La formalisation informatique d'un problème, c'est donc sa réduction à des structures formelles (élémentaires, indécomposables) en utilisant des ordinateurs.

³⁴ Allen, Layman E. « Symbolic Logic: A Razor-Edged Tool for Drafting and Interpreting Legal Documents ». The Yale Law Journal 66, n° 6 (mai 1957): 833. <https://doi.org/10.2307/794073>.

³⁵ Elle est un archétype du langage formel et est si souvent utilisée pour écrire des expressions mathématiques, qu'on l'appelle aussi le langage mathématique. Exemple : $D(x, y) = \{ k \mid \exists j : y[j] < x[k] \} 1$.

³⁶ Allen, Layman E. « Symbolic Logic: A Razor-Edged Tool for Drafting and Interpreting Legal Documents ». The Yale Law Journal 66, n° 6 (mai 1957): 833. <https://doi.org/10.2307/794073>.

en facilitant leur rédaction et analyse³⁷. Ses travaux aboutirent sur la proposition d'un langage qui combinait la sémantique juridique avec la logique symbolique.

Par exemple, pour exprimer une implication en logique symbolique, il est fait recours au signe : ____.

Dans le langage de Allen, les parties pouvaient décrire leurs obligations de la sorte:

- soit $P = \text{Mme. Y paye 15 euros à M. X}$
- et $Q = \text{M. X envoie le bien acheté.}$

Pour dire, *si Mme. Y paye le bien de M. X, alors ce dernier doit le lui envoyer*, il pouvait être simplement écrit :

P

$\neg Q$

10. Les contrats ricardiens et traduits dans le langage de programmation Haskell³⁸.

En 1996, un financier-cryptographe du nom de Ian Grigg inventait les contrats ricardiens : des contrats formalisés³⁹ de sorte qu'ils soient aussi aisément lisibles par des logiciels que par des humains⁴⁰. L'un des buts de cette démarche était ainsi de faciliter l'analyse automatique des conventions.⁴¹ Assez rapidement, des informaticiens remarquèrent que les contrats financiers se

In this Article it is suggested that a new approach to drafting, using certain elementary notions of symbolic logic, can go a long way towards eliminating such inadvertent ambiguity.

³⁷ Allen, Layman E. « Symbolic Logic: A Razor-Edged Tool for Drafting and Interpreting Legal Documents ». The Yale Law Journal 66, n° 6 (mai 1957): 833. <https://doi.org/10.2307/794073>.

In addition, it can be a valuable aid in moving towards a more comprehensive and systematic method of interpretation, as well as drafting.

³⁸ Haskell est un langage de programmation de type fonctionnel datant des années 1990.

³⁹ A noter que la formalisation implémentée par Grigg est "douce" : le contrat n'est pas réécrit dans un langage de programmation mais représenté dans un format interprétable par un programme (JSON, XML, etc).

⁴⁰ Ian Grigg. « The Ricardian Contract », 2004. https://iang.org/papers/ricardian_contract.html#ref_11.

Our innovation is to express an issued instrument as a contract, and to link that contract into every aspect of the payment system. By this process, a document of some broad utility (readable by user and program) is drafted and digitally signed by the issuer of the instrument. This document, the Ricardian Contract, forms the basis for understanding an issue and every transaction within that issue.

⁴¹ La question de l'opportunité de formaliser ou laisser en langage naturel un document juridique pour l'analyser par des programmes fait l'objet d'un débat nourri. Certains pensent que les outils de traitement du langage naturel (autrement dit l'intelligence artificielle) suffisent pour bien analyser les contrats tandis que d'autres estiment qu'il serait opportun de formaliser le texte juridique dans un langage machine afin d'améliorer sa lecture par des programmes.

prêtaient bien à une formalisation : sans doute en raison de la nature très objective des opérations qu'ils mettent en œuvre. Alors l'exercice se poursuivait, notamment à travers des expérimentations comme celles menées par Jean-Marc Eber⁴². Ce dernier utilisa le langage de programmation *Haskell* pour décrire précisément des contrats d'option⁴³ ou des contrats à terme⁴⁴. L'expérience fut si réussie qu'elle donna naissance à une entreprise spécialisée dans la gestion de contrats de ce type, encore prospère aujourd'hui⁴⁵.

11. La formalisation des contrats par Nick Szabo. Quelque temps après avoir introduit le concept de *smart contract*⁴⁶, Nick Szabo poursuit son œuvre en travaillant sur la formalisation des contrats dans un langage de programmation. Dans un article paru en 2002⁴⁷, il proposait un langage formel afin de *spécifier de manière aussi claire, complète et succincte que possible, des*

Harry Surden. « Computable Contracts ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 2012. <https://papers.ssrn.com/abstract=2216866>.

(...) *parties who want computerized analysis of their contractual obligations would thus have to await advances technological processing of natural language [OR] to reorient(...) toward a form more amenable to computer processing.*

Artificial Lawyer. « Machine-Readable Contracts – A New Paradigm For Legal Documentation ». Artificial Lawyer (blog), 28 août 2019. <https://www.artificiallawyer.com/2019/08/28/machine-readable-contracts-a-new-paradigm-for-legal-documentation/>.

If we really want to change this process and solve unstructured data, the answer can't be to add more stages. It must be to remove them. Instead of creating contracts built from unstructured data, getting them to signature and then spending time, money and processing power on trying to structure it, what if contracts were agreed and managed throughout the lifecycle in a structured format ?

⁴² Jones, SL Peyton, J.-M. Eber, J. Seward, et Simon Peyton Jones. « Composing Contracts: An Adventure in Financial Engineering », 1 septembre 2000. <https://www.microsoft.com/en-us/research/publication/composing-contracts-an-adventure-in-financial-engineering/>.

⁴³ Franck Auckenthaler. « Fasc. 2050 : Instruments financiers à terme ou contrats financiers », §28, JurisClasseur Sociétés Traité, 1 août 2020.

Une option est un contrat par lequel une partie, dite « acheteur » de l'option, acquiert contre le paiement d'une prime à l'autre partie, dite « vendeur » de l'option, le droit, mais non l'obligation, d'acheter ou de vendre une quantité déterminée d'un actif sous-jacent à un prix d'exercice donné, pendant une période ou à une ou plusieurs dates déterminées.

⁴⁴ Ibid, §23

Un contrat à terme ferme, dans son expression la plus simple, ressemble à une vente à terme. Il y a, en effet, un accord sur la chose, le sous-jacent, et son prix, et les obligations de payer et de livrer sont affectées d'un terme.

⁴⁵ <https://www.lexifi.com/company/about/>

⁴⁶ Szabo, Nick. « Formalizing and Securing Relationships on Public Networks ». First Monday 2, n° 9 (1 septembre 1997). <https://doi.org/10.5210/fm.v2i9.548>.

⁴⁷ Szabo, Nick. « A Formal Language for Analyzing Contracts | Satoshi Nakamoto Institute », 2002. <https://nakamotoinstitute.org/contract-language/>.

contrats communs ou des clauses contractuelles.⁴⁸ Parmi les objectifs de cette démarche étaient la possibilité de créer des conventions pouvant être analysées automatiquement afin qu'on y détecte *des défauts de logique, de calendrier, des opportunités pour les parties de rompre le contrat*,⁴⁹ et de les rendre exécutoires⁵⁰.

12. Les échanges de données informatisées (EDI). Selon ce dernier toutefois, c'est la technologie des échanges de données informatisées qui constituait un des tous premiers cas de formalisation contractuelle véritablement réussis⁵¹. Son origine peut être remontée au blocus de Berlin en 1948, où pour fluidifier l'envoi des denrées alimentaires par les alliés vers l'Allemagne de l'Ouest, le sergent américain Edward Guilbert créa une sorte de langage formel qui permettait d'exprimer efficacement les commandes de vivres sur fax, télex et téléphone⁵². Le succès du procédé fut tel, qu'il ne tarda pas à être adopté par des entreprises provenant de secteurs aussi variés que l'agro-alimentaire ou l'automobile⁵³.

Aujourd'hui, les EDI désignent une technologie qui permet de faire communiquer des machines entre

⁴⁸ Szabo, Nick. « A Formal Language for Analyzing Contracts | Satoshi Nakamoto Institute », 2002.
<https://nakamotoinstitute.org/contract-language/>.

The main purpose of this language is to, as unambiguously and completely and succinctly as possible, specify common contracts or contractual terms

⁴⁹ Szabo, Nick. « A Formal Language for Analyzing Contracts | Satoshi Nakamoto Institute », 2002.
<https://nakamotoinstitute.org/contract-language/>.

Analyze formally specified contracts for flaws in logic, scheduling, opportunities for parties to breach the contract, and conflicts with other contracts one is already committed to.

⁵⁰ Ibid.

Translate the formal contracts into an existing programming language such as E, and/or cryptographic protocols, for partially self-enforcing and protected execution as smart contracts.

⁵¹ Szabo, Nick. « Smart contracts: Building Blocks for Digital Markets », 1996.
https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html.

The field of Electronic Data Interchange (EDI) (...) can be viewed as a primitive fore runner to smart contracts.

⁵² Hayes, Frank. « The Story So Far ». Computerworld, 17 juin 2002.
<https://www.computerworld.com/article/2576616/the-story-so-far.html>

US Army Master Sergeant Edward Guilbert developed a manifest system that could be transmitted by telex, radio teletype, or telephone.

⁵³ De Filippi, Primavera, et Aaron Wright. Blockchain and the Law: The Rule of Code. Harvard University Press, 2018.

The shipping, food, grocery and automobile industries routinely rely on EDI systems...

elles à l'aide de messages standardisés⁵⁴ : deux systèmes, appartenant à deux entités différentes, peuvent ainsi s'échanger automatiquement des documents commerciaux (devis, commandes, factures, connaissances...) de manière instantanée, peu coûteuse et sans risques d'erreurs⁵⁵. Prenons l'exemple d'une entreprise A qui est fournisseuse de marchandises à l'entreprise B. B fait très régulièrement des commandes de marchandises auprès de A. Pour économiser le coût d'un travail humain, elle configure un programme qui génère directement un devis de marchandises lorsque les stocks exigent une nouvelle commande. Ce devis est traduit dans un format de données spécifique et transmis directement vers le système d'information de A, grâce aux EDI. La réception du devis dans le système de A, lui permettra de déclencher d'autres actions automatiques telles que la production de ladite marchandise et/ou son envoi vers l'entrepôt de B. Cette communication, via les EDI, de documents commerciaux donne ainsi la possibilité aux parties d'automatiser l'ensemble de leur processus contractuel.

13. La formalisation juridique de nos jours. L'effort de formalisation de la prose juridique se poursuit encore aujourd'hui, avec toujours les mêmes desseins. Dans le domaine particulier de la formalisation contractuelle, on citera des *startup* comme *Legalese*⁵⁶ qui utilise un langage de programmation spécifique⁵⁷ pour rédiger des accords juridiques. Sur un terrain plus prospectif, on pourra noter des initiatives ambitieuses comme celles de *Lexon*⁵⁸, qui a pour objectif de proposer un langage à la fois « naturel » et « formalisé » afin d'écrire des contrats intelligents de manière accessible.

⁵⁴ « Ce Qu'est EDI (Échange de Données Informatisé)? | EDI Pour Tous », 20 septembre 2019.
<https://www.edipourtous.fr/ce-qu-est-l-edi/>.

Plusieurs normes en matière d'EDI sont actuellement en vigueur, notamment ANSI, EDIFACT, TRADACOMS et XML.

⁵⁵ Ibid.

(...) les entreprises bénéficient d'avantages significatifs tels que la réduction des coûts, l'amélioration de la vitesse de traitement, la diminution des erreurs et l'amélioration des relations avec leurs partenaires commerciaux.

⁵⁶ <https://legalese.com/>

We start by creating and implementing L4, a domain-specific language (DSL) for legal that is specifically designed to capture the particularities of law, its semantics, deontics, and logic. Just as Cadence's Verilog allows programmers to draft circuits, Legalese's L4 will allow programmers to express law, the applicable laws, but also the regulations, business processes, business logic, and other quasi-legal rules — all the rules and logic that you want your documents to adhere to.

⁵⁷ Par langages spécifiques nous entendons des langages de programmation *ad hoc* : créés spécifiquement pour un seul projet.

⁵⁸ <http://lexon.org/>

Lexon is a plain-text programming language for digital contracts and law. It is the perfect language for blockchain smart contracts. Its readability makes it the interface between trustless tech and the legal system.

b) Cadre législatif

14. Le cadre législatif en construction des contrats intelligents. Notre étude se situe en droit français où aucune disposition nationale aborde nommément les smart contracts. Cependant plusieurs textes intéressent la *blockchain* et les cryptoactifs⁵⁹ ; et une proposition de réglementation européenne semble viser directement le sujet de notre recherche.

15. Ordonnance « *blockchain* ». L'ordonnance n°2016-520 du 28 avril 2016 relative aux bons de caisse figure parmi les toutes premières d'entre elles. Elle instaure à l'article L.223-12 du code monétaire et financier la possibilité d'émettre et de céder des minibons⁶⁰ dans une *blockchain*, qu'elle nomme par ailleurs un dispositif d'enregistrement électronique partagé (DEEP)⁶¹. Cette ordonnance fut suivie d'une autre, n°2017-1674 du 8 décembre 2017, qui étend aux titres financiers non cotés, c'est-à-dire ceux non admis aux opérations d'un dépositaire central de titres, la possibilité d'utilisation d'un DEEP pour les représenter et transmettre⁶². Un an plus tard, un décret n°2018-1226 vint mettre en application les dispositions de ces deux ordonnances. Il précise les qualités que doit présenter un DEEP pour accueillir la représentation et transmission des minibons et titres non cotés, ainsi que la procédure à respecter lors de leur inscription. A cette occasion, il est créé notamment un article L.223-13 du code monétaire et financier prévoyant qu'un transfert de minibons dans un DEEP équivaut à un contrat écrit⁶³ ; ce qui a été argué comme le premier encadrement en droit français d'un

⁵⁹ Cette expression désigne les actifs existants dans une *blockchain* : cryptomonnaies, jetons, etc.

⁶⁰ Arnaud Lecourt. « Bon de caisse » – Article 2 – 5§, Répertoire de droit commercial juin 2022.

Les minibons sont des bons de caisse spécialement conçus pour le financement participatif. Ils doivent pouvoir être échangés sur les plateformes de crowdfunding ou qui ont opté pour un statut PSI ou CIP. Ces minibons peuvent être souscrits tant par des sociétés commerciales dans l'optique de se financer que par des particuliers ou des institutionnels. Le décret no 2016-1453 du 28 octobre 2016, relatif aux titres et aux prêts proposés dans le cadre du financement participatif, vient ainsi compléter le cadre du financement participatif mis en place dès 2014 en vue de diversifier les sources de financement des petites entreprises et des jeunes entreprises innovantes.

⁶¹ Article L.223-12 du code monétaire et financier : Sans préjudice des dispositions de l'article L. 223-4, l'émission et la cession de minibons peuvent également être inscrites dans un dispositif d'enregistrement électronique partagé permettant l'authentification de ces opérations, dans des conditions, notamment de sécurité, définies par décret en Conseil d'Etat.

⁶² Article L.211-7 du code monétaire et financier : (...) Les titres financiers qui ne sont pas admis aux opérations d'un dépositaire central doivent être inscrits, au nom du propriétaire des titres, dans un compte-titres tenu par l'émetteur ou, sur décision de l'émetteur, dans un dispositif d'enregistrement électronique partagé mentionné à l'article L. 211-3. Toutefois, sauf lorsque la loi ou l'émetteur l'interdit, les parts ou actions d'organismes de placement collectif peuvent être inscrites dans un compte-titres tenu par un intermédiaire mentionné à l'article L. 211-3.

⁶³ Article L.223-13 du code monétaire et financier : Le transfert de propriété de minibons résulte de l'inscription de la cession dans le dispositif d'enregistrement électronique mentionné à l'article L. 223-12, qui tient lieu de contrat écrit pour l'application des articles 1321 et 1322 du code civil...

contrat intelligent⁶⁴.

16. Loi PACTE. Au même moment, un phénomène d'ampleur s'empare de la sphère *blockchain*. Il s'agit des ICO (*initial coin offering*), qui sont des méthodes de levée de fonds, fonctionnant avec des smart contracts, dans lesquelles des individus échangent des jetons, aux utilités diverses dans leur projet, contre des fonds. Afin de mettre un frein aux nombreux abus⁶⁵, le législateur instaure dans la loi n° 2019-486 du 22 mai 2019, dite loi « PACTE »⁶⁶ un visa optionnel pour les porteurs de ces projets, afin de se distinguer qualitativement, qui est délivré après examen de l'autorité des marchés financiers (AMF)⁶⁷. La même loi crée un statut de prestataires de services sur actifs numériques (PSAN), délivrable également par l'AMF, par agrément optionnel⁶⁸ ou enregistrement obligatoire⁶⁹. À cette occasion, elle définit clairement les actifs numériques dans un DEEP que

⁶⁴ Xavier Lavyssière. « Blockchain et titres financiers : décret minimaliste pour réforme ambitieuse », Revue Lamy droit des affaires, n° 144 (1 janvier 2019).

(...) l'inscription d'une cession dans le dispositif électronique constitue un contrat écrit ; c'est à notre connaissance le cas le plus clair d'une reconnaissance légale de la valeur contractuelle d'un smart contract.

⁶⁵ Delpech Xavier. « Le projet de loi PACTE, c'est aussi (un peu) du droit des contrats ». AJ contrat n°7, 2018, p.300

Ce régime, de nature facultative, consiste à proposer aux émetteurs de solliciter un visa préalable de l'Autorité des marchés financiers, aux fins de garantir que leur offre présente les garanties de nature à protéger les investisseurs et à prévenir tout abus.

⁶⁶ Loi n° 2019-486 du 22 mai 2019 relative à la croissance et la transformation des entreprises.

⁶⁷ Rontchevsky Nicolas, Storck Michel et de Ravel d'Esclapon Thibault. « Loi PACTE : innovations et modifications en matière de droit financier ». RTD com. 2019. p.713

(...) En effet, les ICO peuvent présenter un caractère spéculatif très marqué. La loi PACTE a donc fait le choix d'une réglementation précise qui repose sur le principe d'un visa optionnel accordé par l'AMF au profit du porteur de projet innovant.

⁶⁸ Article L54-10-5 du code monétaire et financier : *I.-Pour la fourniture à titre de profession habituelle d'un ou plusieurs services mentionnés à l'article L. 54-10-2, les prestataires établis en France peuvent solliciter un agrément auprès de l'Autorité des marchés financiers, dans des conditions prévues par décret.*

⁶⁹ Article L54-10-3 du code monétaire et financier : *Avant d'exercer leur activité, les prestataires des services mentionnés aux 1° à 4° de l'article L. 54-10-2 établis en France ou fournissant ces services en France, sont enregistrés par l'Autorité des marchés financiers, qui vérifie si...*

peuvent être les jetons⁷⁰ et les cryptomonnaies⁷¹(distinction sur laquelle nous reviendrons plus loin⁷²).

17. Règlement MiCA. Ce cadre national inspira le législateur européen et conduit, à raison de l'essor grandissant des cryptomonnaies sur le sol de l'Union Européenne, à la proposition d'un règlement sur les marchés de cryptoactifs du 24 septembre 2020, COM/2020/593 (règlement MiCa)⁷³. Le texte impose un cadre similaire à la loi PACTE aux porteurs de projets d'ICO et prévoit également un agrément obligatoire pour les prestataires fournissant des services sur crypto-actifs (PSCA)⁷⁴. Additionnellement, il crée des obligations spécifiques pour les émetteurs de *stablecoin*, dont les projets d'initiative privée n'ont pas manqué d'inquiéter les instances de l'Union Européenne⁷⁵. Le texte adopté le 20 avril 2023 est prévu d'entrée en vigueur à partir de juillet 2025 à l'issue d'une période de transition ; il remplacera les dispositions de loi PACTE concernant les ICO et les PSAN.

18. Règlement « Data Act ». Enfin, une proposition de règlement datant du 23 juin 2022 pour « l'équité de l'accès aux données et de l'utilisation des données »⁷⁶ mentionne de façon inédite et explicite les contrats intelligents (en visant en fait les smart contracts⁷⁷). Le texte a pour objectif de

⁷⁰ Article L. 552-2 du code monétaire et financier : *constitue un jeton tout bien incorporel représentant, sous forme numérique, un ou plusieurs droits pouvant être émis, inscrits, conservés ou transférés au moyen d'un dispositif d'enregistrement électronique partagé permettant d'identifier, directement ou indirectement, le propriétaire dudit bien.*

⁷¹ Article L54-10-1 du code monétaire et financier : *toute représentation numérique d'une valeur qui n'est pas émise ou garantie par une banque centrale ou par une autorité publique, qui n'est pas nécessairement attachée à une monnaie ayant cours légal et qui ne possède pas le statut juridique d'une monnaie, mais qui est acceptée par des personnes physiques ou morales comme un moyen d'échange et qui peut être transférée, stockée ou échangée électroniquement*

⁷² V., *supra*, §44

⁷³ Règlement du parlement européen et du conseil sur les marchés de crypto-actifs, et modifiant la directive (UE) 2019/1937

⁷⁴ Article 53 du règlement MiCa : *Les services sur crypto-actifs ne sont fournis que par des personnes morales qui ont leur siège dans un État membre de l'Union et qui ont été agréées en tant que prestataires de services sur crypto-actifs conformément à l'article 55.*

⁷⁵ Thierry Granier, « Règlement MiCA : les marchés de crypto-actifs appréhendés par le droit européen », Bulletin Joly Bourse, n°05, 30/09/2023, p. 30

Par ailleurs, certains projets de grandes sociétés puissantes et les tentatives de quelques entreprises technologiques, y compris des banques qui ont essayé de profiter des espaces peu réglementés, ont alerté les pouvoirs publics et les banques centrales sur les risques de déstabilisation du système financier.

⁷⁶ Règlement du parlement européen et du conseil fixant des règles harmonisées pour l'équité de l'accès aux données et de l'utilisation des données (règlement sur les données) com/2022/68

⁷⁷ V., *infra*, §2

favoriser la circulation des données générées par l'utilisation d'objets connectés, notamment afin de stimuler (et surtout démonopoliser) ce marché⁷⁸. Dans son article 11, elle cite les smart contracts comme une des mesures techniques pouvant être utilisée par un détenteur de données souhaitant les transférer de façon sécurisée et automatique à un co-contractant dans le cadre d'un contrat de partage de données⁷⁹. L'article 30 impose alors que ces smart contracts embarquent plusieurs caractéristiques : robustesse et dispositif de contrôle d'accès, possibilité de suspension et de terminaison du programme, même degré de sécurité que celui conféré par un contrat ordinaire et confidentialité des données⁸⁰.

c) Cadre doctrinal

19. La réception en droit des contrats intelligents. Malgré que peu de dispositions visent précisément les contrats intelligents, la doctrine en droit n'a pas manqué de relever que ceux-ci ne s'abstraient pas d'une réception en droit des contrats⁸¹. L'interaction entre ces derniers est souvent paisible : le droit des contrats peut s'adapter aux contrats intelligents en leur fournissant

⁷⁸ Ibid. Contexte de la proposition - Justification et objectifs de la proposition.

Les objectifs spécifiques de la proposition sont exposés ci-dessous : –Faciliter l'accès aux données et l'utilisation de ces dernières par les consommateurs et les entreprises, tout en préservant les incitations à investir dans les moyens de créer de la valeur à partir des données...

⁷⁹ Ibid. Article 11 : *Le détenteur de données peut appliquer des mesures techniques appropriées de protection, y compris des contrats intelligents, afin d'empêcher l'accès non autorisé aux données et de garantir le respect des articles 5, 6, 9 et 10 ainsi que des conditions contractuelles convenues pour la mise à disposition des données.*

⁸⁰ Il est à noter que ces caractéristiques sont celles que nous proposons d'intégrer à nos smart contracts. V., *supra*, §441 et §445.

Ibid. Article 30 : 1. Le vendeur d'une application utilisant des contrats intelligents ou, à défaut, la personne dont l'activité commerciale, l'entreprise ou la profession nécessite le déploiement de contrats intelligents pour des tiers dans le cadre d'un accord de mise à disposition des données, respecte les exigences essentielles suivantes: (a)robustesse: veiller à ce que le contrat intelligent ait été conçu de manière à offrir un degré très élevé de robustesse afin d'éviter les erreurs fonctionnelles et de résister aux tentatives de manipulation par des tiers; (b)résiliation et interruption en toute sécurité: veiller à ce qu'il existe un mécanisme permettant de mettre fin à l'exécution continue des transactions: le contrat intelligent intègre des fonctions internes qui peuvent réinitialiser le contrat ou lui donner instruction de cesser ou d'interrompre l'opération afin d'éviter de futures exécutions (accidentelles); (c)archivage et continuité des données: prévoir, si un contrat intelligent doit être résilié ou désactivé, la possibilité d'archiver les données relatives aux transactions, la logique et le code du contrat intelligent afin de conserver l'enregistrement des opérations effectuées sur les données dans le passé (vérifiabilité); et (d)contrôle de l'accès: un contrat intelligent est protégé par des mécanismes rigoureux de contrôle d'accès au niveau de la gouvernance et des contrats intelligents.

⁸¹ Garance Cattalano. « Smart contracts et droit des contrats ». AJ Contrats d'affaires - Concurrence - Distribution, 1 juillet 2019, 321.

Pour l'heure, le législateur n'est intervenu que pour donner quelques lignes destinées à régir les technologies blockchain et les cryptoactifs, laissant les smart contracts dans l'ombre. Ce qui ne les empêche pas, nous le verrons, d'être soumis au droit mais, pour le comprendre, il faut savoir à quoi servent les smart contracts.

un cadre clair et adapté malgré les originalités qu'ils apportent⁸², mais non sans limites.

20. Théorie générale des contrats et contrats intelligents. Certaines d'entre elles viennent de la théorie générale des contrats. L'idéologie derrière l'avènement des smart contracts semble, en effet, s'opposer à celle qui sous-tend notre droit des contrats moderne. Selon certains auteurs, les smart contracts seraient l'avatar d'une vision contractuelle dans laquelle les parties seraient en complète opposition entre elles, où il serait sans cesse recherché à optimiser le contrat⁸³, le plus souvent au profit de la partie forte⁸⁴. Leur contexte d'utilisation serait donc celui des relations dites discrètes⁸⁵, soit celles dans lesquelles les individus ne se connaissent pas ou peu, ont des liens ponctuels, impersonnels, marqués par de la défiance⁸⁶. Dans ces contrats, peu de place est laissé au dialogue ou à l'entretien du lien social entre les parties. Les smart contracts peuvent alors être opportunément utilisés pour garantir froidement et implacablement le respect des stipulations de l'accord. Or il existe des contrats dits « relationnels », dans lesquels la relation contractuelle est

⁸² Ancel Bruno. « Contrats - Les smart contracts : révolution sociétale ou nouvelle boîte de Pandore ? Regard comparatiste ». Communication Commerce électronique n° 7-8, Juillet 2018, étude 13

Certes, les smart contracts témoignent d'une révolution numérique ainsi que des mentalités. Ils illustrent l'émergence d'un nouveau paradigme qui s'est construit autour d'impératifs spécifiques au monde des affaires. Toutefois, le droit commun des obligations apparaît a priori comme un fondement suffisant pour réglementer ces contrats sources d'un nouveau souffle démocratique. En effet, pour un auteur, « les principes directeurs du droit des contrats (liberté contractuelle, autonomie de la volonté, consensualisme et licéité des conventions sur la preuve) permettent de « créer un cadre juridique assez souple ».

⁸³ Dominique Legeais. « Blockchain et crypto-actifs : état des lieux », RTDcom., n° 754 (2018).

Le smart contract, comme cela a été souligné, est un droit des forts inspiré par l'analyse économique du droit de l'école de Chicago. Il s'agit d'optimiser les coûts du contrat plus que de veiller à l'intérêt des parties.

⁸⁴ Alain Séché. La morale de la machine, p.148. E. Malfère., 1929.

La morale de la machine est la morale des forts.

⁸⁵ Farshad Ghodoosi. « Contracting in the Age of Smart contracts ». SSRN Scholarly Paper. Rochester, NY, 1 mars 2021. <https://doi.org/10.2139/ssrn.3449674>.

Smart contracts resemble the “truly discrete” exchange transaction hypothetical that Professor Macneil put forward in 1977. Such a transaction would be separated from all present, past, and future relations, and occur between “total strangers, brought together by chance (not by any common social structure)” while each party “would have to be completely sure of never again seeing or having anything else to do with the other.”

⁸⁶ Yves-Marie Laithier. « A propos de la réception du contrat relationnel en droit français », Recueil Dalloz, n° 1003 (2006).

A suivre Macneil et ses partisans, les rapports contractuel se situent sur un axe qui va du contrat « discret » au contrat relationnel. Si le contrat est « discret » (impersonnel), l'intensité du rapport est faible : l'unique but poursuivi par les parties est la réalisation d'un échange ponctuel et isolé. C'est le contrat de l'instant. Les parties ne se connaissent pas nécessairement et, quand bien même elles se rencontreraient pour échanger leurs consentements, elles ne nouent aucun lien psychologique. L'opération contractuelle se ramène à l'échange économique effectué.

marquée par un fort lien social entre les parties, qui dépasse les bords du simple accord écrit⁸⁷. Dans ceux-ci, l'exécution rigoureuse des smart contracts peut être inappropriée. En cas d'irrespect des engagements, il peut être plus opportun de tolérer un écart afin de préserver la relation commerciale dans son ensemble. Ceci fait écho à la théorie du solidarisme contractuel, qui met en avant qu'un contrat doit faire l'objet d'un partenariat, d'une collaboration entre les parties où prime la loyauté et la souplesse ; et ne doit pas être forcément le théâtre d'une opposition d'intérêts antagonistes⁸⁸. Cette rigueur dans l'exécution amenée par l'utilisation des smart contracts s'oppose ainsi au mouvement croissant de protection des parties vulnérables dans les contrats ; en particulier aux mesures défenderesses des consommateurs dans les contrats de masse, qui semblent être un des terrains privilégiés d'application des smart contracts^{89 90}.

21. Qualification des contrats intelligents. En matière de qualification, la relation des contrats intelligents avec le droit des contrats semble moins antagonique. En dépit de la variété dont les smart contracts peuvent être employés dans un contexte contractuel, le code civil peut être susceptible de reconnaître un véritable contrat. En la matière, le principe est en effet celui du

⁸⁷ Hugues Bouthinon-Dumas. « Les contrats relationnels et la théorie de l'imprévision », *Revue internationale de droit économique*, n° 2001/3 (t. XV, 3) (2001): p.339 à 373.

Selon Macneil, les hommes ont recours au contrat pour réaliser des projets qui ne peuvent pas s'insérer dans les cadres étroits du schéma des transactions discrètes.(...) les contrats relationnels reposent en effet sur une implication des parties. une prise en considération de leurs caractéristiques singulières, l'instauration d'une relation durable et le fait qu'il est une source de satisfaction entre les parties.

⁸⁸ Catherine Thibierge-Guelfucci. « Libres propos sur la transformation du droit des contrats », *RTD Civ.*, 1997, p.357.

Le contrat n'apparaît plus, en effet, comme un monde fermé, hermétiquement clos ; il est soumis aux influences extérieures, entre en interaction avec l'ordre juridique qui l'accueille et le modèle. Notre croyance en son intangibilité peut s'en trouver malmenée, notre aspiration à la sécurité aussi ... sauf à voir dans cette nouvelle plasticité contractuelle la source d'une adaptabilité susceptible de doter le contrat d'une pérennité qui pourrait lui conférer, dans notre monde en mutation, une force plus grande que celle de l'immobilisme.

⁸⁹ Garance Cattalano. « Smart contracts et droit des contrats ». *AJ Contrats d'affaires - Concurrence - Distribution*, 1 juillet 2019, 321.

Finalement, le domaine rêvé du smart contract est celui des contrats de masse aux prestations simples. C'est pour eux que le smart contract sera le mieux taillé et l'automatisation des prestations la plus utile. Mais cela suppose que les utilisateurs du service développé par la société qui y a recours soient eux-mêmes utilisateurs d'une blockchain dédiée au développement de ces smart contracts, et qu'ils l'aient donc librement accepté, et à condition que ce recours à des smart contracts ne crée pas de déséquilibre significatif à leur détriment puisque le contrat sera probablement un contrat d'adhésion, voire de consommation.

⁹⁰ Philippe Le Tourneau. *Contrats du numérique 2022-2023 - Informatiques et électroniques*. §011.47, p.32. 12e éd. Dalloz Référence, 2022.

Des contrats ne peuvent être ainsi programmés à l'avance que pour des transactions simples et répétitifs. Ils doivent évidemment respecter toutes les règles du droit des contrats dont, le cas échéant, les règles protectrices des consommateurs.

consensualisme, prévoyant qu'un contrat puisse être formé dès lors qu'est établi un accord de volontés entre des parties capables de contracter et que le contenu de l'accord est libre et certain⁹¹. Dans sa thèse, Madame Leveneur relève trois types de recours aux smart contracts pouvant recevoir la qualification de contrats⁹² :

- celui où les smart contracts sont uniquement dédiés à l'application d'un texte préalablement rédigé en langage naturel détaillant le rapport entre les parties. Cet usage reçoit sans difficulté la qualification de contrat grâce au texte écrit en langage naturel qui renseigne le consentement, le contenu et la capacité des parties en les identifiant⁹³.
- celui où les smart contracts constituent le support unique de la relation entre les parties. Autrement dit, l'accord est exclusivement formalisé par le code informatique des programmes dans la *blockchain*. Déterminer la réunion des conditions de validité d'un contrat peut alors être plus difficile : le code se révèle être instrument médiocre pour éclairer sur le *negotium*⁹⁴, et l'anonymat des parties ne permet pas de déterminer leur capacité à contracter⁹⁵. Pourtant ces caractéristiques n'empêchent pas, en soi, la qualification possible de contrat⁹⁶.

⁹¹ Article 1128 du code civil.

⁹² Claire Leveneur. « Les smart contracts : étude de droit des contrats à l'aune de la blockchain ». §387, p. 287. Thèse de doctorat, Université Paris-Panthéon-Assas, 2022.

Ainsi, l'analyse a permis de découvrir trois nouvelles figures de contrat : le pur smart contract, le smart contract d'application d'un contrat cadre et le contrat hybride. À l'inverse, lorsqu'un contrat prévoit seulement l'exécution de certaines de ses obligations par un smart contract, le smart contract ne constitue qu'une modalité d'exécution et non un nouvel accord de volontés. La réalité d'un smart contract qualifié de contrat à part entière se rapproche et pose des questions importantes lorsqu'il s'agit précisément de mettre en oeuvre la qualification de contrat.

⁹³ Catherine Barreau. « La régulation des smart contracts et les smart contracts des régulateurs ». Annales des Mines - Réalités industrielles Août 2017, n° 3, p. 74-76.

Il suffit alors que les parties à un contrat valablement conclu en dehors de la chaîne de blocs conviennent, dans ce contrat, de recourir à un smart contract pour l'exécution de leur accord. Les principes directeurs du droit des contrats (liberté contractuelle, autonomie de la volonté, consensualisme et licéité des conventions sur la preuve) s'allient pour créer un cadre juridique assez souple et sûr.

⁹⁴ Thierry Debard et Guinchard Serge. Lexique des termes juridiques 2020-2021 - 28e ed. Negotium. Edition 2020-2021. Dalloz, 2020.

Dans un acte juridique ou dans un contrat, le negotium concerne la question de fond que vise cet acte ou ce contrat, par opposition à l'instrumentum qui, en la forme, traduit matériellement la volonté des contractants.

⁹⁵

⁹⁶ Michel Vivant. « Lamy droit du numérique ». Partie 3 - Numériques et contrats - Division 2 le régime général des contrats du numérique de droit privé - Chapitre I - La qualification juridique des différents contrats du numérique - Section 5 Le cas particulier des "smart contracts". Editions Lamy, 2012.

En effet, si le « smart contract » n'est pas, au sens traditionnel du terme un contrat mais plutôt un moyen d'en assurer son

- enfin celui où les smart contracts sont utilisés concurremment avec des stipulations écrites en langage naturel pour régir une relation ; ici, la qualification de contrat sera facilitée, une fois encore, par les stipulations écrites.

22. Régime des contrats intelligents. En dépit de cette qualification, les contrats intelligents n'en soulèvent pas moins de nombreuses interrogations sur leur compatibilité avec le droit des contrats au stade de leur exécution. Comment faire comprendre aux smart contracts les concepts subjectifs de bonne foi, raisonnable ou suffisamment grave afin qu'ils déclenchent des actions⁹⁷? Leur fonctionnement inaltérable est-il conciliable avec les concepts d'exception d'inexécution⁹⁸ ou de révision pour imprévision⁹⁹ ? Comment mettre en œuvre la nullité d'un contrat après l'exécution du *smart contract* ? Pour certains, et dans le *data Act*¹⁰⁰, toutes ces questions mettent en lumière le besoin de réintroduire des tiers humains dans l'automatisme du contrat intelligent¹⁰¹, malgré l'atteinte que cela porterait à l'opportunité de recourir aux smart contracts en premier lieu.

exécution, il serait réducteur de ne pas voir que cet instrument numérique peut être la manifestation tangible du contrat conclu par ailleurs (y compris directement en ligne), voire le résultat de la transmutation de la forme du contrat d'origine en son avatar numérique.

⁹⁷ Gaëtan Guerlin. « Considérations sur les smart contracts », Dalloz IP/IT, 2017, 512.

On pourrait multiplier les illustrations. Comment, par exemple, concilier l'obligation d'exécuter les contrats de bonne foi avec le caractère automatique des smart contracts ? Leurs concepteurs pourront-ils programmer l'exigence de loyauté ?

⁹⁸ Arnaud Lecourt. « Droit des sociétés et numérique – Chapitre 2 Numérique et fonctionnement de la société - Section 2 Les opérations sur les titres », Répertoire IP/IT et Communication, novembre 2020.

(...) Il sera par exemple difficile pour un smart contract d'apprécier, lors de la mise en œuvre d'une exception d'inexécution, que l'inexécution du co-contractant serait suffisamment grave au sens de l'article 1219 du code civil.

⁹⁹ Gaëtan Guerlin. « Considérations sur les smart contracts », Dalloz IP/IT, 2017, 512.

Les contrats enregistrés sur la blockchain sont réputés infalsifiables, intangibles. Mais comment pourra-t-on, demain, en réviser les termes, en application du nouvel article 1195 du code civil ? À l'issue d'une révision judiciaire, le smart contract pourra-t-il lui-même être aisément reprogrammé ?

¹⁰⁰ V., *infra*, §18

¹⁰¹ Célia Zolynski. « Blockchain et smart contracts : premiers regards sur une technologie disruptive ». Revue de Droit Bancaire et financier étude n°4, n° 1 (1 janvier 2017), p. 84-85.

Mais alors, comment concilier cette technique d'exécution des engagements avec certains des concepts fondamentaux de notre droit des contrats tel que celui de bonne foi ou de résiliation unilatérale consacrés par la récente réforme du Code civil ? Diverses solutions pourraient être envisagées, par exemple de programmer ab initio dans le code une faculté de modification des conditions de l'exécution automatique ou de prévoir en amont l'intervention d'un tiers de confiance afin de faire jouer des éléments de flexibilité préalablement identifiés.

23. **Nécessité d'un document *fiat* dans un contrat intelligent.** En tout état de cause, il semble s'être dessiné un consensus de la doctrine juridique sur la nécessité d'encadrer le recours aux smart contracts par des stipulations écrites en langage naturel¹⁰². Le terme de contrat *fiat* a même été dégagé pour désigner spécialement ce document contractuel devant figurer dans le corpus d'un contrat intelligent¹⁰³. Il servirait, d'abord, à garantir l'efficacité du contrat, puisqu'il identifierait clairement des parties autrement cachées derrière des adresses cryptographiques de la *blockchain*¹⁰⁴. Ensuite, il serait surtout utile pour gérer l'imprévisible¹⁰⁵ du recours aux smart contracts : en décrivant la démarche à suivre en cas de dysfonctionnement de ceux-ci et en organisant les modalités de restitution des actifs éventuellement perdus¹⁰⁶. Le document *fiat* fournirait aussi facilement la preuve du *negotium* du contrat, contrairement au contrat intelligent formalisé uniquement par du code informatique. Enfin, il servirait à lever les incertitudes sur les questions de droit international découlant de l'usage des smart contracts, en désignant clairement par des clauses écrites le droit et la juridiction applicable¹⁰⁷.

¹⁰² Mekki Mustapha. « Le smart contract, objet du droit (Partie 2) », Dalloz IP/IT 2019 n°1, p. 27.

Le contrat, instrument de gestion des risques - Contrairement aux smart contractualistes, nous sommes d'avis qu'il convient de « contractualiser » le smart contract, c'est-à-dire de l'enrichir de stipulations figurant au sein d'un contrat fiat. Deux raisons principales peuvent être avancées.

¹⁰³ Mekki Mustapha. « Les mystères de la blockchain ». Recueil Dalloz, n° 37 (2 novembre 2017), p. 2160.

En amont, le plus souvent et encore pour longtemps, le smart contract est précédé d'un contrat « fiat », qui n'est pas algorithmé. C'est au service de ce contrat que le smart contract va être programmé, contrat dans lequel figure également des clauses qui vont venir aménager les risques nés de l'utilisation de ce protocole. Pourquoi ce contrat « fiat » ne peut-il pas être lui-même algorithmé et pourquoi le fonctionnement du smart contract doit lui-même être contractuellement encadré ?

¹⁰⁴ Lord Chancellor and Secretary of State for Justice. « Smart legal contract, Advice To Government », §3. 21, novembre 2021.

However, an agreement reached between parties unknown to one another may give rise to difficulties in practice. Herbert Smith Freehills made the point that, practically speaking, it may be very difficult for a party to seek and enforce a remedy against a counterparty whose identity is unknown.

¹⁰⁵ Mekki Mustapha. « Le smart contract, objet du droit (Partie 2) », Dalloz IP/IT 2019 n°1, p. 27.

La première série de clauses contractuelles doit permettre de gérer l'imprévisible. N'oublions pas que le smart contract n'a rien d'un contrat intelligent. Il ne peut faire que ce que le programmeur a prévu qu'il fasse.

¹⁰⁶ Jeffrey D. Neuburger, Wai L. Choy, et Kevin P. Milewski. « Smart contracts: Best Practices », Thomson Reuters - Practical Law, 2019, p. 11-19.

In addition to typical indemnity clauses, the parties should include in the governing traditional contract provisions to address smart contracts specifically. For example, indemnities protecting the parties that did not code the smart contract with respect to intellectual property infringement by the smart contract and damages resulting from improper operation of the smart contract or from other errors in the smart contract.

¹⁰⁷ Mathias Audit. « Le droit international privé confronté à la blockchain », Revue critique de droit international privé, n° 669, octobre 2020.

III – Intérêt de la recherche

24. Recherche de l'opportunité d'un contrat intelligent. Une *blockchain* peut présenter de multiples avantages pour l'exécution d'un contrat (a), comme elle peut s'avérer inadéquate à plusieurs égards (b). L'objectif de nos travaux est de déterminer une manière de confectionner ces contrats intelligents qui retient toutes leurs opportunités tout en neutralisant leurs défauts (c).

a) Opportunité de la *blockchain* pour l'exécution contractuelle

25. Bénéfices d'une exécution contractuelle dans la *blockchain*. L'usage de la *blockchain* améliore à plusieurs égards l'exécution des conventions. Dans un contrat intelligent, les parties bénéficient en effet :

- d'une réduction du nombre d'intermédiaires nécessaires à son exécution¹⁰⁸,
- d'une transparence complète sur le déroulé des opérations¹⁰⁹,
- et surtout d'une garantie d'exécution bien supérieure à celle procurée par un programme ordinaire, grâce l'immutabilité des smart contracts¹¹⁰.

En revanche, si l'anonymat peut être levé, soit qu'il s'agisse d'une blockchain privée qui l'autorise, soit même qu'il s'agisse d'une blockchain publique ayant intégré dans son code cette possibilité sous certaines conditions, alors la recherche du droit applicable au smart contract reprend tout son sens. Le choix de loi tel que prévu par des conditions générales permettrait d'y procéder de manière très simple.

¹⁰⁸ Lorsqu'un *smart contract* est déployé et fonctionne en autonomie dans la *blockchain*, en principe seule cette dernière est nécessaire pour le faire fonctionner. Il n'y a besoin de nul autre *intermédiaire*. Nous verrons que parfois il doit être réintroduit un tiers pour fournir une information à la *blockchain*.

Wright Aaron, et Primavera De Filippi. « Decentralized Blockchain Technology and the Rise of Lex Cryptographia ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 10 mars 2015.
<https://papers.ssrn.com/abstract=2580664>.

(...) today intermediaries ultimately control the services they provide and retain the power to intervene and unilaterally alter the rules governing their platforms if so desired. Because intermediaries often are identifiable, governments can force them to shut down or modify their rules without impacting other online services. Systems deployed on a blockchain (...) are not subject to the same kinds of limitations.

¹⁰⁹ Dominique Legeais. « Fasc. 534 : Blockchain », *JurisClasseur Commercial*, 1 juin 2023.

La blockchain présenterait plusieurs atouts [pour l'exécution des contrats] (...). Enfin, il y a la transparence. L'ensemble des contrats s'exécutent publiquement.

¹¹⁰ Lord Chancellor and Secretary of State for Justice. « Smart legal contract, Advice To Government », § 2.108, novembre 2021. <https://s3-eu-west-2.amazonaws.com/lawcom-prod-storage-11jsxou24uy7q/uploads/2021/11/Smart-legal-contracts-accessible.pdf>.

26. **L’immuabilité comme proposition de valeur unique de la *blockchain*.** La probabilité qu’un programme informatique fonctionne correctement dépend, pour grande partie, de l’infrastructure dans laquelle il est déployé : si le serveur l’abritant est compromis, alors le programme le sera aussi, ainsi que l’exécution du contrat auquel il est affecté. Même s’il est impossible de créer des programmes sans bogues¹¹¹, la résilience et la disponibilité de l’infrastructure les hébergeant peuvent être maîtrisées. Or une *blockchain* peut fournir l’environnement le plus robuste et inaltérable qui soit pour supporter des programmes exécutant des contrats¹¹². La garantie d’exécution procurée par ces programmes offrent même aux parties la possibilité d’envisager de se passer de la force obligatoire pour faire respecter leur accord¹¹³.

En effet, ce sont ces traits qui ont constitué l’idéal libertaire auquel aspirait les *cypherpunks*¹¹⁴, le groupe de cryptographes ayant inspiré Bitcoin¹¹⁵. Les smart contracts garantiraient de façon si indépendante le respect des engagements, qu’ils rêvaient qu’il soit possible de se passer totalement

(...) If properly coded, a smart legal contract is simply unable to refuse to act, to omit a condition, or to fail to perform so long as the requisite conditions are met. Consultees, including Transpact and Dr Robert Herian, predicted that enforcement action for failure to perform obligations under a contract may therefore be less common in relation to smart legal contracts as compared to traditional contracts.

¹¹¹ Hosking, Ben « The Hosk ». « You Cannot Create Software Without Bugs, Problems and Mistakes ». Geek Culture (blog), 18 septembre 2021. <https://medium.com/geekculture/you-cannot-create-software-without-bugs-problems-and-mistakes-615b6540bc3f>.

¹¹² Chainlink. « What Are Smart contracts and How Can They Revolutionize the Future », 7 juin 2019. <https://blog.chain.link/the-power-of-smart-contracts-what-they-are-and-how-they-can-revolutionize-the-future/>.

Smart contracts are tamper-proof programs on blockchains (...) (they) evolve from today’s probabilistic state, where they will probably execute as desired, to a new deterministic state where they are guaranteed to execute according to their code.

¹¹³ Wright, Aaron, et Primavera De Filippi. « Decentralized Blockchain Technology and the Rise of Lex Cryptographia ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 10 mars 2015. <https://papers.ssrn.com/abstract=2580664>.

As set forth above, through the deployment of increasingly complex systems of smart contracts and decentralized organizations, the technology can be used to establish rules and structures for organizations, formal entities, and potentially even governmental bodies (...). Judicial enforcement of law could also be displaced by blockchain technology.

¹¹⁴ Cypherpunk est un mot-valise formé avec les mots “cipher” (qui signifie chiffrement - pour évoquer la technologie cryptographique) et “cyberpunk” (archétype du personnage rebel appartenant au genre du même nom et évoluant dans une société futuriste et dystopique).

¹¹⁵ Eric Hugues. « A Cypherpunk’s Manifesto », 9 septembre 1993. <https://www.activism.net/cypherpunk/manifesto.html>.

Cypherpunks deplore regulations on cryptography, for encryption is fundamentally a private act. The act of encryption, in fact, removes information from the public realm. Even laws against cryptography reach only so far as a nation’s border and the arm of its violence. Cryptography will ineluctably spread over the whole globe, and with it the anonymous transactions systems that it makes possible.

des institutions légales¹¹⁶ pour contracter¹¹⁷. Nul besoin du *Leviathan* pour faire respecter la parole donnée¹¹⁸. Même en dehors de ces raisons idéologiques, un individu peut avoir divers intérêts à garantir l'exécution d'une convention sans nécessiter le tiers judiciaire. Il peut souhaiter ne pas faire reposer l'efficacité de sa convention sur l'office du juge afin d'éviter sa lenteur, le coût d'une procédure judiciaire, ou la corruption de l'institution...¹¹⁹. Il peut encore avoir besoin de *smart contract* car la force publique refuse de prêter son concours à l'exécution de certains contrats.

27. Le contrat de pari dénué de force obligatoire. Il est pensé, par exemple, aux contrats de pari, qui sans être explicitement prohibés, sont dépourvus de force obligatoire en cas d'irrespect des termes par l'un des parieurs.¹²⁰ Imaginons A, faisant le pari avec B que X événement se produise à la date Y: si X se produit à Y, A remporte une somme d'argent que B doit lui verser ; et inversement si X ne se produit pas à Y. Peu importe l'issu du pari, ni A et B n'auront la possibilité de porter l'affaire devant les tribunaux si l'un ou l'autre ne respecte pas ses engagements à l'échéance. L'article 1965 du code civil à cet égard est sans équivoque : *La loi n'accorde aucune action pour une dette de*

¹¹⁶ Jean-Christophe Roda. « Smart contracts, dumb contracts ? » Dalloz IP/IT, n° 07-08 (4 juillet 2018), p. 397.

À y regarder de plus près, on retrouve chez certains des apôtres des smart contracts la même charge contre les juges et les autres tiers, et une certaine obsession pour la réduction des « coûts de transaction ». Les grandes lignes sont les mêmes, mais le marché a été remplacé par le numérique et l'intelligence artificielle.

¹¹⁷ On pense aux conventions "aberrantes" fantasmées dans les écrits des *cipherpunks* comme les contrats d'assassinat ou d'extorsion.

C.May, Timothy. « The Crypto Anarchist Manifesto », 22 novembre 1992.
<https://www.activism.net/cypherpunk/crypto-anarchy.html>.

An anonymous computerized market will even make possible abhorrent markets for assassinations and extortion.

¹¹⁸ Dans l'œuvre de Hobbes, le Léviathan est figure pour désigner l'Etat/la force publique qui est nécessaire pour faire respecter les contrats passés entre les Hommes dans l'état de nature.

Thomas Hobbes. *Léviathan ou Matière, forme et puissance de l'État chrétien et civil*. Gallimard. Folio Essais, 1651.

If a covenant be made wherein neither of the parties perform presently, but trust one another, in the condition of mere nature (which is a condition of war of every man against every man) upon any reasonable suspicion, it is void: but if there be a common power set over them both, with right and force sufficient to compel performance, it is not void. For he that performeth first has no assurance the other will perform after, because the bonds of words are too weak to bridle men's ambition, avarice, anger, and other passions, without the fear of some coercive power...

¹¹⁹ Cohney, Shaanan, et David A. Hoffman. « Transactional Scripts in Contract Stacks ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 2020. <https://doi.org/10.2139/ssrn.3523515>.

Similarly, in regimes where institutional trust is at a nadir and centralized trading repositories are unreliable, scripts can provide significant value.

¹²⁰ Pierre-Yves Gautier. « Passions et raison du droit en matière de jeux d'argent ». *Pouvoirs* n° 139, n° 4 (2011): 91-101.

Par conséquent, le gagnant à un jeu ne peut réclamer en justice le paiement forcé de son gain ; de sorte que le perdant bénéficie d'une « exception de jeu » : il pourra opposer au premier que ni la loi civile ni le juge ne lui permettront d'obtenir condamnation en justice.

jeu ou pour le paiement d'un pari. Pour contourner le risque d'inexécution, A et B pourront utiliser un *smart contract* dans lequel ils déposeront chacun la somme qu'ils devront verser s'ils perdaient. Celui-ci prévoirait que si X se produit à Y, A récupérera sa somme ainsi que celle de B ; et inversement si X ne se produit pas à Y. À condition d'être correctement codé, le pari entre A et B se retrouverait ainsi solidement garanti d'exécution à l'aide seule d'un *smart contract*¹²¹.

28. La commodité de la *blockchain* pour l'automatisation de transferts d'actifs. La *blockchain* constitue, en sus, un environnement extraordinairement commode pour automatiser dans un contrat un transfert d'actifs (en particulier d'argent¹²²). Cette caractéristique peut justifier à elle-seule l'opportunité de son utilisation en tant qu'instrument d'exécution des contrats, nonobstant celle d'immutabilité. Imaginons que des parties souhaitent automatiser le versement d'une somme d'argent à l'occurrence d'une certaine date. Si elles désirent le faire sans recourir à la *blockchain*, elles devront très probablement créer et déployer leurs programmes sur la plateforme d'une société tierce¹²³. Celle-ci leur fera adhérer à ses conditions générales d'utilisation, sans possibilité de négociation¹²⁴. En outre, elle leur facturera un prix pour l'utilisation de sa plateforme, dont les possibilités d'actions seront limitées et en principe, le fonctionnement du programme sera opaque à la partie non conceptrice. Enfin, la société s'appuiera sur les services d'autres sociétés pour fournir le sien, ce qui ajoutera autant de risques de défaillance sur les programmes des parties. Tandis que dans une *blockchain*, dès lors que l'argent est valablement matérialisé par des cryptoactifs¹²⁵, les parties pourront librement et gratuitement déployer un programme (un *smart contract*) contrôlant, sans limites, les modalités de sa circulation, sur une infrastructure hautement résiliente. Elles pourront chacune suivre son exécution qui sera transparente et ne dépendra d'aucune autre entité que la *blockchain*. Cette dernière offrira donc résilience, libre-accessibilité, désintermédiation, transparence

¹²¹ Dans notre exemple, l'exécution fructueuse du *smart contract* dépend de la bonne récupération et de l'intégrité de l'information sur la réalisation de X à Y. Donc le pari est aussi sécurisé par cette source d'information (l'oracle).

¹²² Nous reviendrons bien plus en détail sur cette notion de transfert d'actifs dans des développements plus bas. Il faut simplement retenir à ce stade qu'on entend la circulation d'éléments ayant une valeur et étant appropriable : la monnaie, des titres financiers, des droits de propriété intellectuelle, etc.

¹²³ Ce serait probablement un prestataire de service de paiement (article L512-1 du code monétaire et financier) comme Mangopay (<https://mangopay.com/>).

¹²⁴ L'extrême majorité du temps, les conditions générales de plateforme ne sont pas négociables et considérées comme des contrats d'adhésion. A tel point que la première monture de l'article 1110 du code civil définissant le contrat d'adhésion employait explicitement le terme de conditions générales. Preuve que ces types de contrat constituent l'archétype des contrats d'adhésion.

Le contrat d'adhésion est celui dont les conditions générales, soustraites à la négociation, sont déterminées à l'avance par l'une des parties

¹²⁵ V., *supra*, §47

et simplicité d'utilisation¹²⁶ aux parties qui souhaitent automatiser dans leur contrat un transfert d'actifs.

b) Inopportunité de la *blockchain* pour l'exécution contractuelle

29. L'inadéquation de la *blockchain* pour l'exécution de contrats. En dépit de ses avantages, la *blockchain* peut, à d'autres égards, se révéler inopportune pour des parties souhaitant s'en servir afin d'exécuter leur contrat. D'abord car sa transparence, par défaut, peut poser problème à celles qui cherchent à contracter à l'abri des regards¹²⁷. En outre, sa résilience est à relativiser : il arrive qu'une *blockchain* fasse l'objet de défaillances passagères interrompant son fonctionnement¹²⁸, ou de défaillances plus graves la compromettant pour un temps prolongé¹²⁹. Ces incidents peuvent alors entraîner l'arrêt des programmes exécutant les contrats, sans qu'il soit possible de rechercher un responsable en raison de la nature décentralisée de l'infrastructure¹³⁰. De plus, malgré sa robustesse et sa transparence, une *blockchain* offre des performances et un confort d'utilisation ne rivalisant pas encore avec ceux des environnements *cloud*¹³¹, qui ont eu le temps de se perfectionner depuis des

¹²⁶ Lee Alexander. « What Is Programmable Money? », FEDS Notes, 23 juin 2021.
<https://www.federalreserve.gov/econres/notes/feds-notes/what-is-programmable-money-20210623.html>.

One facet of successful public blockchain systems that may provide some clarity is how they closely link digital value and programmability in a single system that only functions properly when both are present.

¹²⁷ Clifford Chance. « Smart contracts – Legal Framework and Proposed Guidelines for Lawmakers ». European Bank for Reconstruction and Development, §3. 21, septembre 2018.

The contracting parties may wish to keep the existence of a smart contract and/or its terms and conditions private – and such privacy may also be required under applicable laws (e.g., data protection laws or the laws governing specific industry sectors such as the banking and insurance or health sector).

¹²⁸ V., *supra*, §353

¹²⁹ V., *supra*, §242

¹³⁰ Jean-Michel Mis. « Les technologies de rupture à l'aune du droit ». Dalloz IP/IT n°07-08 2019 p.425

Une autre grande interrogation juridique subsiste également pour ce qui est de la notion de responsabilité. Une blockchain n'a aucune autorité et n'est pas contrôlée par un État, mais par des milliers d'ordinateurs de particuliers et par l'algorithme qui la génère automatiquement à chaque transaction, sans qu'on connaisse forcément l'auteur de cet algorithme, souvent anonyme. À qui donc la faute en cas d'erreur de certification ? Quelles conséquences si le code est mal écrit ? Il est quasiment impossible de trouver un responsable.

¹³¹ Autorité de la Concurrence. Avis 23-A-08 du 29 juin 2023 portant sur le fonctionnement concurrentiel de l'informatique en nuage ("cloud")

L'informatique en nuage, ou « cloud », constitue une des évolutions technologiques au cœur de la numérisation de l'économie. Le cloud comprend l'ensemble des services mutualisés, accessibles via Internet, à la demande, payés à l'usage et, par extension, certaines des infrastructures sous-jacentes (centres de données notamment). En comparaison avec l'informatique traditionnelle, le cloud offre de multiples avantages économiques pour les entreprises. Il permet notamment de nouvelles organisations du travail, sur la base de ressources partagées et accessibles à distance. Ces

années. Autrement dit, il peut sembler plus pertinent pour des parties de faire exécuter leur contrat par des programmes déployés dans des serveurs centralisés dans les nuages, car ils sont privés, commodes, permettent une exécution rapide et sont fournis par des entreprises identifiées s'engageant sur des niveaux de disponibilités élevés¹³².

30. Les risques posés par les smart contracts pour l'exécution de contrats. Les smart contracts en particulier peuvent poser des risques importants en tant qu'instruments d'exécution de contrats. En premier lieu, le fait qu'ils soient principalement utilisés pour gérer des actifs de valeur rend les conséquences de leur dysfonctionnement plus dévastatrices, en plus de faire d'eux des cibles très attrayantes pour les pirates¹³³. Et malgré l'accumulation de savoir-faire autour de ces programmes, ils restent des outils relativement récents qui ne bénéficient pas du même niveau d'expertise en matière de sécurité que des programmes plus vieux et traditionnels. Cependant, leur problème majeur réside dans le fait qu'ils soient inaltérables. Il est techniquement impossible de revenir en arrière et de corriger une opération mal exécutée par ces programmes¹³⁴. Par conséquent, la correction d'une erreur s'avère bien plus délicate à mettre en œuvre, bien que possible¹³⁵, que pour un programme traditionnel. Pour les parties d'un contrat intelligent, la propriété d'inaltérabilité est ainsi à double tranchant : elle garantit comme nul autre le respect des engagements, mais elle rend difficile la gestion des conséquences en cas d'erreur d'exécution du *smart contract*¹³⁶.

caractéristiques peuvent constituer des sources de gains de productivité pour les entreprises et de création de valeur pour l'économie.

¹³² Thibault Verbiest. « Le Service Level Agreement dans les contrats informatiques ». Droit & Technologies, 11 novembre 2003. <https://www.droit-technologie.org/actualites/le-service-level-agreement-dans-les-contrats-informatiques/>.

Le Service Level Agreement ou SLA est une expression anglaise qui peut être traduite par « Accord de Niveau de Service ». [Il] est le contrat ou la partie du contrat spécifiant l'ensemble des niveaux de services à fournir par le prestataire informatique au(x) client(s).

¹³³ Hugo Bordet. « DAO : une nouvelle forme d'activisme pour les associations ? » Juris associations 2022, n°670, p.27

L'un des risques majeurs auxquels sont exposées les DAO [programmes blockchain qui répliquent le fonctionnement d'organisation] est le piratage du smart contract. En effet, le montant du financement que ces « automates » stockent réellement ne cesse d'augmenter tandis qu'ils se complexifient également sérieusement. Cette situation conduit parfois à des bugs et des exploits coûteux, pouvant mettre à mal la pérennité d'une DAO. En effet, nombreuses sont les fois où des hackers ont utilisé les failles liées au développement d'un smart contract pour soustraire et liquider des applications décentralisées. L'exemple le plus connu est l'affaire The DAO.

¹³⁴ Ludovic Mounoussamy. « Le Smart contract, acte ou hack juridique ? », Petites affiches, n° 037 (20 février 2020).

Le smart contract présente plusieurs caractéristiques. Il est autonome. C'est-à-dire qu'une fois déployé, il n'est plus possible de le modifier ou d'empêcher son exécution, sauf dans les cas pré-paramétrés (...). L'élément principal qui définit le smart contract est donc son caractère immuable, intangible, irréversible.

¹³⁵ V., *supra*, §447

¹³⁶ Célia Zolynski. « Blockchain et smart contracts : premiers regards sur une technologie disruptive ». Revue de Droit

c) Objectif et démarche de la recherche

31. Objectif des recherches et problématique. Les parties souhaitant créer un contrat intelligent doivent donc effectuer une étude d'opportunité : les avantages qu'elles comptent en retirer surpassent-ils ses inconvénients ? Puis, elles doivent veiller à ce que son utilisation soit également plus avantageuse que celle des contrats « manuellement exécutés » et des contrats exécutés par des programmes classiques. Nous arguons que les contrats intelligents sont plus opportuns que ces deux contrats lorsqu'ils sont :

- utilisés dans des contextes précis,
- formés d'un *instrumentum* en langage naturel pour garantir leur sécurité juridique (en dépit du consensualisme possible de ces contrats),
- et exécutés par des smart contracts codés et déployés d'une façon particulière.

Notre problématique est donc la suivante : quelle méthodologie d'écriture de contrat intelligent permet d'assurer aux parties l'opportunité de son utilisation ? L'objectif de notre recherche est d'établir un plan de mise en œuvre de ces contrats qui exploite au maximum leurs qualités, tout en réduisant les risques juridiques et techniques associés à leur utilisation.

32. Une démarche interdisciplinaire. Dans cette optique, notre approche adoptera nécessairement une dimension interdisciplinaire. Nous mènerons un travail de réflexion juridique pour déterminer les contextes d'utilisation les plus pertinents des contrats intelligents, ainsi que les moyens légaux à mettre en place pour les sécuriser. Ainsi, nous ferons nôtres les conclusions de la doctrine prônant la rédaction en langage naturel d'un contrat *fiat* dans un contrat intelligent. Et nous déterminerons, sur le plan informatique, la meilleure manière de développer ses smart contracts et les infrastructures les plus adéquates pour les déployer. Nous proposerons, à cette occasion, des modèles de *smart contract* que des parties pourront intégrer dans leur contrat intelligent.

33. Plan de la thèse. La première étape de notre étude consistera à délimiter les domaines du contrat intelligent. Nous établirons, sur le plan matériel, ce qui convient le mieux à exécution par *smart contract* et déterminerons, sur le plan formel, la façon la plus sécurisée dont le contrat doit être formé (PARTIE I). Une fois cette phase achevée, nous aborderons l'élaboration proprement dite du contrat intelligent qui impliquera la rédaction d'un *instrumentum* en langage naturel comportant un certain nombre de clauses et le développement des smart contracts (PARTIE II).

Bancaire et financier étude n°4,(1 janvier 2017), p. 84-85.

Si l'intérêt de la blockchain réside dans la garantie qu'elle offre quant à l'immutabilité des termes du contrat et la publicité des transactions, et donc de leur sécurité, là se trouve dans le même temps la limite possible de cette technique d'exécution de l'engagement contractuel.

PARTIE I – DELIMITATION DU CONTRAT INTELLIGENT

34. Délimitation des domaines du contrat intelligent. Nous avons proposé de scinder le plan d’actions des parties souhaitant créer un contrat intelligent en deux grandes phases : l’une est consacrée à ce que nous nommons la phase de délimitation et l’autre la phase d’élaboration. A l’amorce de la phase de délimitation, les parties ont l’intention d’utiliser la technologie *blockchain* pour l’exécution de leur contrat, mais n’ont encore aucune idée concrète de la mise en œuvre.

Nous estimons que dans un premier temps, elles devront déterminer ce qu’elles souhaitent exécuter dans la *blockchain*¹³⁷ (Titre I). Imaginons que des parties soient tentées d’exécuter par *smart contract* plusieurs contrats de leurs relations d’affaires. Elles devront commencer par sélectionner parmi leurs conventions celles qui se prêtent le mieux à une exécution *on-chain* : est-il opportun de recourir à la *blockchain* pour exécuter un accord de confidentialité, un contrat de vente, un contrat de prestation de services ? Si oui, quelles sont les clauses de ces contrats qui se prêtent à une traduction en code ?

Ensuite, les parties devront définir le domaine formel de leur contrat, c’est-à-dire choisir *l’instrumentum* de leur contrat intelligent (Titre II). Nous verrons que si il est possible de concevoir ces contrats sans les pourvoir de textes écrits en langage naturel, les parties cherchant à conférer un une sécurité juridique satisfaisante à leurs conventions devront les en doter et leur donner un rôle central.

¹³⁷ Cette étape est régulièrement citée par des plateformes de développement de contrats intelligents comme l’une des toutes premières à effectuer dans la réalisation de ces conventions :

Clause. « REALLY Smart (and Legal!) Contracts ». Clause (blog), 28 mars 2018. <https://medium.com/clause-blog/really-smart-and-legal-contracts-a77fcd1d0d10>.

That brings us to the important question of which clauses in a legal contract should be made “smart”. The ability to automate the execution of a clause depending on many factors : what is the business value in automating this clause? Is this a condition that occurs frequently for this type of contract?

Titre I – Le domaine substantiel du contrat intelligent

35. **Les processus contractuels enclins à être exécutés *on-chain*.** Quels sont les processus contractuels qu'il est opportun d'exécuter dans une *blockchain* ? Une fois prise la décision de recourir à la *blockchain* pour exécuter leur contrat, les parties devront commencer par déceler dans leur accord ce qu'il est pertinent de formaliser dans une *blockchain*. C'est-à-dire, reconnaître quelles sont les opérations stipulées dans un contrat qui conviennent le mieux à une traduction en code afin d'être exécutées par un programme informatique (Chapitre I).

Nous ne prétendons pas que seules ces dernières devront être sélectionnées par les parties pour une exécution *on-chain* ; mais nous arguons qu'elles figureront, une fois codés, parmi celles exploitant le mieux les propriétés bénéfiques d'une *blockchain*. Elles constitueront donc des lignes directrices que des parties pourront suivre, dans la mesure qui leur plait, afin d'être assurées qu'elles retiennent tout l'intérêt d'un recours à la *blockchain* dans leur démarche de réalisation d'un contrat intelligent.

36. **Liste de clauses et contrats exécutables *on-chain*.** Une fois ceci établi, nous listerons de façon non exhaustive des contrats et clauses composés de ces processus formant ainsi d'excellents candidats à une exécution *on-chain* (Chapitre II). A cette occasion, nous exposerons pour chacun d'entre eux, la manière avec laquelle un *smart contract* peut être utilisé afin de les exécuter.

37. **Critères d'éligibilité d'un processus à une exécution *on-chain*.** Quels critères doivent posséder un processus contractuel pour former un bon candidat à une formalisation dans la *blockchain* ? Nos recherches font ressortir que ce sont les processus dotés des critères suivants :

- ils mettent en œuvre un transfert d'actif, lequel actif est représentable par un jeton dans la *blockchain* (Section I) ;
- ce transfert est déclenché à l'occurrence d'une condition qui doit avoir des caractéristiques spécifiques (Section II).

Section I - Les transferts d'actifs matérialisés par des jetons

Notions de transferts d'actifs. Nous commencerons par définir ce que nous entendons précisément par les notions de « transferts d'actifs » (§1) et de « jetons » (§2).

§ I - Les transferts d'actifs

Les processus faisant circuler un actif. Ce sont les processus dits « opérationnels » d'un contrat qui se prêtent le mieux à une exécution par *smart contract*, et en particulier ceux mettant en œuvre la circulation (A), à quelque titre juridique que ce soit¹³⁸, d'un actif (B).

A - Les processus de transferts

38. **Les clauses opérationnelles d'un contrat.** Il est possible, dans un contrat, de distinguer ses aspects « opérationnels » et ses aspects « non opérationnels »¹³⁹. Les premiers

¹³⁸ Cession, location, prêt, dépôt... Dans chacun de ces contrats, une chose peut être « déplacée » afin d'être mise à disposition d'une personne.

¹³⁹ ISDA. « Smart contracts and Distributed Ledger – A Legal Perspective – International Swaps and Derivatives Association », août 2017. <https://www.isda.org/2017/08/03/smart-contracts-and-distributed-ledger-a-legal-perspective/>.

désigneraient les clauses d'un contrat imposant des actions concrètes à une partie dépendant d'un événement précis¹⁴⁰. Tandis que les seconds viseraient des clauses n'imposant pas d'actions concrètes ou performatives aux co-contractants¹⁴¹. Par exemple, la clause d'un contrat de vente stipulant que le vendeur doit envoyer un bien après en avoir reçu le paiement est une clause opérationnelle. Elle impose une action claire (l'envoi du bien) après la réalisation d'une condition (la réception du prix). Celle déterminant le droit applicable dans ce même contrat est considérée comme non opérationnelle. Il s'agit davantage d'une indication qui n'impose pas directement d'actions tangibles aux parties. La clause d'un contrat d'entreprise stipulant l'obligation que doit réaliser un prestataire est une clause opérationnelle, celle dans le même contrat indiquant la juridiction applicable ne l'est pas.

Cette distinction n'a pas de fondements juridiques ni doctrinaux. Elle procède plutôt d'une analyse d'experts informatiques, qui s'étaient posés la question de savoir lesquels des aspects d'un contrat sont les plus susceptibles d'être transformés en code¹⁴². Pour eux, il s'agissait des clauses qui, comme le code d'un programme, servaient à instruire des performances¹⁴³. Le code est souvent illustré par la formule « *if....then...* ». Même si celle-ci ne représente qu'une infime partie de ce qu'il est possible d'exprimer dans un langage de programmation, elle démontre bien qu'il a principalement vocation à exprimer la prescription d'actions déclenchées à l'avènement de conditions précises : « si tel évènement se réalise, alors telle action doit être effectuée ».

A legal agreement can be analysed as containing operational and non-operational clauses.

¹⁴⁰ Fabien Gillioz. « Du contrat intelligent au contrat juridique intelligent », Dalloz IP/IT, 2019, 16.

Les clauses opérationnelles se réfèrent à des obligations qui requièrent une action déterminée en fonction de l'avènement d'une condition spécifique, ou d'une période de temps. Par exemple : un paiement contre le transfert d'un bien.

¹⁴¹ Ibid.

Quant aux clauses non opérationnelles, elles se réfèrent à des obligations qui ne sont pas déterminées ou qui n'ont pas de logique conditionnelle. Par exemple : clause de for, clause de droit applicable, clause de confidentialité, clause d'intégralité du contrat.

¹⁴² Clack, Christopher D., Vikram A. Bakshi, et Lee Braine. « Smart contract Templates: foundations, design landscape and research directions ». arXiv:1608.00771 [cs], 15 mars 2017. <http://arxiv.org/abs/1608.00771>.

Part of our remit is to consider the semantics of a contract — i.e. what is the “meaning” of a contract? We view a legal contract as having two aspects: 1. The operational aspects: these are the parts of the contract that we wish to automate, which typically derive from consideration of precise actions to be taken by the parties and therefore are concerned with performing the contract. 2. The non-operational aspects: these are the parts of the contract that we do not wish to (or cannot) automate.

¹⁴³ Dave SMALL, ST Magazine, novembre 1992.

Un langage de programmation est censé être une façon conventionnelle de donner des instructions à un ordinateur(...).

Ce sont donc les processus dans les contrats ayant cette même essence qui peuvent facilement être retranscrits dans des langages de programmation¹⁴⁴. Par exemple, lorsqu'une clause d'un contrat de vente prévoit qu'un vendeur pourra envoyer la marchandise après en avoir reçu le prix, c'est une action qui peut être exprimée selon la logique booléenne. Cela signifie qu'elle peut être exprimée sous la forme d'une structure conditionnelle¹⁴⁵ :

si A (le vendeur) reçoit de B (l'acheteur) la somme correspondant au prix du bien X qu'il lui vend,
→ alors A doit envoyer la marchandise à B.

Cela ne signifie pas que les clauses non opérationnelles ne peuvent pas être formalisées dans un langage de programmation afin d'être traités informatiquement. Depuis des décennies, des chercheurs s'emploient avec succès à représenter la prose juridique dans des langages formels¹⁴⁶. Mais il demeure que les clauses d'un contrat décrivant des performances devant être effectuées à l'occurrence d'événements précis sont les plus enclines à être exprimées dans un langage de programmation afin d'être exécutées par des programmes.

39. L'opération de transfert comme clause opérationnelle. Les smart contracts étant des programmes¹⁴⁷, ils sont donc indiqués pour exécuter ces aspects les plus opérationnels d'un contrat¹⁴⁸ ; mais nous savons également qu'ils fonctionnent dans des infrastructures (les blockchains)

¹⁴⁴ Clack, Christopher D., Vikram A. Bakshi, et Lee Braine. « Smart contract Templates: foundations, design landscape and research directions ». arXiv:1608.00771 [cs], 15 mars 2017. <http://arxiv.org/abs/1608.00771>.

(...) *the semantics of the operational aspects might be simple and easily encoded for automation.*

¹⁴⁵ Larousse, Éditions. « Définitions : booléen, boolien - Dictionnaire de français Larousse ». Consulté le 16 février 2023. <https://www.larousse.fr/dictionnaires/francais/bool%C3%A9en/10192>.

Par logique booléenne, on entend par une structure conditionnelle comme précédemment décrit. L'adjectif booléen signifie « *une variable susceptible de prendre deux valeurs s'excluant mutuellement, par exemple 0 et 1.* »

¹⁴⁶ Wright, Aaron, et Primavera De Filippi. « Decentralized Blockchain Technology and the Rise of Lex Cryptographia ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 10 mars 2015. <https://papers.ssrn.com/abstract=2580664>.

(...) *for decades, scholars have recognized that symbolic logic, like software code, can decrease contractual ambiguity by turning promises into objectively verifiable technical rules.*

¹⁴⁷ V., *infra*, §3

¹⁴⁸ Lord Chancellor and Secretary of State for Justice. « Smart legal contract, Advice To Government », novembre 2021. <https://s3-eu-west-2.amazonaws.com/lawcom-prod-storage-11jsxou24uy7q/uploads/2021/11/Smart-legal-contracts-accessible.pdf>.

Contractual obligations which follow a conditional logic ("if X, then Y") are good candidates for being drafted in code, as conditional logic is inherent in computer programming (...) Cuneyt Eti referred to provisions of this kind as "operational clauses", and provided the following example: party A agrees to pay party B a certain amount of money on the last day of the month until the contract is terminated. Obligations of this kind lend themselves well to automation

très indiquées pour faire circuler de la valeur¹⁴⁹. Aussi, ce n'est pas n'importe quel aspect opérationnel d'un contrat qui sied parfaitement à une exécution dans la *blockchain*, mais celui correspondant au transfert d'actif ou de valeur¹⁵⁰. Ce faisant, les parties utilisent les smart contracts pour ce qu'ils savent faire de mieux¹⁵¹ : transférer des actifs. Par exemple, dans un contrat de vente, la clause imposant au vendeur de céder son bien et celle obligeant l'acheteur de payer le prix sont deux clauses opérationnelles qui consistent en des transferts d'actifs (que sont le bien vendu et l'argent dû, respectivement). Elles conviennent donc très bien à une transformation en *smart contract*. Encore faut-il savoir ce que recouvre précisément les actifs qui peuvent faire l'objet d'un transfert.

B - Les actifs

40. Définition d'un actif. Un *smart contract* est donc particulièrement indiqué pour automatiser la partie opérationnelle des contrats correspondant au transfert d'actifs. Il convient de préciser ce que nous entendons par actif ? D'après l'article 211-1 du plan général comptable (PCG), un actif serait *un élément identifiable du patrimoine d'un agent ayant une valeur économique positive pour lui (...)*¹⁵². Autrement dit, pour qu'il soit considéré comme tel, un actif doit répondre aux critères suivants :

- il est identifiable par son propriétaire ; c'est-à-dire que l'actif peut être isolé d'autres actifs et

because they can easily be converted into code.

¹⁴⁹ Yousif, Ahmed. « Blockchain: The Protocol of Value ». BSV Blockchain (blog), 1 juin 2022. <https://bsvblockchain.org/news/blockchain-the-protocol-of-value/>

The blockchain can be described as a protocol for the exchange of value.

¹⁵⁰ Allen, J. G. « Wrapped and Stacked: 'Smart contracts' and the Interaction of Natural and Formal Language ». European Review of Contract Law 14, n° 4 (19 décembre 2018): 307-43. <https://doi.org/10.1515/ercl-2018-1023>.

The connection between smart contracts and cryptocurrency is not accidental, because smart contracts lend themselves particularly well to manipulating assets such as digital tokens that take the form of immaterial object. the core use case of smart contracts would seem to be where the subject matter of the contract is an immaterial object which can be manipulated directly by the smart contract algorithm.

¹⁵¹ A cet égard, nous pouvons également citer les mots de Vitalik Buterin qui, dès la conception d'Ethereum, voyait les smart contracts avant tout comme des logiciels transférant des actifs numériques selon des règles arbitraires.

Vitalik Buterin. « Ethereum Whitepaper ». ethereum.org. Consulté le 12 octobre 2021. <https://ethereum.org>.

“[smart contracts are] systems which automatically move digital assets according to arbitrary pre-specified rules”

¹⁵² Définition issue du règlement CRC n° 2004-06 du 23 novembre 2004 relatif à la définition, la comptabilisation et l'évaluation des actifs.

être séparable. Son isolation peut, sans y être limitée, résulter d'un droit légal ou contractuel (même si celui-ci n'est pas transférable),¹⁵³

- il est contrôlé par son propriétaire : l'agent propriétaire maîtrise les avantages résultant de cet élément et assume l'essentiel des risques qui y sont liés¹⁵⁴,
- et il a une certaine valeur : l'article 211-2 du PCG parle d'avantage économique futur d'un actif, soit le potentiel qu'a cet actif de contribuer, directement ou indirectement, à des flux nets de trésorerie.

En définitive, la notion "d'actif" peut inclure toutes choses rares ou ayant une valeur, sur lesquelles une personne a un certain contrôle et pouvant être cédée. Ces actifs peuvent être :

- corporels, et donc inclure tous les objets tangibles qui répondent aux trois critères sus-évoqués;
- ou incorporels et inclure pêle-mêle des droits de propriété intellectuelle (droits d'auteurs, brevets, marques...), des droits contractuels (créances, droit au bail...) , des titres financiers (actions, obligations...), de la monnaie (scripturale), etc. Cela peut même inclure des actifs très originaux (comme il est fréquent d'en découvrir dans le milieu de la *blockchain*) comme des points de réputation¹⁵⁵, des droits à intérêts¹⁵⁶, etc.

41. Notion juridique de bien. Il est à noter que nous aurions pu mobiliser la notion

¹⁵³ Article 211-3 du plan comptable général

¹⁵⁴ Article 211-2 du plan comptable général

¹⁵⁵ Born, Cody. « Tokenized Reputation ». The Capital (blog), 26 janvier 2019. <https://medium.com/the-capital/tokenized-reputation-dee463fbc631>.

¹⁵⁶ Dans les protocoles DeFi, il est courant de retrouver des jetons représentant des droits sur les intérêts générés par un dépôt d'autres jetons :

“Lors d'un dépôt dans le protocole Aave, des Aave Tokens (aTokens) sont forgés. Ces tokens particuliers suivent la valeur de l'actif sous-jacent et sont brûlés lors du remboursement du prêt. Pendant que l'actif sous-jacent est prêté aux emprunteurs, les aTokens produisent des intérêts en temps réel.”

Cryptoast. « Aave (AAVE), le protocole de prêt de cryptomonnaies non-custodial », 7 novembre 2020. <https://cryptoast.fr/aave-protocole-pret-cryptomonnaies/>.

juridique de bien à la place de celle d'actif. Ils recouvrent le même sens¹⁵⁷, mais nous avons choisi le terme d'actif car il est la traduction directe de *asset*, qui est le mot systématiquement employé pour décrire ce que sont capables de manipuler les smart contracts.¹⁵⁸ Tout au long de l'étude, le terme actif aura donc la même signification que celui de bien.

Pour que son transfert puisse se faire uniquement à l'aide de *smart contract*, l'actif doit pouvoir être matérialisé par un jeton.

§ II - Les jetons

42. **Actifs enclins à une tokenisation.** Afin que les actifs puissent être transférés par des smart contracts, ils doivent être matérialisés dans la *blockchain* par des jetons (A) ; or seuls certains de ces actifs se révèlent propices à une telle transformation (B).

A - Définition d'un jeton

43. **Différences de définitions des jetons.** Il est intéressant de constater la différence entre la définition technique (a) et légale (b) d'un jeton, où la première est sensiblement plus large que la seconde.

a) Définition technique

44. **Distinction avec le coin.** Les jetons sont, au même titre que les *coins*, des cryptoactifs¹⁵⁹. Les seconds sont générés à l'occasion du processus de consensus d'une *blockchain*

¹⁵⁷ Debard, Thierry, et Serge Guinchard. Lexique des termes juridiques 2020-2021 - 28e ed. Edition 2020-2021. Dalloz, 2020.

Définition d'un bien : « *Au sens juridique, le terme recouvre, d'une part, toute chose, caractérisée par sa rareté, dont l'utilité justifie l'appropriation, d'autre part, tout droit subjectif (réel ou personnel), voire l'absence d'obligation, telle l'absence d'obligation de garantie pour un assureur.* »

¹⁵⁸ Vitalik Buterin. « Ethereum Whitepaper ». ethereum.org. Consulté le 12 octobre 2021. <https://ethereum.org>.

“[smart contracts are] systems which automatically move digital assets according to arbitrary pre-specified rules”

¹⁵⁹ Harsh Kumar. « Demystified: The Difference Between Crypto Coins And Crypto Tokens. Read Here For Details ». <https://www.outlookindia.com/business/demystified-the-difference-between-crypto-coins-and-crypto-tokens-read-here-for-details-news-197683>, 21 mai 2022.

afin de récompenser les nœuds validant les interactions avec la *blockchain*¹⁶⁰. Tandis que *token* est le terme qui a émergé pour désigner les cryptoactifs qui sont créés indépendamment de ce processus de validation des blocs¹⁶¹. Malgré qu'ils s'échangent et cohabitent au sein d'une *blockchain* avec les *coins*¹⁶², les jetons sont créés par leur émetteur à l'aide de *smart contract* et ont le rôle que celui-ci leur choisit. Leur caractère protéiforme rend vain à notre avis tout exercice de catégorisation, mais il est courant de les classer en fonction de trois de leurs utilités les plus populaires¹⁶³ :

- les *currency token*, qui désignent les jetons qui servent à représenter de la monnaie dans la *blockchain* et qui sont utilisés comme moyen d'échange,
- les *utility token*, pour nommer les jetons représentant des droits dans différents produits et services *on-chain*,
- et les *security token*, qui représenteraient des instruments financiers *on-chain*.

b) Définition légale

45. Définition en droit français du jeton. En France, la loi PACTE du 24 mai 2019 a

“We refer to them both as coins as well as tokens. For that matter, many don't even know whether they are buying crypto tokens or crypto coins. Are they both the same?”

¹⁶⁰ V., *infra*, §2

¹⁶¹ Etienne Froment. « Quelle différence entre un *token* et un *coin* pour les cryptomonnaies ? » Consulté le 14 avril 2022. <https://www.20minutes.fr/high-tech/3179779-20211123-token-et-coin-comprendre-les-differences>.

Théoriquement, il est beaucoup plus facile de créer un token qu'un coin. Parce que dans le cas d'un coin, il faut créer toute une technologie qui va avec, et notamment un modèle économique. Un token peut théoriquement être créé en quelques minutes seulement.

¹⁶² Legeais Dominique. *Blockchain et actifs numériques*. 2e édition. Actualité. Paris: LexisNexis, 2021. Chapitre 2, §276.

Dans l'écosystème blockchain, on appelle token ou jeton tout actif transférable numériquement entre deux personnes. Il est émis et échangeable sur une blockchain.

¹⁶³ Nicolas BARBAROUX, Richard BARON, et Amélie FAVREAU. « Blockchain et finance – approche pluridisciplinaire – Les actifs numériques », Répertoire IP/IT et Communication, juin 2020.

« Les jetons : un triptyque de fait. (...) En effet, on peut distinguer trois formes principales de jetons : (1) Les jetons qui servent d'unité de compte et de moyens d'échanges (...) (2) Les jetons qui ouvrent des droits d'accès à des produits ou services futurs proposés par l'émetteur (...) On parlerait ici de jetons utilitaires (« utility tokens » en anglais) (...) (3) Les jetons qui confèrent des droits de participation à la gouvernance ou d'intéressement aux profits futurs. On parlera ainsi de jetons sécuritaires (« security tokens » en anglais) »

donné la définition suivante d'un jeton : *tout bien incorporel représentant sous forme numérique, un ou plusieurs droits pouvant être émis, inscrits, conservés ou transférés au moyen d'un dispositif d'enregistrement électronique partagé permettant d'identifier directement ou indirectement, le propriétaire dudit bien* ¹⁶⁴. Cette définition manifeste le contexte à l'occasion duquel elle a été conçue : celui de régulation des ICO¹⁶⁵ (*Initial Coin Offerings*), où en effet la plupart des jetons émis représentaient des droits dans les produits proposés par des émetteurs¹⁶⁶. La nature conjoncturelle de cette définition a donc pour effet de la rendre trop réductrice : un jeton peut représenter davantage que des droits, il peut représenter un point de réputation¹⁶⁷, un morceau d'identité¹⁶⁸, un certificat d'authenticité... Il peut, en fait, avoir autant de rôles différents que lui donne son émetteur ; et ce dernier peut en créer à un contexte totalement différent de celui d'une ICO.

46. Définition en droit européen du jeton. Le règlement européen “MiCa”¹⁶⁹ définit les

¹⁶⁴ Article L552-2 du Code monétaire et financier créé par l'article 85 de la loi n° 2019-486 du 22 mai 2019 relative à la croissance et la transformation des entreprises

¹⁶⁵ AMF. « Qu'est-ce qu'une Initial Coin Offering (ICO)? » Consulté le 17 février 2023. <https://www.amf-france.org/fr/quest-ce-quune-initial-coin-offering-ico>.

Une offre au public de jetons (Initial Coin Offering ou ICO) est une opération de levée de fonds par laquelle une société ayant un besoin de financement émet des jetons, aussi appelés « tokens », auxquels les investisseurs souscrivent principalement avec des crypto-monnaies. Ces jetons peuvent leur permettre d'accéder, dans le futur, à des produits ou services de cette société.

¹⁶⁶ Charpiat, Hubert de Vauplane et Victor. « [Textes] La réglementation des initial coin offerings (ICO) en France par la loi «PACTE» ». La lettre juridique, 23 mai 2019. <https://www.lexbase.fr/article-juridique/51430539-textes-la-reglementation-des-iinitial-coin-offerings-i-ico-en-france-par-la-loi-pacte>.

Qu'importe, le calendrier législatif n'est pas celui des innovations technologico-financières. La loi relative à la croissance et à la transformation des entreprises (loi «PACTE»), qui crée un cadre réglementaire pour les ICO (francisées en «offres de jetons») et les prestataires de services sur actifs numériques, est le résultat d'un long processus législatif et politique initié durant le second semestre 2017, en parallèle de la fièvre spéculative qui a vu le cours du Bitcoin passer d'environ 2 000 euros à 17 000 euros entre juillet 2017 et décembre 2017 et les fonds levés par le biais d'ICO atteindre des records.

¹⁶⁷ Born, Cody. « Tokenized Reputation ». The Capital (blog), 26 janvier 2019. <https://medium.com/the-capital/tokenized-reputation-dee463fbc631>.

¹⁶⁸ Les *SoulBoundTokens* (SBT) sont un projet de NFT intransférables qui donnent des éléments d'identification sur un individu dans la blockchain de manière décentralisée.

Weyl, E. Glen, Puja Ohlhaber, et Vitalik Buterin. « Decentralized Society: Finding Web3's Soul ». SSRN Scholarly Paper. Rochester, NY, 10 mai 2022. <https://doi.org/10.2139/ssrn.4105763>.

Our key primitive is accounts, or wallets, that hold publicly visible, non-transferable (but possibly revocable-by-the issuer) tokens. We refer to the accounts as “Souls” and tokens held by the accounts as “Soulbound Tokens” (SBTs). (...) Imagine a world where most participants have Souls that store SBTs corresponding to a series of affiliations, memberships, and credentials...

¹⁶⁹ Règlement du parlement Européen et du conseil sur les marchés de crypto-actifs, et modifiant la directive (UE) 2019/1937 com/2020/593

cryptoactifs, qui est une catégorie plus large que celle des jetons¹⁷⁰, comme suit : *des représentations numériques d'une valeur ou de droits qui peuvent être transférées et stockées électroniquement, et qui utilisent une technologie de registre distribuée ou une technologie similaire*¹⁷¹. Le texte propose également un triptyque des jetons :

- les jetons *utilitaires*¹⁷², qui se trouvent définis comme *un type de crypto-actif destiné à fournir un accès numérique à un bien ou à un service, disponible sur la DLT, et uniquement accepté par l'émetteur de ce jeton*¹⁷³;
- les jetons *se référant à un ou des actifs définis* qui sont définis comme *un type de crypto-actif qui vise à conserver une valeur stable en se référant à la valeur de plusieurs monnaies fiat qui ont cours légal, à une ou plusieurs matières premières ou à un ou plusieurs crypto-actifs, ou à une combinaison de tels actifs*¹⁷⁴;
- et enfin les jetons *de monnaie électronique* définis comme *un type de crypto-actif dont l'objet principal est d'être utilisé comme moyen d'échange et qui vise à conserver une valeur stable en se référant à la valeur d'une monnaie fiat qui a cours légal*¹⁷⁵.

Bien que ce triptyque soit plus moderne que celui de la loi PACTE, il ne comprend pas les jetons représentant des titres financiers et les jetons non-fongibles car ils sont en dehors du champ d'application de ce règlement¹⁷⁶.

¹⁷⁰ Bertrand CORBI. « Les risques des cryptoactifs pour la stabilité financière », Dalloz Actualité, 19 novembre 2022.

(...) *cette définition large permet d'inclure l'ensemble des cryptoactifs...*

¹⁷¹ Article 3.§1.(2) du règlement sur les marchés de cryptoactifs

¹⁷² Article 3.§1.(5) Ibid

¹⁷³ Jérôme SUTOIR. « Etat des lieux sur la réglementation des « nouveaux actifs » ». Consulté le 17 février 2023. <https://cms.law/fr/fra/news-information/etat-des-lieux-sur-la-reglementation-des-nouveaux-actifs>.

« les jetons utilitaires de MICA ne couvrent qu'imparfaitement le concept d'utility tokens puisque le projet de texte européen n'envisage leur acceptabilité que par leur émetteur, là où le texte français reste plus large. »

¹⁷⁴ Article 3.§1.(3) du règlement sur les marchés de cryptoactifs

¹⁷⁵ Article 3.§1.(4) du règlement sur les marchés de cryptoactifs

¹⁷⁶ *Si cette définition large permet d'inclure l'ensemble des cryptoactifs, le règlement exclut expressément les security tokens, c'est-à-dire les jetons présentant les caractères de titres financiers (Règl. MiCA, art. 3a) ainsi que les NFT (jetons non fongibles) (Règl. MiCA, art. 2a).*

Bertrand CORBI. « Les risques des cryptoactifs pour la stabilité financière », Dalloz Actualité, 19 novembre 2022.

B – L'opération de « tokenisation »

47. Définition de la tokenisation. La « tokenisation »¹⁷⁷ désigne le fait de représenter ou transformer un actif du monde réel en un jeton dans une *blockchain*. Il s'agit donc du processus préalable que doit subir un actif du « monde réel » avant qu'il puisse circuler dans la *blockchain*. Ainsi, pour pouvoir exécuter le paiement d'une somme d'argent à l'aide de *smart contract* il est nécessaire que les euros soient matérialisés par des « jetons-euros » dans la *blockchain*. Tous les actifs, cependant, ne se prêtent pas également à ce processus. Pour que la transformation soit opportune, il faut que le bien ayant vocation à être « tokenisé » ne soit pas matériel (a) mais immatériel (b), afin d'éviter le problème de l'hybridation.

a) Les actifs non propices à une tokenisation

48. L'hybridation. Nous nommons « phénomènes d'hybridation » les situations dans lesquelles il est fait recours, de manière quasi-équilibrée, à la fois à la *blockchain* et à des éléments en dehors de la *blockchain* pour exécuter un processus. Prenons pour exemple des parties qui souhaitent automatiser une vente de matériels informatiques par *smart contract* : au paiement du prix (*on-chain*¹⁷⁸), le matériel est envoyé. Le versement du prix est un processus de transfert d'actif (cf..) qui pourra être aisément réalisé dans la *blockchain*, mais l'envoi du matériel devra lui être exécuté dans le « monde réel ». Les processus de ce contrat seront donc exécutés à la fois *on-chain* et *off-chain* : une situation hybride. Or, nous pensons que l'opportunité de recourir aux smart contracts pour automatiser des transferts d'actif est, dans une large mesure, fonction du degré d'inclusion de ce processus dans une *blockchain*¹⁷⁹.

49. Les inconvénients de l'hybridation. Cela signifie que plus les parties utiliseront

¹⁷⁷ Fortuné B. AHOULOUMA. « La « tokenisation » des valeurs mobilières dans l'espace OHADA », Revue Lamy droit des affaires, n° 149 (1 juin 2019).

La blockchain ouvre la voie à diverses applications aux fonctionnalités nombreuses au titre desquels figurent la représentation et le transfert de « ressources numériques rares » sous la forme de « tokens » ou de jetons .

¹⁷⁸ *Onchain* signifie dans la *blockchain* et *offchain* en dehors de la *blockchain*.

¹⁷⁹ Cette affirmation est à préciser. En effet, nous verrons plus tard qu'il constitue une bonne pratique lors du développement d'applications, de réaliser le moins d'opérations possibles *on-chain* (afin d'améliorer sa performance). Ces opérations *off-chain*, dans ce cas, sont tout de même effectuées dans un environnement liée à la *blockchain*, c'est pour cette raison que nous nous permettons d'utiliser ce raccourci.

uniquement la *blockchain* pour exécuter leur processus de transferts d'actifs, mieux elles retiendront ses propriétés positives. Tandis que la pratique de l'hybridation aura pour effet de diminuer les effets positifs de la *blockchain* et d'ajouter des effets négatifs dus de la mise en place de l'architecture hybride. Autrement dit, lorsque des parties décident d'utiliser la *blockchain* pour automatiser un transfert d'actif, mais que ces actifs ont une matérialisation en dehors d'elle, les parties sont:

- en tout état de cause, exposées aux effets négatifs classiques du recours à une *blockchain*. Soient les potentiels bugs du protocole, la transparence qui peut ne pas être souhaitée par les parties, l'immutabilité qui peut s'avérer une tare en cas de dysfonctionnement du *smart contract*,
- exposés aux effets négatifs des éléments hors *blockchain*,
- et enfin exposés aux effets négatifs de l'architecture hybride (coût d'implémentation, bugs)¹⁸⁰.

50. Illustration des inconvénients de l'hybridation. Prenons comme exemple original celui d'un individu qui souhaite vendre des bouteilles de vins tokenisées¹⁸¹. L'achat d'un jeton d'une bouteille de vin dans la *blockchain* donnerait droit à une véritable bouteille de vin dans le monde réel. Puisque l'actif est physique, l'achat, par une partie, d'un jeton ne suffit pas à libérer le vendeur de son obligation¹⁸². Ce dernier doit en plus livrer, dans le monde réel, la bouteille de vin à son acheteur¹⁸³. Il y a donc une dissociation entre le jeton et l'actif que le vendeur doit gérer. Il doit

¹⁸⁰ Blockchain Partner. « Comprendre La DeFi (Decentralized Finance) : Définition, Usages, Enjeux, Perspectives – Blockchain Partner ». Consulté le 20 janvier 2022. <https://blockchainpartner.fr/comprendre-open-finance-definition-usages-enjeux-perspectives/>.

« Comme l'explique Joey Krug : les actifs traditionnels qui migreront vers les blockchainss auront des restrictions liées au fait qu'ils touchent le monde réel. Plus les interactions entre le monde réel et les blockchainss sont fortes, plus ces initiatives seront compliquées. »

¹⁸¹ BlockBar. « Glenfiddich Sera Le Premier Partenaire à Commercialiser Un Whisky Rare via NFT Avec BlockBar, La Première Plateforme NFT de Vente Directe Aux Consommateurs Pour Les Vins et Spiritueux ». Consulté le 22 février 2023. <https://www.prnewswire.com/news-releases/glenfiddich-sera-le-premier-partenaire-a-commercialiser-un-whisky-rare-via-nft-avec-blockbar-la-premiere-plateforme-nft-de-vente-directe-aux-consommateurs-pour-les-vins-et-spiritueux-841290934.html>.

« (...) la plateforme NFT innovante de BlockBar, qui permet à Glenfiddich de numériser et de vendre des spiritueux exclusifs sous forme de NFT directement aux consommateurs, que ce soit à des fins de consommation personnelle, de collection ou d'investissement. En achetant le NFT, l'acheteur peut alors devenir le propriétaire du produit physique du monde réel représenté par le NFT... »

¹⁸² Article 1603 du code civil : « [Le vendeur] a deux obligations principales, celle de délivrer et celle de garantir la chose qu'il vend. »

¹⁸³ « BlockBar est également chargée de stocker le produit et, sur demande, de le livrer à l'acheteur. »

BlockBar. « Glenfiddich Sera Le Premier Partenaire à Commercialiser Un Whisky Rare via NFT Avec BlockBar, La Première Plateforme NFT de Vente Directe Aux Consommateurs Pour Les Vins et Spiritueux ». Consulté le 22 février

s'assurer de la correspondance entre un jeton et une bouteille de vin et ainsi mettre en place des systèmes de communication entre le monde réel et la *blockchain* afin de faire correspondre ces deux réalités.

Il est alors exposé à la fois aux inconvénients du recours à un *smart contract*, à ceux du monde *off-chain* (recours à des intermédiaires faillibilité, coût¹⁸⁴, possible perte de produits pendant la livraison) et enfin aux inconvénients découlant de cette architecture hybride (risque de bugs sur le système de communication). La question peut être posée dans ce cas de savoir si les avantages qu'espère retirer ce vendeur de l'automatisation *on-chain* de ses processus contractuels excèdent bien tous ses inconvénients.

b) Actifs propices à une tokenisation

51. Actifs incorporels. Aussi, nous estimons que les actifs neutralisant le mieux ces difficultés sont les actifs incorporels ; car ceux-ci peuvent n'être matérialisés que sur un seul plan : celui de la *blockchain*. Dès lors, les parties n'auront pas besoin d'assurer une correspondance entre deux univers et ne devront seulement se préoccuper de celui dans lequel l'actif est matérialisé. Il n'est, à cet égard, pas surprenant que le phénomène de la tokenisation ait surtout concerné les actifs immatériels¹⁸⁵ suivants :

2023. <https://www.prnewswire.com/news-releases/glenfiddich-sera-le-premier-partenaire-a-commercialiser-un-whisky-rare-via-nft-avec-blockbar-la-premiere-plateforme-nft-de-vente-directe-aux-consommateurs-pour-les-vins-et-spiritueux-841290934.html>.

¹⁸⁴ Albert Ho How. « How Does Tokenization Work, Anyway? » freeCodeCamp.org, 20 octobre 2018. <https://www.freecodecamp.org/news/how-does-tokenization-work-anyway-afb5fed1ac47/>.

There are also other cases when I can trade tokens, but have no guarantees that I can verify the authenticity of the underlying asset. In the case of real estate, it is easier to verify, but other examples include gold bars. If it takes a lot of costs and resources to verify the authenticity, tokenization might not be a viable solution.

¹⁸⁵ Allen, J. G. « Wrapped and Stacked: 'Smart contracts' and the Interaction of Natural and Formal Language ». *European Review of Contract Law* 14, n° 4 (19 décembre 2018): 307-43. <https://doi.org/10.1515/ercl-2018-1023>.

The connection between smart contracts and cryptocurrency is not accidental, because smart contracts lend themselves particularly well to manipulating assets such as digital tokens that take the form of immaterial object. the core use case of smart contracts would seem to be where the subject matter of the contract is an immaterial object which can be manipulated directly by the smart contract algorithm.

- la monnaie ou les devises¹⁸⁶, qui sont déjà largement représentées de façon numérique¹⁸⁷;
- les actifs financiers¹⁸⁸;
- et plus récemment les actifs de propriété intellectuelle¹⁸⁹, à travers le développement fulgurant des NFT.

Ces trois types d'actifs figurent donc parmi ceux posant le moins de difficultés pratiques à une tokenisation et manipulation *on-chain*. La difficulté est que leur tokenisation ne peut pas se faire sans l'aval du législateur et/ou de l'ingénierie contractuelle afin qu'elle porte effet juridiquement. Il doit exister des lois et/ou des stipulations contractuelles qui prévoient qu'un jeton puisse représenter valablement un titre financier (i), une monnaie (ii), ou un actif de propriété intellectuelle (iii).¹⁹⁰

¹⁸⁶ Nous faisons surtout référence aux *stablecoin* sous leur forme la plus répandue (celle où les jetons sont garantis par une réserve d'actifs du monde réel), qui sont souvent appelés « *tokenized cash* »

Liao, Gordon. « Macprudential Considerations for Tokenized Cash ». SSRN Scholarly Paper. Rochester, NY, 23 septembre 2022. <https://doi.org/10.2139/ssrn.4228268>.

¹⁸⁷ Eswar Prasad. *The Future of Money: The End of Cash and the Rise of Digital Currencies*, 2021. <https://www.brookings.edu/events/the-future-of-money-the-end-of-cash-and-the-rise-of-digital-currencies/>.

Moreover, we define and primarily focus on tokenized cash, a class of stablecoins backed by cash-equivalent assets such as treasury bills, and offer one-to-one on-demand convertibility with fiat cash.

¹⁸⁸ OCDE. « The Tokenisation of Assets and Potential Implications for Financial Markets ». OECD Blockchain Policy Series, 2020. www.oecd.org/finance/The-Tokenisation-of-Assets-and-PotentialImplications-for-Financial-Markets.htm.

When it comes to financial assets, tokenisation of securities (equity and/or debt) is seen by the market as the sector with the most imminent potential for growth. This is mainly driven by the recent hype around tokens issued in, mostly unregulated, ICOs and the currently trending 'Security Token Offerings' or STOs, which has been marketed as a more "regulatory-compliant" successor of ICOs aiming to raise capital, as well as 'Security Tokens' representing existing securities in secondary DLT markets.

¹⁸⁹ Wilkof, Neil. « Tokenization of intellectual property for IP rights management ». The IPKat (blog). Consulté le 23 février 2023. <https://ipkitten.blogspot.com/2022/01/tokenization-of-intellectual-property.html>.

"In a nutshell, "tokenization" means using a smart contract (i.e., a computer program) to create a token that is then anchored in a blockchain (...) The result is that the smart contract allows you to represent any IP, e.g., trademarks, designs, patents or copyrights, with a token. Different standards have been developed to mint tokens, such as ERC20 for fungible tokens and ERC721 for non-fungible tokens."

¹⁹⁰ Distributed. « How Tokenization Is Putting Real-World Assets on Blockchains ». Consulté le 23 février 2023. <https://www.nasdaq.com/articles/how-tokenization-putting-real-world-assets-blockchains-2017-03-30>.

"The challenge with intangible assets is ensuring that the blockchain system's model of asset transfer lines up with the real-world legal model of transfer. There may also be jurisdictional differences that can make transfers difficult (although similar, copyright laws differ around the world). That said, intangible assets are often easier to tokenize than physical objects because there are fewer concerns regarding storage and shipment."

i) Les actifs financiers

52. Tokenisation des titres financiers en droit français. En France, depuis l'ordonnance du 8 décembre 2017¹⁹¹ et son décret d'application du 24 décembre 2018¹⁹², il est possible de tokeniser des titres financiers¹⁹³ qui ne sont ni admis aux opérations d'un dépositaire central, à savoir les titres non cotés, ni livrés dans un système de règlement et de livraison d'instruments financiers¹⁹⁴. Cela signifie que la matérialisation de ces titres financiers dans un *dispositif d'enregistrement électronique partagé* (un DEEP¹⁹⁵), a la même valeur légale qu'une inscription en compte¹⁹⁶.

53. Les conditions d'éligibilité des DEEP pour représenter des titres financiers. L'article L. 211-3 du Code monétaire et financier précise donc que la tokenisation des titres financiers sur une *blockchain* est possible, dès lors qu'elle présente *des garanties, notamment en matière d'authentification, au moins équivalentes à celles présentées par une inscription en compte-titres*. L'article R211-9-7 du code monétaire et financier issu du décret d'application du 24 décembre 2018 explicite quelles sont ces garanties :

- le DEEP doit être *conçu et mis en œuvre de façon à garantir l'enregistrement et l'intégrité des inscriptions....*¹⁹⁷ Cela signifie que l'infrastructure hébergeant le registre de titres doit être

¹⁹¹ Ordonnance n° 2017-1674 du 8 décembre 2017 relative à l'utilisation d'un dispositif d'enregistrement électronique partagé pour la représentation et la transmission de titres financiers (s. d.).

¹⁹² Décret n° 2018-1226 du 24 décembre 2018 relatif à l'utilisation d'un dispositif d'enregistrement électronique partagé pour la représentation et la transmission de titres financiers et pour l'émission et la cession de minibons

¹⁹³ Article L211-1 du Code monétaire et financier : (...) *Les titres financiers sont 1. Les titres de capital émis par les sociétés par actions ; 2. Les titres de créance ; 3. Les parts ou actions d'organismes de placement collectif.*

¹⁹⁴ Article 120 de la Loi n° 2016-1691 du 9 décembre 2016 relative à la transparence, à la lutte contre la corruption et à la modernisation de la vie économique, 2016-1691 § (2016) : *le Gouvernement est autorisé à prendre par voie d'ordonnance (...) les mesures relevant du domaine de la loi nécessaires pour : adapter le droit applicable aux titres financiers et aux valeurs mobilières afin de permettre la représentation et la transmission, au moyen d'un dispositif d'enregistrement électronique partagé, des titres financiers qui ne sont pas admis aux opérations d'un dépositaire central ni livrés dans un système de règlement et de livraison d'instruments financiers...*

¹⁹⁵ Le nom que donne le législateur à une *blockchain*. Article L211-3 du Code monétaire et financier : *L'inscription dans un dispositif d'enregistrement électronique partagé tient lieu d'inscription en compte.*

¹⁹⁶ Ibid.

¹⁹⁷ Xavier Lavayssière. « Blockchain et titres financiers : décret minimaliste pour réforme ambitieuse », Revue Lamy droit des affaires, n° 144 (1 janvier 2019).

En informatique, garantir l'intégrité des données consiste précisément à garantir leur inaltérabilité et prévenir leur destruction volontaire ou accidentelle. Pléthore de solutions, dont la réplication des données et les mécanismes de

pérenne et inaltérable. Cette condition est aisément satisfaite pour une *blockchain* publique, qui est généralement bien décentralisée et donc garantie d'être inviolable¹⁹⁸;

- le DEEP doit permettre *une identification directe ou indirecte des propriétaires de la nature et du nombre de titres détenus*. Lorsqu'un registre de titre est déployé dans une *blockchain*, il indique le solde des titres tokenisés associés à chaque adresse. Cette dernière identifie indirectement leur propriétaire¹⁹⁹ ;
- les inscriptions réalisées dans ce DEEP doivent faire *l'objet d'un plan de continuité d'activité actualisé comprenant notamment un dispositif externe de conservation périodique des données*. L'utilisation du DEEP pour héberger les titres tokenisés doit être accompagné d'un dispositif parant à l'éventualité d'une défaillance (du *smart contract* et/ou de la *blockchain*), c'est-à-dire assurant la continuité du service²⁰⁰;
- enfin, *lorsque des titres sont inscrits dans ce DEEP, le propriétaire de ces titres peut disposer de relevés des opérations qui lui sont propres*. Cela signifie qu'un programme devra être mis à disposition des propriétaires des titres tokenisés afin de leur permettre d'extraire leurs relevés d'opérations.

Ces conditions remplies, l'inscription (ou la tokenisation) des titres financiers dans la *blockchain* aura la même valeur qu'une inscription en compte ; et le transfert *on-chain* de ces titres ainsi tokenisés équivaldra à une cession « par papier ».

hachage peuvent concourir à cet objectif. Ce qui est généralement désigné par « blockchain » est de fait une combinaison de ces outils.

¹⁹⁸ V., *supra*, §333

¹⁹⁹ Stéphane BLEMUS et Claire PION. « Blockchain, minibons et titres financiers, Des règles ad hoc pour les chaînes de bloc », *Revue de droit bancaire et financier*, n° 1 (1 janvier 2019).

Le fait que l'identification puisse se faire « directement ou indirectement » autorise qu'un propriétaire de titres sur un DEEP puisse être identifié soit directement par son nom soit indirectement par un pseudonyme (clé cryptographique publique...).

²⁰⁰ Stéphane BLEMUS et Claire PION. « Blockchain, minibons et titres financiers, Des règles ad hoc pour les chaînes de bloc », *Revue de droit bancaire et financier*, n° 1 (1 janvier 2019).

D'ailleurs, cet article ajoute la nécessité que l'utilisation d'un DEEP s'accompagne de la prévision d'un PCA (plan de continuité d'activité) actualisé sur les inscriptions dans le DEEP, et notamment d'un « dispositif externe de conservation périodique des données », visant à renforcer la bonne gestion et l'intégrité dans le temps des données inscrites sur un DEEP et de mettre en place des procédures efficaces pour faire face aux dysfonctionnements éventuels d'un DEEP.

ii) La monnaie

54. Les jetons stables. Il existe de nombreuses cryptomonnaies ayant vocation à être utilisées en tant que monnaie²⁰¹. Celles, toutefois, ressemblant le plus à de la monnaie ordinaire sont les *stablecoin*²⁰² : des jetons ayant une valeur stable, laquelle est le plus souvent celle d'une devise réelle comme le dollar, l'euro ou le yuen²⁰³. Ces jetons constituent des devises tokenisées²⁰⁴ car ils sont la matérialisation, dans la *blockchain*, de monnaies *fiat*²⁰⁵. Seulement, en dépit de leur prolifération, qui a considérablement contribué à l'essor de la *DeFi*²⁰⁶, leur qualification juridique demeure incertaine²⁰⁷. Or, dépourvu du cours légal, des parties ne pourront indifféremment s'en servir

²⁰¹ Satoshi Nakamoto. « Bitcoin: A Peer-to-Peer Electronic Cash System | Satoshi Nakamoto Institute », 31 octobre 2008. <https://nakamotoinstitute.org/bitcoin/>.

Originellement, la cryptomonnaie (ou le *coin*) Bitcoin avait vocation à être une alternative aux monnaies ordinaires. Mais des années plus tard, elle est bien plus utilisée comme réserve de valeur ou bien de spéculation.

²⁰² Matthieu LUCCHESI. « Stablecoins privés et secteur des paiements Une innovation conditionnée par la réglementation », Cahiers de droit de l'entreprise, n° 6 (novembre 2021).

Cependant, ces crypto-actifs sont en général très volatiles. Pour répondre à cette volatilité, une catégorie spécifique de crypto-actifs s'est très largement développée : les stablecoins. La caractéristique commune des stablecoins, qui les distingue des autres crypto-actifs, est de poursuivre l'objectif d'avoir une valeur intrinsèque stable. Cette stabilisation est assurée par des mécanismes variés, dont l'utilisation d'algorithmes et/ou la constitution d'une réserve sous-jacente, composée d'actifs divers.

²⁰³ Nous avons déjà vu que le règlement MiCa distingue deux types de *stablecoin* :

jeton se référant à un ou des actifs: un type de crypto-actif qui vise à conserver une valeur stable en se référant à la valeur de plusieurs monnaies fiat qui ont cours légal, à une ou plusieurs matières premières ou à un ou plusieurs crypto-actifs, ou à une combinaison de tels actifs;

jeton de monnaie électronique: un type de crypto-actif dont l'objet principal est d'être utilisé comme moyen d'échange et qui vise à conserver une valeur stable en se référant à la valeur d'une monnaie fiat qui a cours légal;

Article 3 « Définitions » 1. (3) et (4) du Règlement du parlement Européen et du Conseil sur les marchés de crypto-actifs, et modifiant la directive (UE) 2019/1937 (s. d.) :

²⁰⁴ Liao, Gordon. « Macroprudential Considerations for Tokenized Cash ». SSRN Scholarly Paper. Rochester, NY, 23 septembre 2022. <https://doi.org/10.2139/ssrn.4228268>.

(...) tokenized cash, a class of stablecoins backed by cash-equivalent assets.

²⁰⁵ Le terme de « monnaie *fiat* » désigne les monnaies conventionnelles, ayant cours légal comme l'euro ou le dollar.

Lars, Ludovic. « Qu'est-Ce Qu'une Monnaie Fiat ? » Cryptoast, 20 décembre 2021. <https://cryptoast.fr/monnaie-fiat-definition-explications/>.

²⁰⁶ Thais Lino Cardoso. Stablecoins: The Driving Force behind DeFi, 2022. <https://www.juliusbaer.com/en/insights/think-tank-podcast/stablecoins-the-driving-force-behind-defi/>.

Stablecoins form the backbone of crypto, especially in decentralised finance (DeFi), where their daily transacted value is frequently higher than Bitcoin.

²⁰⁷ Matthieu LUCCHESI. « Stablecoins - Stablecoins privés et secteur des paiements Une innovation conditionnée par

pour éteindre les obligations en somme d'argent de leur contrats intelligents.²⁰⁸ Elles seront contraintes de prévoir des schémas contractuels originaux afin de constituer ces actifs comme moyen de paiement

209 .

55. Les monnaies numériques de la banque centrale. La prolifération des projets de *stablecoin* a conduit des Etats et unions d'Etats à contempler l'opportunité d'émettre leur propre monnaie sous forme numérique²¹⁰ : les monnaies numériques de banque centrale (MNBC), pouvant fonctionner dans une *blockchain*²¹¹. Une telle démarche relève également d'une tokenisation de la monnaie, avec la particularité que le cours légal de cette monnaie ne poseraient pas question²¹². Des

la réglementation - Etude par - Lexis 360 Intelligence », Cahiers de droit de l'entreprise, n° 6 (1 novembre 2021).

En droit français, différentes qualifications juridiques prévues sont susceptibles de s'appliquer pour les stablecoins.

²⁰⁸ En effet, en France la loi impose que le paiement en somme d'argent se fasse en euros.

Article 1343-3 du code civil

Le paiement, en France, d'une obligation de somme d'argent s'effectue en euros.

Toutefois, le paiement peut avoir lieu en une autre monnaie si l'obligation ainsi libellée procède d'une opération à caractère international ou d'un jugement étranger. Les parties peuvent convenir que le paiement aura lieu en devise s'il intervient entre professionnels, lorsque l'usage d'une monnaie étrangère est communément admis pour l'opération concernée.

²⁰⁹ Nous verrons que les parties peuvent mobiliser la technique juridique de la dation en paiement ou l'obligation alternative.

Marin, Gaetan. « Le bitcoin à l'épreuve de la monnaie », AJ Contrat, n° 522 (décembre 2017).

(...) lorsque la dette est libellée en euro, les parties peuvent convenir que le règlement sera réalisé dans une autre devise. Aux termes de cette dation en paiement, l'obligation est éteinte par l'exécution d'une prestation différente de celle due, sans changement ni modification de l'objet et sans aucune création parallèle d'un rapport juridique indépendant » (...) Les commerçants qui acceptent en paiement des bitcoins libellent le prix de leurs produits ou services en euro, mais acceptent que leur créance soit éteinte par le versement de bitcoins.

V., *supra*, §271

²¹⁰ Banque centrale européenne. « Un euro numérique », 8 novembre 2022.
https://www.ecb.europa.eu/paym/digital_euro/html/index.fr.html.

Nous envisageons de lancer une monnaie numérique de banque centrale en Europe pour répondre à la demande croissante de moyens de paiement électroniques sûrs et fiables.

²¹¹ Ibid.

L'Eurosystème teste plusieurs approches et technologies permettant de fournir un euro numérique, y compris des solutions centralisées et décentralisées, comme la DLT. Aucune décision n'a cependant été prise à ce stade.

²¹² Dans l'hypothèse où la banque centrale européenne décide de lancer une MNBC, elle envisagerait de lui donner cours légal. European Central Bank. « Report on a digital euro », octobre 2020.
https://www.ecb.europa.eu/pub/pdf/other/Report_on_a_digital_euro~4d7268b458.en.pdf.

Legal tender status would be a desirable feature of the digital euro.

parties pourraient alors sereinement s'en emparer comme moyen de paiement dans l'exécution de leurs contrats sur *blockchain*. Actuellement, l'Union Européenne est en phase d'investigation sur le projet d'un euro numérique, et prévoit l'amorçage d'une réalisation à partir de octobre 2023²¹³.

56. **La monnaie électronique.** Dans cette attente, certains se sont interrogés sur la qualification des *stablecoin* en tant que monnaie électronique²¹⁴. L'article L315-1 du Code monétaire et financier la définit en effet comme (...) *une valeur monétaire stockée sous une forme électronique, y compris magnétique, représentant une créance sur l'émetteur, qui est émise contre la remise de fonds aux fins d'opérations de paiement définies à l'article L. 133-3 et qui est acceptée par une personne physique ou morale autre que l'émetteur de monnaie électronique*. Or, il a pu être estimé que certains *stablecoin*, pouvant être échangés à tout moment auprès de l'émetteur contre une unité de la monnaie *fiat* qu'ils matérialisent *on-chain*, représentaient effectivement une créance sur l'émetteur et donc remplissaient les conditions de qualification de la monnaie électronique²¹⁵. Le règlement MiCa reconnaît à certains de ces jetons la valeur de monnaie électronique et imposent donc à leurs émetteurs des obligations strictes²¹⁶.

iii) Les actifs de propriété intellectuelle

57. **Les NFT (*non-fungible token*).** L'engouement, à partir de 2021, autour des NFT a révélé l'intérêt de la tokenisation des actifs de propriété intellectuelle, et en particulier des droits

²¹³ Banque centrale européenne. « Un euro numérique », 8 novembre 2022.
https://www.ecb.europa.eu/paym/digital_euro/html/index.fr.html.

La phase d'étude, qui devrait durer environ deux ans, a commencé en octobre 2021 et devrait prendre fin en octobre 2023.

²¹⁴ Hubert de Vauplane. « La nature juridique des *stablecoin* ». Chronique Digitalisation et droit financier, RTDF, 4 septembre 2019.

²¹⁵ European Banking Authority. « Report with Advice for the European Commission on Crypto-Assets », 9 janvier 2019.
<https://www.eba.europa.eu/eba-reports-on-crypto-assets>.

The token is issued on the receipt of fiat currency and is pegged to the given currency (e.g. EUR 1 to 1 token). The token can be redeemed at any time. The actual payment on this network is the underlying claim against Company A or the right to get the claim redeemed. In the assessment of the competent authority the token: (...) Therefore, in the assessment of the competent authority, Company A's proposed token satisfies the definition of 'electronic money' under the EMD2.

²¹⁶ Règlement du parlement Européen et du Conseil sur les marchés de crypto-actifs, et modifiant la directive (UE) 2019/1937

Une troisième sous-catégorie de crypto-actifs comprend les crypto-actifs destinés essentiellement à constituer un moyen de paiement dans le but de stabiliser leur valeur en se référant à une monnaie fiat unique. La fonction de ces crypto-actifs est très semblable à celle de la monnaie électronique(...). Ces crypto-actifs sont appelés «jetons de monnaie électronique».

d'auteur²¹⁷. En effet, leur usage le plus populaire a été celui d'être le support d'œuvres d'arts numériques : la détention d'un jeton équivalait, par convention sociale²¹⁸, à être le propriétaire de l'œuvre auquel il est lié. Le jeton constitue ainsi la matérialisation *on-chain* de la propriété « sociale », et non légale, d'une œuvre, qui peut être aisément échangée, vendue, etc. Pourtant cette tokenisation n'a aucune portée juridique si rien n'est prévu dans la loi ou un contrat²¹⁹.

58. Conditions de cession d'un droit d'auteur. En effet, l'article L131-2 du code de propriété intellectuelle impose que la cession de droits d'auteur soit constatée par écrit et l'article L131-3 du même code explicite les mentions obligatoires qui doivent y figurer.²²⁰ Cela signifie que les parties ne peuvent pas tokeniser des droits d'auteurs, sans avoir au moins mis en place des techniques juridiques permettant que le transfert du jeton représentant les droits soit légalement considéré comme une cession conforme aux règles du code de la propriété intellectuelle²²¹.

59. Conclusion de la section I. Les processus contractuels potentialisant le mieux l'intérêt du recours à une *blockchain* nous semblent être ceux mettant en œuvre des transferts d'actifs immatériels, lesquels doivent pouvoir être matérialisés par des jetons. Cette matérialisation doit porter

²¹⁷ Bamakan, Seyed Mojtaba Hosseini, Nasim Nezhadsistani, Omid Bodaghi, et Qiang Qu. « Patents and Intellectual Property Assets as Non-Fungible Tokens; Key Technologies and Challenges ». Scientific Reports 12, n° 1 (9 février 2022): 2178. <https://doi.org/10.1038/s41598-022-05920-6>.

NFTs are developing remarkably and have provided many applications such as artist royalties, in-game assets, educational certificates, etc (...). Intellectual Property, including patent, trademark, and copyright, is an important area where NFTs can be applied usefully and solve existing problems.

²¹⁸ Sophie Goossens et Nick Breen. « NFTs: Ownership in the Metaverse – the Birth of a New Concept ». Reed Smith Guide to the Metaverse, 1 août 2022. <https://www.reedsmith.com/en/perspectives/metaverse/2022/08/nfts-ownership-in-the-metaverse>.

(...) the concept of NFTs is ingenious and yet very simple: If one cannot own a digital item made of free-flowing information, then let's find something else that may be "owned," separately from the intellectual property. For example, an unfalsifiable certificate of authenticity associated with that digital item.

²¹⁹ Jean Lapousterle. « Les NFT artistiques à l'épreuve des droits d'auteur », Dalloz IP/IT, 2023.

Dans ce sens, une étude récente a pu relever que « la plupart des NFT actuels ne confèrent aucun droit de propriété sur les œuvres d'art sous-jacentes à leurs détenteurs de jetons » et que les « accords entre les émetteurs de NFT et les détenteurs de jetons ressemblent à un dédale de contrats de licence opaques, trompeurs, complexes et restrictifs ».

²²⁰ Article L131-3 du code de la propriété intellectuelle : *La transmission des droits de l'auteur est subordonnée à la condition que chacun des droits cédés fasse l'objet d'une mention distincte dans l'acte de cession et que le domaine d'exploitation des droits cédés soit délimité quant à son étendue et à sa destination, quant au lieu et quant à la durée.*

²²¹ Votre serviteur propose une méthode pour réaliser cette liaison dans un article.

Diallo, Abdoulaye. « TokenIP : Une Proposition de Tokenisation Des Droits d'auteurs En Droit Français ». Medium (blog), 30 décembre 2020. <https://abdoulaye77124.medium.com/tokenip-une-proposition-de-tokenisation-des-droits-dauteurs-en-droit-fran%C3%A7ais-43624b402ecc>.

effet juridiquement, ce qui signifie que le législateur doit lui reconnaître des effets juridiques ou que les parties doivent lui en conférer contractuellement. Ainsi, lorsqu'elles s'interrogeront sur les aspects de leurs contrats propices à une transformation en *smart contract*, elles seront avisées de choisir ces processus si elles veulent être assurées de l'opportunité de leur démarche. Nous verrons, en sus, que ces transferts d'actifs devront être déclenchés à la réalisation d'évènements aux caractères spécifiques.

Section II - Les conditions relatives au déclenchement des transferts

60. Les caractéristiques des conditions de déclenchement des transferts. Comme évoqué, ce sont les processus contractuels ayant, ou pouvoir avoir, la structure suivante : *if...then...*²²² qui apparaissent être les plus opportuns à une transformation en *smart contract*. Après avoir établi que ce qui doit se trouver après le *then* est le déplacement d'un actif, notre attention doit maintenant porter sur ce qui doit se trouver après le *if* ; soit la condition à remplir pour déclencher le transfert d'actif. Selon nous, les parties doivent s'efforcer de choisir une condition objective (§1), automatique (§2) et provenant de la *blockchain* (§3).

§ I – Une condition objective

61. L'évènement déclenchant le transfert de l'actif doit préférablement avoir une nature objective (A), en raison de l'inadéquation des conditions subjectives pour des déclencher des transferts (B).

A – La nécessité d'un évènement objectif

62. Evaluation aisée de la satisfaction de la condition. Les parties doivent s'efforcer de choisir des processus de transferts d'actifs dans lesquels la condition à remplir pour les déclencher a une nature objective. Cela signifie que l'évaluation de la réalisation de la condition doit être le moins possible sujet à contention. Par exemple dans un contrat de cession d'actions, le transfert d'actifs (les actions) a lieu à l'occurrence d'un évènement précis et objectif (le paiement du prix) : soit le prix

²²² V., *infra*, §30

convenu a été payé et le cessionnaire reçoit ses actions, soit il ne l'a pas été et il ne reçoit rien. Il ne peut pas y avoir de discussions sur la réalisation ou non de cet événement. Dans ce cas, le processus de transfert d'actifs s'exécutera avec fluidité et moins de chance de contestation et donc de retour en arrière ; ce qui aura pour effet de renforcer les bénéfices de l'automatisation et des smart contracts.²²³

Dans le cas inverse, où l'événement déclencheur du transfert a une nature très potentiellement discutable ou contentieuse, les parties prendraient le risque d'atténuer le déterminisme du *smart contract*. Autrement dit, elles ouvriraient la porte à la possibilité de l'interrompre, de le retarder ou même de revenir sur son résultat. Cela nuirait aux bénéfices de l'automatisation, en outre de porter atteinte à l'une des propriétés les plus bénéfiques des smart contracts : leur capacité à constituer des alternatives solides à la force obligatoire²²⁴. Imaginons un contrat de développement de site internet exécuté *on-chain* : le prestataire est automatiquement payé dès lors qu'il termine le site. Le transfert d'actifs, qu'est l'envoi du prix au prestataire, dépend d'un événement, qu'est le développement complet du site, qui peut être grandement sujet à dispute²²⁵. Ce n'est donc pas une condition objective.

Or, si l'événement déclenchant un transfert est aisément disputable, il y a des chances pour que les parties contestent souvent le résultat de l'automatisation contractuelle devant les juges chaque fois qu'il ne leur plaît pas et/ou qu'il y a un doute sur sa réalisation. Pourtant l'intérêt de recourir à un *smart contract* est, dans une certaine mesure, de lui abandonner l'exécution de processus contractuels afin de les rendre autonomes. Si des parties reviennent constamment sur son résultat, elles heurtent

²²³ Dans un article, Dan SELMAN parle de transformer en *smart contract* seulement la partie du contrat correspondant au « *happy path* », soit le processus du contrat qui est le moins susceptible de poser problème lors de son exécution.

Dan Selman. « REALLY Smart (and Legal!) Contracts ». Clause (blog), 28 mars 2018. <https://medium.com/clause-blog/really-smart-and-legal-contracts-a77fcd1d0d10>.

In software development we sometimes talk about the Happy Path and exception handling. There's enormous value in automating the Happy Path through a contract (the expected behaviours) while using the existing human arbitration and court system to handle exceptional circumstances that rarely occur, or where the desired behavior is uncertain.

²²⁴ Leveneur Claire. « Les smart contracts : étude de droit des contrats à l'aune de la blockchain ». Thèse de doctorat, Université Paris-Panthéon-Assas, 2022. <https://www.theses.fr/2022ASSA0063>.

Dans toute sa rigidité, l'inflexibilité offerte par le code informatique est le garant du respect des engagements des parties et de leur exécution. L'inflexibilité est synonyme de certitude de l'issue de la transaction, puisqu'il n'y a pas de porte ouverte à l'interprétation. Cette caractéristique est alors un nouvel outil au service de la maxime « pacta sunt servanda » et donne à la force obligatoire des contrats toute sa puissance.

²²⁵ Anthony BEM. « Responsabilité des concepteurs et développeurs de sites internet : l'importance du "PV de recette" ». Legavox (blog), 30 octobre 2013. <http://www.legavox.fr/blog/maitre-anthony-bem/responsabilite-concepteurs-developpeurs-sites-internet-12867.htm>.

La réalisation, la conception et le développement d'un site internet donnent lieu à un contentieux fréquent et fourni.

l'opportunité d'y recourir en premier lieu. Il faut donc, en amont, choisir des processus où ce risque est diminué²²⁶.

De plus, il est tout simplement plus simple d'évaluer des conditions par nature objective que des conditions subjectives. En effet, souvent les événements objectifs sont produits ou facilement quérables par des programmes tandis que les conditions subjectives nécessitent presque toujours le jugement d'un humain ; ce qui ajoute au coût de constitution du *smart contract*.²²⁷ Ainsi, il est plus simple de coder un *smart contract* envoyant une somme d'argent à une date précise que de coder le même programme déclenchant le transfert de sommes après l'écoulement d'un délai raisonnable. Dans le premier cas, un programme pourra facilement et fiablement récupérer l'information sur la date convenue, tandis que dans le deuxième cas il faudra faire intervenir un humain afin qu'il détermine si un délai raisonnable s'est écoulé.

63. Exemples de conditions objectives. Parmi ces conditions objectives, peu contestables par nature, nous pouvons trouver :

- Le résultat d'opérations mathématiques, comme des additions, soustractions, multiplications, divisions, relations d'égalité, de supériorité, d'infériorité, qui sont des données issues de sciences exactes²²⁸ et qui ne vont donc pas susciter de contestations sur leur réalisation ou non. Par exemple une clause de reversement de dividendes : des parties font dépendre le transfert de sommes d'argent au fait qu'un individu ait un solde en « jetons-actions » supérieur à 0²²⁹ ;

²²⁶ Surden, Harry. « Computable Contracts ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 2012. <https://papers.ssrn.com/abstract=2216866>.

“These approaches are not suited for contracting scenarios involving significant amounts of uncertainty, abstraction, or complexity. Rather, they are geared toward the subset of contracting in which the application of contract terms is expected to be relatively non-controversial in the ordinary case.”

²²⁷ Tjong Tjin Tai, Eric. « Formalizing Contract Law for Smart contracts ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 18 septembre 2017. <https://papers.ssrn.com/abstract=3038800>.

“A simple evaluative rule using objectively qualified values and simple logical or mathematical operators. This lends itself to direct formulation in a programming language. A complex evaluative rule of which the inputs are unclear, may refer to the outside, offline world, while the manner of ‘weighing’ the inputs is unclear, and furthermore the viewpoint of the other party would usually be required. (...) Such a rule at present requires human judgement for assessment, and assumes input from both parties.”

²²⁸ Dans une large mesure.

²²⁹ Si un individu est indiqué avoir un solde en « jetons-actions » supérieur à 0 cela signifie qu'il est actionnaire et qu'à ce titre il doit lui être reversé des dividendes.

- Les dates sont des données particulièrement objectives qui peuvent conditionner le transfert d'un actif. Par exemple, envoyer le montant d'un loyer tous les 15 du mois ;
- Sur la situation géographique. Si l'instrument le mesurant est fiable, il est difficile de contester la localisation géographique d'une personne ou d'un objet. Par exemple le transfert d'une somme d'argent dès lors qu'un colis a traversé une frontière ;
- Un phénomène naturel tels qu'une température, une tempête, un taux de précipitation...Ce sont des événements qui ne posent pas de questions sur la détermination de leur occurrence ;
- L'état d'une personne : la naissance ou la mort d'une personne (physique ou morale). Ce sont des événements très objectifs dans leur nature dont l'évaluation ne pose pas non plus de difficultés.

De manière générale, il s'agira de tout événement auquel des individus pourront répondre de façon binaire, "oui ou non", avec très peu de possibilité de débat, si ils ont eu lieu.

B - L'inadéquation des conditions subjectives

64. Les standards et concepts *mous*. Les événements dont il est facile de débattre de la réalisation ne constituent pas des conditions objectives, idéales pour déclencher des transferts d'actifs. La doctrine juridique a ainsi l'habitude de relever que les standards juridiques²³⁰ et autres concepts juridiques nécessitant le jugement d'humains figurent par ces informations qui ne peuvent pas être comprises par des programmes et qui forment donc de « mauvaises » conditions activant les smart contracts²³¹: l'« importance déterminante²³² », « l'avantage manifestement excessif²³³ », « le

²³⁰ Jean-Michel Bruguière. Les standards de la propriété intellectuelle. Dalloz. Thèmes & commentaires, 2018.

²³¹ Mekki, Mustapha. « Le smart contract, objet du droit (Partie 2) », Dalloz IP/IT, 2019, 27.

« L'algorithme du smart contract peut également difficilement intégrer les notions à contenu variable. Comment algorithmer la « disproportion manifeste », les « conséquences suffisamment graves », le « raisonnable », le « légitime », le « significatif » ? »

²³² Article 1112-1 du Code civil alinéa 2 : *Ont une importance déterminante les informations qui ont un lien direct et nécessaire avec le contenu du contrat ou la qualité des parties.*

²³³ Article 1143 du Code civil : *Il y a également violence lorsqu'une partie, abusant de l'état de dépendance dans lequel se trouve son cocontractant à son égard, obtient de lui un engagement qu'il n'aurait pas souscrit en l'absence d'une telle contrainte et en tire un avantage manifestement excessif.*

déséquilibre significatif²³⁴ », la « disproportion manifeste ²³⁵», « la gravité suffisante²³⁶ »...

Contrairement à ce qu'une partie d'entre elle avance cependant²³⁷, nous pensons que les smart contracts peuvent tout de même déclencher des versements à l'occurrence de ces conditions « molles ». Il est tout à fait possible de confier la tâche à un humain de déterminer s'il s'est, par exemple, écoulé un délai raisonnable et de renseigner cette information dans un *smart contract*. Dans ce cas-là, ce dernier "comprendrait" qu'un délai raisonnable s'est écoulé et pourrait ainsi transférer des actifs. Ce serait pourtant inopportun d'utiliser une telle condition à la place d'une plus objective car celle qui est subjective serait beaucoup plus susceptible de mener à de la contention, c'est-à-dire à pousser les parties à contester devant le juge le résultat du *smart contract*.

§ II – Une condition automatique

65. Production automatique de la condition. Un des intérêts du recours à un *smart contract* est de rationaliser l'exécution contractuelle : soit augmenter sa productivité en déléguant à des programmes le soin d'exécuter les stipulations d'un contrat. S'agissant du processus de transfert d'actif (qui est donc exécutable par un *smart contract*), la condition déclenchante du transfert doit ainsi, préférablement, être produite par un programme afin de conserver les bénéfices de la rationalisation. Autrement dit, il faut que les parties s'efforcent d'automatiser intégralement le processus de transfert d'actif, ce qui inclut la production et l'évaluation des événements déclenchant lesdits transferts²³⁸.

²³⁴ Article L442-1 du Code de commerce : *Engage la responsabilité de son auteur et l'oblige à réparer le préjudice causé le fait, dans le cadre de la négociation commerciale, de la conclusion ou de l'exécution d'un contrat, par toute personne exerçant des activités de production, de distribution ou de services : (...) 2° De soumettre ou de tenter de soumettre l'autre partie à des obligations créant un déséquilibre significatif dans les droits et obligations des parties.*

²³⁵ Article 1221 code civil : *Le créancier d'une obligation peut, après mise en demeure, en poursuivre l'exécution en nature sauf si cette exécution est impossible ou s'il existe une disproportion manifeste entre son coût pour le débiteur de bonne foi et son intérêt pour le créancier.*

²³⁶ Nathalie Finck, Samuel Seroc. « Notion de faute d'une gravité suffisante justifiant le licenciement d'un salarié protégé », *Gazette du Palais*, n° 01 (11 janvier 2022): 31.

²³⁷ "Tout versement supposant une condition dépendant d'un standard ou d'une appréciation personnelle ne peut faire l'objet d'un smart contract (paiement d'un prix si la qualité fournie est jugée « suffisante », ou si une livraison est intervenue dans un délai « raisonnable », bonne foi, clauses de « best efforts »)".

Garance, Cattalano. « Smart contracts et droit des contrats ». *AJ Contrats d'affaires - Concurrence - Distribution*, 1 juillet 2019, 321.

²³⁸ "Mais évidemment, faire appel à un tiers pour décider du déclenchement du smart contract, c'est perdre l'automatisme et l'efficacité qu'on lui alloue généralement."

66. Exemples de conditions produites par des programmes. Ces programmes fournissant les informations sur la réalisation des événements déclenchant des transferts peuvent ainsi être :

- des smart contracts . Exemple : le versement d'actions-jetons à un associé ayant fait un apport en euros-jetons dans le *smart contract* d'une SAS. Le transfert d'actifs (le versement des actions-jetons) est déclenché par un événement (l'apport dans le *smart contract*) provenant du *smart contract*²³⁹ ;
- le logiciel d'un thermomètre connecté²⁴⁰. Exemple : une assurance paramétrique dont l'objet est d'indemniser un agriculteur en cas de sécheresse. Le transfert d'actifs (le versement de l'indemnité) est déclenché par un événement (la dépassement d'une certaine température) mesuré par le programme du thermomètre;
- un logiciel SaaS ordinaire. Exemple : imaginons que des parties s'accordent sur les conditions de niveaux de service²⁴¹ d'un logiciel ; si la disponibilité d'un logiciel descend en dessous d'un certain seuil, alors l'utilisateur du logiciel est indemnisé d'un montant. Le transfert d'actifs (le versement du montant de la clause pénale) est déclenché par un événement (l'indisponibilité du logiciel SaaS) mesuré par un logiciel ordinaire.

D'autres informations peuvent ne pas avoir été originellement produites par des programmes mais être renseignées de façon si redondante sur différentes bases données qu'elles sont aussiaisément quérables que des données générées par des programmes. Ces données sont généralement accessibles

Garance, Cattalano. « Smart contracts et droit des contrats ». AJ Contrats d'affaires - Concurrence - Distribution, 1 juillet 2019, 321.

²³⁹ Gaurav Agrawal. « How to Build a Dividend Token With Solidity », 15 novembre 2018. <https://www.crowdbotics.com/blog/how-to-build-a-dividend-token-with-solidity>.

²⁴⁰ Evans, Steve. « Parsyl Sensor Driven Parametric Insurance Pays-out in Less than 8 Hours - Artemis.Bm ». Artemis.bm - The Catastrophe Bond, Insurance Linked Securities & Investment, Reinsurance Capital, Alternative Risk Transfer and Weather Risk Management site, 29 juin 2021. <https://www.artemis.bm/news/parsyl-sensor-driven-parametric-insurance-pays-out-in-less-than-8-hours/>.

²⁴¹ Thibault Verbiest. « Le Service Level Agreement dans les contrats informatiques ». Droit & Technologies, 11 novembre 2003. <https://www.droit-technologie.org/actualites/le-service-level-agreement-dans-les-contrats-informatiques/>.

Le Service Level Agreement ou SLA est une expression anglaise qui peut être traduite par « Accord de Niveau de Service ». [II] est le contrat ou la partie du contrat spécifiant l'ensemble des niveaux de services à fournir par le prestataire informatique au(x) client(s).

par des requêtes d'API²⁴² :

- des informations sur une personne morale (chiffre d'affaires, bénéfices, existence d'une procédure collective ou non)²⁴³,
- des informations sur une personne physique (naissance, mort²⁴⁴, statut étudiant, demandeur d'emploi²⁴⁵),
- des informations sur un bien (valeur foncière d'une maison²⁴⁶).

§ III – Une condition provenant de la *blockchain*

67. En raison de l'inopportunité de l'hybridation²⁴⁷ (A), les parties seront avisées de ne choisir comme événements déclencheurs des transferts d'actif, ceux provenant uniquement de la *blockchain* (B).

A - L'inopportunité de l'hybridation pour les événements déclencheurs de transferts d'actifs

68. Un événement *on-chain*. Nous avons déjà évoqué pourquoi les parties doivent s'efforcer de choisir des actifs qui peuvent n'être matérialisés que sur une *blockchain* afin que leur

²⁴² « Interface de programmation d'application (API) | CNIL ». Consulté le 27 février 2023. <https://www.cnil.fr/fr/definition/interface-de-programmation-d-application-api>.

Une API (application programming interface ou « interface de programmation d'application ») est une interface logicielle qui permet de « connecter » un logiciel ou un service à un autre logiciel ou service afin d'échanger des données et des fonctionnalités.

²⁴³ Ces informations peuvent être récupérées en sollicitant l'API du site infogreffe.com

²⁴⁴ Ces informations peuvent être récupérées en sollicitant l'API du site deces.matchid.io à l'aide des données de l'INSEE.

« Le site <https://deces.matchid.io> est dorénavant réalisé conjointement avec la DNUM (ministère de l'Intérieur) (...). Les travaux conjoints ont permis l'élaboration d'une première version d'API Rest de recherche des personnes décédées, qui motorise entièrement le site. Les données suivent le modèle des données source INSEE... »

« API des personnes décédées - data.gouv.fr ». Consulté le 27 février 2023. <https://www.data.gouv.fr/fr/reuses/api-des-personnes-decede/>.

²⁴⁵ Ces informations peuvent être récupérées en sollicitant l'API du gouvernement : <https://api.gouv.fr/>

²⁴⁶ Cette information peut être récupérée en sollicitant l'API DVF du gouvernement : <https://www.economie.gouv.fr/particuliers/prix-immobilier-estimation-demande-valeur-fonciere>

²⁴⁷ V., *infra*, §49

transfert ne s'opère que sur un seul plan (ce qui retient mieux les effets bénéfiques du recours à une *blockchain*²⁴⁸). Ce même raisonnement est applicable aux conditions déclenchant les transferts : toujours dans le but d'éviter les inconvénients de l'hybridation, les événements à l'origine des transferts doivent, préférablement, ne pas provenir d'un autre endroit que celui de la *blockchain*.

L'information provoquant le déplacement ou non de l'actif doit être issue du même lieu dans lequel réside ce dernier. Si les parties s'extirpent de l'îlot que constitue la *blockchain* pour récupérer cette information, alors elles créeront une situation hybride qui fera diminuer les effets positifs de l'univers *on-chain* de ceux négatifs de l'univers *off-chain*.

69. Illustration avec un contrat d'assurance. Imaginons deux contrats d'assurance²⁴⁹ exécutés à l'aide de *smart contract*:

- dans le premier, le risque assuré est le défaut de remboursement d'un prêt²⁵⁰ réalisé entièrement dans la *blockchain*. Un prêteur met à disposition d'un emprunteur des jetons-euros. Si ce dernier ne rembourse pas à échéance l'emprunteur en lui transférant le montant de jetons-euros emprunté, alors le prêteur peut se voir indemniser par la compagnie l'assurance.
- dans le second, le risque assuré est un retard d'avion. Si l'avion qu'a pris un assuré atterrit plus de 2 heures après son heure d'arrivée indiquée sur le billet, alors il recevra une indemnisation de la compagnie d'assurance²⁵¹.

²⁴⁸ V., *infra*, §48

²⁴⁹ Nous verrons plus tard en quoi les contrats d'assurance paramétrique constituent des contrats très appropriés à une transformation en contrat intelligent.

²⁵⁰ Il s'agit d'une « assurance-crédit » qui est « *une assurance qui protège les entreprises contre le risque d'impayés en leur permettant d'être couvertes et indemnisées en cas de non-paiement de leurs créances commerciales, sur leur marché domestique comme à l'export.* »

Diry, Jean-Charles, Christine Lechat, Serge Pintiaux, Fabien Guirao, Monique Teyssier, Catherine Sainty, Bruno Mousset, et al. BLOC 1 - Gérer la relation avec les clients et les fournisseurs de la PME. Foucher, 2018.

²⁵¹ Il s'agit du fameux exemple d'assurance paramétrique de Axa, Fizzy, qui n'existe plus aujourd'hui.

After the successful payment of the customer, the new insurance policy is created and written on the blockchain with the help of the smart contract. The smart contract then obtains flight data or flight delay data from publicly available databases via an oracle. The time of arrival is also processed onto the blockchain, and if the flight is delayed by more than 2 h, a payment is automatically made to the customer.

Hoffmann, Christian Hugo. « A double design-science perspective of entrepreneurship – the example of smart contracts in the insurance market ». *Journal of Work-Applied Management* 13, n° 1 (1 janvier 2020): 69-87.
<https://doi.org/10.1108/JWAM-08-2020-0037>.

Dans le premier contrat, la condition qui déclenchera le transfert de l'indemnité provient de la *blockchain*, puisque le remboursement doit avoir lieu en son sein. L'information indiquant que l'emprunteur a remboursé ou non le prêteur à la date d'échéance est, en effet, lisible *on-chain*. Le processus se retrouve donc entièrement exécuté dans la *blockchain* : la condition et le transfert d'actif proviennent du même endroit, donc le processus est assuré d'être exécuté de manière à conserver les propriétés bénéfiques de la *blockchain* de bout en bout.

Dans le second contrat, le transfert de l'indemnité sera réalisé *on-chain*, mais l'information sur le retard de l'avion proviendra forcément d'en dehors de la *blockchain*. Les parties feront nécessairement recours à un oracle (un acteur chargé d'approvisionner au *smart contract* l'information sur l'heure d'arrivée de l'avion²⁵²), qui fera déclencher ou non le transfert de l'indemnité²⁵³. Or cette information n'a pas les mêmes propriétés de fiabilité que celle du premier contrat provenant de la *blockchain* : elle sera centralisée et donc soumise à un risque d'indisponibilité, et de perte d'intégrité²⁵⁴. En d'autres termes, la condition déclenchant le transfert (alors même qu'elle est objective et produite par un programme) fera perdre des bénéfices de la *blockchain*. Cela aura pour effet de diminuer l'opportunité générale de recourir à un *smart contract* sans toutefois l'éliminer²⁵⁵.

²⁵² V., *supra*, §454

²⁵³ Hoffmann, Christian Hugo. « A double design-science perspective of entrepreneurship – the example of smart contracts in the insurance market ». *Journal of Work-Applied Management* 13, n° 1 (1 janvier 2020): 69-87. <https://doi.org/10.1108/JWAM-08-2020-0037>.

²⁵⁴ Egberts, Alexander. « The Oracle Problem - An Analysis of How Blockchain Oracles Undermine the Advantages of Decentralized Ledger Systems ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 12 décembre 2017. <https://papers.ssrn.com/abstract=3382343>.

Even if one or several nodes are shut down, the Blockchain will continue working. This advantage however is annulled when a dependence on Oracles is established. The Oracle reintroduces the single point of failure in regard to all interactions based on its informational input.(...) Alternatively, the Oracle itself could present a point of failure, either by tampering with the information before putting it on the Blockchain or, in the case of Human Oracles, by providing a wrong answer to begin with.

²⁵⁵ Tsankov, Alexander. « The “Oracle Problem” Isn’t a Problem, and Why Smart contracts Makes Insurance Better for Everyone. » Medium (blog), 21 juin 2018. <https://antsankov.medium.com/the-oracle-problem-isnt-a-problem-and-why-smart-contracts-makes-insurance-better-for-everyone-8c979f09851c>.

The framing of the “oracle problem” as a problem for smart contracts is similar to saying that the speed of light is a problem for space travel, i.e. it’s a fact of reality that makes certain things impossible, but doesn’t disqualify the endeavor entirely.(...) The oracle problem is just a statement that, at some point, you need to trust some outside source to accurately give input as to what the state of reality is.

B - Une condition provenant de la *blockchain*

70. Exemples d'évènements *on-chain*. Les conditions déclenchant un transfert de jetons doivent donc être préférablement issues de la *blockchain*. Parmi ces événements *on-chain* déclencheurs de transferts nous pouvons trouver :

- les opérations arithmétiques déjà évoquées plus haut : en effet, celles-ci peuvent être effectuées par un *smart contract* et donc être lisibles dans la *blockchain*. (Ré)imaginons qu'une clause de versement de dividendes soit exécutée à l'aide de *smart contract*. Le transfert de jetons stables à un actionnaire sera notamment conditionnée au fait que celui-ci ait un solde en jetons-actions de l'organisation supérieur à 0. Cette opération arithmétique ($\text{nombreActionsDeLactionnaire} > 0$) est une information qui peut être lue dans la *blockchain* ;
- les conditions temporelles. La date est une information qui peut être récupérée dans une *blockchain*²⁵⁶. Il est donc possible de coder un transfert de jetons dépendant d'une condition temporelle, sans quitter cette infrastructure. Exemple : payer une partie après que ce soit écoulé un certain délai.

71. La nécessité des oracles. Réalistiquement cependant, il y a peu d'évènements déclenchant des transferts d'actifs qui proviennent uniquement de la *blockchain* : les actions opérationnelles prescrites dans des contrats dépendent très souvent d'évènements du monde réel. Le paiement d'un salarié, d'un prestataire de service, d'un vendeur dépendent d'évènements dont l'évaluation de la réalisation ne peuvent se faire qu'après observation du monde réel²⁵⁷. Même le transfert de dividendes dont nous avons précédemment parlé nécessite comme condition additionnelle

²⁵⁶ Il est en effet possible de connaître le temps/la date dans une *blockchain* dans les informations contenues dans un bloc de transactions. Lorsqu'un mineur ou validateur inclut son bloc dans la chaîne, la date à laquelle ce bloc a été miné/validé est stockée dans une donnée qui peut être récupéré par les développeurs de *smart contract*.

"The timestamp or timestamp is a small data stored in each block as a unique serial and whose main function is to determine the exact moment in which the block has been mined and validated by the blockchain network."

José Maldonado. « What Is Timestamp on Blockchain? - Bit2Me Academy ». Consulté le 28 février 2023. <https://academy.bit2me.com/en/blockchain-timestamp/>.

²⁵⁷ *"However, for smart contracts to realize upwards of 90% of their potential use cases, they must connect to the outside world. For example, financial smart contracts need market information to determine settlements, insurance smart contracts need IoT and web data to make decisions on policy payouts, trade finance contracts need trade documents and digital signatures to know when to release payments...."*

Chainlink. « What Is the Blockchain Oracle Problem? L Chainlink ». Chainlink Blog, 27 août 2020. <https://blog.chain.link/what-is-the-blockchain-oracle-problem/>.

de transfert l'information sur les bénéfices de la société ; cette donnée est accessible dans des bases sur internet, en dehors de la *blockchain a priori*²⁵⁸. Cette situation explique que ce soit développée une industrie autour du service d'oracles : des programmes ou individus chargés d'approvisionner des smart contracts en informations issues de l'extérieur d'une *blockchain*. Même si leur recours revient à tomber dans l'hybridation critiquée précédemment (un oracle est un intermédiaire pouvant être compromis, défaillant, payant et fermé), leur utilisation est quasiment indispensable et ne porte pas un gros préjudice à l'opportunité de recours au *smart contract* si leurs propriétés approchent celles des événements *on-chain*. C'est-à-dire que l'information fournie par l'oracle est décentralisée, transparente et ouverte²⁵⁹.

72. Conclusion du chapitre I. En définitive, même si d'autres processus peuvent être choisis par des parties afin d'être automatisés à l'aide de *smart contract*, ceux exploitant le mieux les propriétés de la *blockchain* nous apparaissent être les transferts d'actifs matérialisables en jetons. Idéalement, ces transferts doivent être déclenchés par des conditions objectives, produites et évaluables par des programmes provenant de la *blockchain*. Ce seront les clauses et les contrats composés de ces processus qui formeront alors d'excellents candidats pour être transformés en contrats intelligents.

²⁵⁸ Sauf dans l'hypothèse où toute l'activité de la société serait réalisée dans la *blockchain*. Dans ce cas on peut imaginer que les données financières de l'entreprise soient quérables *on-chain*.

²⁵⁹ Egberts, Alexander. « The Oracle Problem - An Analysis of How Blockchain Oracles Undermine the Advantages of Decentralized Ledger Systems ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 12 décembre 2017. <https://papers.ssrn.com/abstract=3382343>.

Ultimately, the Oracle Problem arises with the centralization of data feeds. Therefore, the only valuable approach to decreasing the dependence on data sources can be found in decentralizing the retrieval of data again.

Chapitre II - Les contrats sujets à une exécution dans la *blockchain*

73. **Panorama des clauses et contrats sujets à une exécution *on-chain*.** Les processus convenant le mieux à une exécution dans la *blockchain* étant identifiés, se pose la question de savoir quels sont les clauses et/ou contrats contenant ces dits processus et formant donc des bons candidats à une exécution par *smart contract*. Leur nombre est évidemment trop élevé pour tenter de les lister exhaustivement. De plus, de la même manière que d'autres processus que ceux décrits dans le chapitre précédent peuvent être opportunément exécutés *on-chain*, des contrats et clauses ne contenant pas ces derniers peuvent tout de même convenir à une exécution dans la *blockchain*. Toutefois, nous estimons que les parties trouveront dans les clauses et contrats composés de ces processus, un vivier important des conventions les plus adaptées à une exécution *on-chain*. Il est possible de les catégoriser entre celles essentiellement composées de transferts d'actifs (Section I) et celles partiellement composées de ces transferts (Section II).

Section I - Les contrats essentiellement constitués de transferts d'actifs

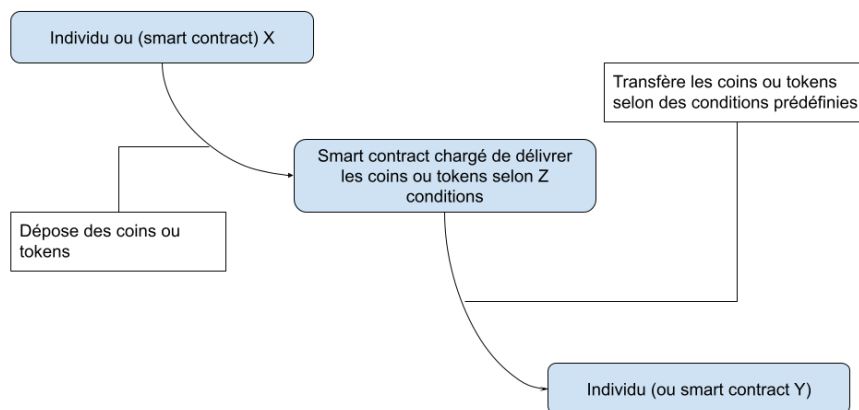
74. **Contrats en séquestre.** Les contrats mettant essentiellement en œuvre des transferts d'actifs conviennent particulièrement bien à une exécution *on-chain*. En général, ces contrats ont une structure peu complexe : leur objet consiste en des transferts d'actifs dépendant de conditions relativement simples. Il est possible de distinguer, parmi ces contrats, entre ceux ayant une architecture correspondant au mécanisme du séquestre, qui ressemble au fonctionnement d'un *smart contract* (§1), et les autres dont l'architecture est modélisable en séquestre (§2).

§ I - Les contrats aux opérations de séquestre

75. **La définition opérationnelle du séquestre.** Un *smart contract* a, en principe, un fonctionnement s'apparentant au mécanisme du séquestre ou de l'entiercement (*escrow*) : soit le fait de confier un ou plusieurs actifs à un tiers chargé de les garder et délivrer dans des conditions déterminées²⁶⁰. Lorsqu'il manipule des jetons, un *smart contract* consiste, en substance, en un

²⁶⁰ « Escrow », 22 mars 2023. <https://dictionary.cambridge.org/fr/dictionnaire/anglais/escrow>.

programme à qui sont confiés des actifs tokenisés, qui les relâche selon des conditions précodées²⁶¹.



Aussi, pour qu'un *smart contract* contrôle un actif, il faut d'abord qu'il lui soit "déposé", par un individu ou un autre *smart contract*. C'est à partir du moment où l'actif est "détenu" par le programme, qu'il est sous son contrôle ; et qu'il peut donc être relâché selon des conditions définies²⁶². Les contrats dans lesquels un tiers détient des actifs qu'il s'engage à délivrer selon certaines conditions à des parties, forment donc d'excellents candidats à une exécution dans la *blockchain* car leur architecture est déjà celle des smart contracts. Il est possible de proposer une liste des contrats ayant ce squelette et les classer entre ceux encadrant des opérations de dépôts (A) et ceux encadrant des opérations de garanties (B).

an agreement between two people or organizations in which money or property is kept by a third person or organization until a particular condition is met.

²⁶¹ Vitalik Buterin. « Ethereum Whitepaper ». ethereum.org. Consulté le 12 octobre 2021. <https://ethereum.org>.

Smart contracts [are] cryptographic "boxes" that contain value and only unlock it if certain conditions are met (...)

²⁶² Un *smart contract* ne peut pas, en effet, retirer des jetons du *wallet* d'un individu pour les transférer ensuite. Il est nécessaire de les lui avoir donné au préalable.

Kasireddy, Preethi. « How Does Ethereum Work, Anyway? » Medium (blog), 29 octobre 2019. <https://preethikasireddy.medium.com/how-does-ethereum-work-anyway-22d1df506369>.

Unlike externally owned accounts (les wallet), contract accounts (les smart contracts) can't initiate new transactions on their own.

A – Les contrats aux opérations de dépôts

76. Définition des contrats de dépôts. Nous appelons « contrats aux opérations de dépôts », les conventions dans lesquelles des parties déposent auprès d'un tiers un ou plusieurs actifs, afin que celui-ci les garde, puis les délivre selon les conditions spécifiées dans le contrat, à un ou plusieurs bénéficiaires. Sans qu'ils aient nécessairement la qualification juridique de dépôt²⁶³, les contrats de séquestre (a), de fiducie (b) et d'entiercement logiciel (c) figurent parmi ces conventions les mieux sujettes à être exécutées par *smart contract*.

a) La convention de séquestre

77. Définition juridique du séquestre. La définition juridique du séquestre est différente de celle du séquestre opérationnel que nous avons précédemment donnée²⁶⁴. L'article 1956 du code civil le définit comme *le dépôt fait par une ou plusieurs personnes, d'une chose contentieuse, entre les mains d'un tiers qui s'oblige de la rendre, après la contestation terminée, à la personne qui sera jugée devoir l'obtenir*. Il peut être autant conventionnel que judiciaire²⁶⁵.

Le contrat de séquestre correspond donc à une opération entre, d'une part, des parties qui sont en litige sur une chose et, d'autre part, un tiers désigné par le juge ou ces parties, qui est chargé de garder la chose et la rendre à la personne ayant obtenu gain de cause. Dans l'attente du résultat du litige, le tiers séquestre administre le bien selon ce qui est convenu au contrat. Enfin, il doit le rendre à la fin du contentieux sauf dans le cas où la volonté commune des parties l'en décharge avant ou pour une cause jugée légitime²⁶⁶.

78. Convenance du séquestre pour une exécution par *smart contract*. La logique opérationnelle du contrat de séquestre ressemble donc au fonctionnement basique d'un *smart contract* : des actifs sont confiés à une structure tierce qui les transfère ensuite à l'une ou l'autre des personnes l'ayant constitué en fonction d'un événement particulier. En l'occurrence, les actifs détenus par le

²⁶³ Article 1915 code civil : *Le dépôt, en général, est un acte par lequel on reçoit la chose d'autrui, à la charge de la garder et de la restituer en nature.*

²⁶⁴ V., *infra*, §75

²⁶⁵ Article 1955 du Code civil : *Le séquestre est ou conventionnel ou judiciaire.*

²⁶⁶ Article 1960 du Code civil : *Le dépositaire chargé du séquestre ne peut être déchargé avant la contestation terminée, que du consentement de toutes les parties intéressées, ou pour une cause jugée légitime.*

tiers dans un contrat de séquestre peuvent être mobiliers ou immobiliers²⁶⁷ et l'événement déclencheur de leur transfert est, sauf exceptions, la fin du litige portant sur cette chose²⁶⁸.

Il est ainsi retrouvé plusieurs des critères que doivent contenir les processus sujets à une exécution *on-chain*:

- un transfert d'actifs. Pour préserver l'opportunité de transformation en *smart contract*, les parties seront avisées de sélectionner des contrats de séquestre portant sur des actifs immatériels tokenisables, comme des actions ou de la monnaie²⁶⁹;
- un événement objectif déclencheur du transfert. Ici, il s'agit de la fin du litige et de la désignation de la partie gagnante. Cet événement est peu susceptible de discussion sur l'évaluation de sa réalisation : le prononcé du jugement indique clairement le gagnant et la fin du litige.

79. Séquestre exécuté par un *smart contract*. Il est donc peu surprenant de trouver le mécanisme de ce contrat déjà reproduit dans la *blockchain*. Le protocole d'arbitrage décentralisé Kleros²⁷⁰, par exemple, consiste en un ensemble de *smart contract* implémentant un mécanisme de séquestre juridique : lorsqu'un litige portant sur des jetons survient, ceux-ci peuvent être envoyés dans un *smart contract* de Kleros qui les tient en séquestre jusqu'à ce que l'un des arbitres de leurs protocoles intervienne, statue sur le litige et indique au *smart contract* à qui verser les jetons séquestrés²⁷¹. Nonobstant l'efficacité juridique de ce projet²⁷², son existence illustre l'intérêt

²⁶⁷ Article 1959 du Code civil : *Le séquestre peut avoir pour objet, non seulement des effets mobiliers, mais même des immeubles.*

²⁶⁸ Article 1960 du Code civil : *Le dépositaire chargé du séquestre ne peut être déchargé avant la contestation terminée, que du consentement de toutes les parties intéressées, ou pour une cause jugée légitime.*

²⁶⁹ V., *infra*, §51

²⁷⁰ <https://kleros.io/>

²⁷¹ Bergolla, Luis, Karen Seif, et Can Eken. « Kleros: A Socio-Legal Case Study Of Decentralized Justice & Blockchain Arbitration ». SSRN Scholarly Paper. Rochester, NY, 30 juillet 2021. <https://doi.org/10.2139/ssrn.3918485>.

Technically anyone with an Ethereum wallet can be a party to a Kleros dispute. By registering their transaction on Kleros' Escrow Dapp, contracting parties opt-in to Kleros as the chosen dispute resolution forum in their smart-contract. The opt-in process requires placing the contract funds into an escrow account that Kleros controls.

²⁷² Favreau, Amélie. « « Justice distribuée par une blockchain » et procédure civile ». Dalloz IP/IT, n° 01 (26 janvier 2022), p 24.

d'automatiser un contrat de séquestre dans la *blockchain*.

Des parties pourraient donc tout à fait faire recours à un *smart contract* pour exécuter une véritable convention de séquestre portant sur des actifs tokenisables comme de l'argent ou des titres financiers²⁷³. Dans ce cas de figure, le *smart contract* pourrait agir comme l'outil du tiers séquestre. Le programme serait en charge de :

- la réception des actifs, en lieu et place d'institutions financières comme la caisse des dépôts et consignations²⁷⁴, s'il s'agit de fonds monétaires ;
- leur gestion ;
- et leur délivrance, sur information du tiers séquestre.

Il n'est ni possible légalement et ni souhaitable qu'un *smart contract* se substitue complètement au tiers du contrat de séquestre. L'article 1956 du code civil impose, en effet, que ce dernier soit une personne juridique, *s'obligeant à rendre la chose*²⁷⁵. En sus, il vaut mieux que ce soit une personne humaine qui soit chargée d'informer le *smart contract* de l'issue du contentieux et qui garde un certain contrôle sur son fonctionnement, même si théoriquement l'information pourrait être automatiquement quérable sur internet²⁷⁶.

Ajoutons que la sentence arbitrale doit être écrite (mentions obligatoires), secrète et signée des arbitres. Le consensus technique (preuve de travail ou d'enjeu) aboutissant à la décision distribuée arbitrale qui s'opère par une simple transaction automatisée (smart contract) ne permet pas de respecter ces exigences et la sanction est la nullité (C. pr. civ., art. 1480 et 1483).

²⁷³ Mekki, Mustapha. « Le contrat, objet des smart contracts (Partie 1) ». Dalloz IP/IT, 2018, 409.

La force du smart contract réside dans son automaticité en dehors de toute intervention humaine. Par le protocole informatique des smart contracts, les parties peuvent automatiser un transfert de fonds après que des documents aient été remis, des faits établis ou des actes accomplis, l'ensemble étant authentifié au moyen de la technologie blockchain. On perçoit ici l'effet disruptif sur certaines activités telles que celles de séquestre.

²⁷⁴ Article L518-17 du Code monétaire et financier : *La Caisse des dépôts et consignations est chargée de recevoir les consignations de toute nature, en numéraire ou en titres financiers, prévues par une disposition législative ou réglementaire ou ordonnées soit par une décision de justice soit par une décision administrative.*

²⁷⁵ Article 1956 du Code civil : *Le séquestre conventionnel est le dépôt fait par une ou plusieurs personnes, d'une chose contentieuse, entre les mains d'un tiers qui s'oblige de la rendre, après la contestation terminée, à la personne qui sera jugée devoir l'obtenir.*

²⁷⁶ Il serait, en effet, possible de coder un logiciel chargé d'extraire des informations d'une décision de justice ou d'arbitrage accessible en ligne.

Cette limitation du rôle du *smart contract* ne nuirait pas pour autant à l'opportunité de son recours. Dans le cas où ce dernier serait chargé de la réception, gestion et délivrance de sommes d'argent, par exemple, il remplacerait totalement le rôle d'une institution financière. Ce qui aurait pour effet de supprimer tous les coûts habituellement dépensés pour son service, en plus de rationaliser sensiblement les processus : constitution dématérialisée et presque instantanée du séquestre, dépôts et retraits sans délai à tout moment, de n'importe où, etc. Le tiers séquestre serait, en fait, chargé de la seule tâche à *haute valeur ajoutée*²⁷⁷ de son mandat : celle d'indiquer à quel moment et vers qui les actifs litigieux doivent être remis.

b) Le contrat de fiducie

80. Définition de la fiducie. La fiducie est définie à l'article 2011 du code civil comme *l'opération par laquelle un ou plusieurs constituants transfèrent des biens, des droits ou des sûretés, ou un ensemble de biens, de droits ou de sûretés, présents ou futurs, à un ou plusieurs fiduciaires qui, les tenant séparés de leur patrimoine propre, agissent dans un but déterminé au profit d'un ou de plusieurs bénéficiaires*. Les mécanismes mis en œuvre par la fiducie et le séquestre sont donc semblables : des actifs sont confiés à un tiers chargé de les administrer et délivrer selon des conditions particulières.

81. Différences de la fiducie avec le séquestre. Mais ils diffèrent sur des points importants. Dans une fiducie, l'actif subit un véritable transfert de propriété dans le patrimoine d'affectation du fiduciaire²⁷⁸, contrairement au séquestre²⁷⁹. Autre différence, seules certaines

²⁷⁷ Il s'agit d'une expression régulièrement employée dans le contexte d'une innovation technologique qui a pour effet d'effectuer à la place d'un professionnel certaines de ses tâches. Généralement le programme ne permet de faire que les tâches les moins sophistiquées du professionnel pour justement le laisser concentrer sur celles les plus importantes, qui sont dites des activités à *haute valeur ajoutée*.

Muriel Féraud-Courtin. « Legaltech : derrière la technologie, l'Homme ! », La lettre des juristes d'Affaires, 24 janvier 2020. <http://www.lja.fr/point-de-vue-d-expert/legaltech-derriere-la-technologie-lhomme-545715.php/?latest>.

Le fastidieux et le chronophage doivent être pris en charge par la technologie pour laisser place à des missions à plus forte valeur ajoutée : conseil, prévision, analyse ou encore management de projet.

²⁷⁸ Lequel est distinct de son patrimoine personnel.

Article 2011 du code civil : *La fiducie est l'opération par laquelle un ou plusieurs constituants transfèrent des biens(...) à un ou plusieurs fiduciaires qui, les tenant séparés de leur patrimoine propre ...*

²⁷⁹ Cour de Cassation, Chambre civile 1, 30 septembre 2015, 14-21.111, Publié au bulletin

(...) 3°/ que le contrat de séquestre n'emporte pas transfert de la propriété de la chose séquestrée au séquestre ; qu'en jugeant que la Carpa était propriétaire des fonds déposés sur le compte séquestre du bâtonnier sans s'interroger, comme elle y était pourtant invitée, sur la nature particulière du contrat de séquestre en exécution duquel le bâtonnier avait reçu les fonds déposés, faisant obstacle au transfert de leur propriété au bâtonnier puis à la Carpa, la cour d'appel a privé sa

catégories de personnes ont droit d'être fiduciaires, alors que le séquestre est ouvert à tout le monde²⁸⁰ : notamment les établissements bancaires, les assurances, les sociétés de gestion de portefeuille et les avocats²⁸¹. Enfin, la condition de délivrance de l'actif dans la fiducie est arbitraire : il est possible de la constituer en attente de la résolution d'un litige, mais également pour de nombreuses autres choses. Une fiducie peut être constituée :

- pour organiser une transmission : un constituant transfère ses biens à un fiduciaire qui est chargé de les administrer et les remettre ensuite à un ayant-droit/bénéficiaire de la fiducie²⁸² ;
- comme outil de gestion : un constituant transmet un ou plusieurs biens, pendant une durée déterminée, au fiduciaire pour qu'il les gère à son bénéfice ou au bénéfice d'un tiers²⁸³ ;
- à titre de sûreté²⁸⁴, sur lequel nous reviendrons plus en détail²⁸⁵ ;
- pour sécuriser des transactions : lorsque la somme de la vente est très élevée, un fiduciaire peut intervenir comme l'agent tripartite sécurisant l'échange²⁸⁶.

décision de base légale au regard de l'article 1956 du code civil ;

²⁸⁰ Article 1956 du code civil : *Le séquestre conventionnel est le dépôt fait par une ou plusieurs personnes, d'une chose contentieuse, entre les mains d'un tiers qui s'oblige de la rendre, après la contestation terminée, à la personne qui sera jugée devoir l'obtenir.*

²⁸¹ Article 2015 du code civil : *Seuls peuvent avoir la qualité de fiduciaires les établissements de crédit mentionnés au I de l'article L. 511-1 du code monétaire et financier, les institutions et services énumérés à l'article L. 518-1 du même code, les entreprises d'investissement mentionnées à l'article L. 531-4 du même code, les sociétés de gestion de portefeuille ainsi que les entreprises d'assurance régies par l'article L. 310-1 du code des assurances. Les membres de la profession d'avocat peuvent également avoir la qualité de fiduciaire.*

²⁸² Article 2019 du code civil, alinéa 3 : *La transmission des droits résultant du contrat de fiducie et, si le bénéficiaire n'est pas désigné dans le contrat de fiducie, sa désignation ultérieure doivent, à peine de nullité, donner lieu à un acte écrit enregistré dans les mêmes conditions.*

²⁸³ Bénédicte FRANÇOIS. « Fiducie – Constitution de la fiducie », Répertoire des sociétés, septembre 2011.

La fiducie-gestion permet au constituant de faire gérer, de façon autonome, un ou plusieurs biens ou droits en les extrayant de son patrimoine pour les soumettre aux pouvoirs du fiduciaire. Le bénéficiaire est alors le constituant lui-même (C. civ., art. 2016), le fiduciaire devant, à terme, lui rétrocéder le patrimoine transféré. Ainsi le constituant transfère hors de son patrimoine certains biens qui sont gérés pour son bénéfice.

²⁸⁴ Article 2372-1 du code civil : *La propriété d'un bien mobilier ou d'un droit peut être cédée à titre de garantie d'une obligation en vertu d'un contrat de fiducie conclu en application des articles 2011 à 2030.*

²⁸⁵ V., *supra*, §98

²⁸⁶ Arnold Rouah. « La fiducie au service du M&A ». Village de la Justice (blog), 21 novembre 2015. <https://www.village-justice.com/articles/fiducie-service,40749.html>.

82. Convenance de la fiducie pour une exécution par smart contract. L'opération économique derrière la fiducie est probablement celle ressemblant le plus au fonctionnement d'un smart contract. Elle est, en effet, l'implémentation française du *trust*²⁸⁷, qui a la même logique que l'*escrow*²⁸⁸ (séquestre en anglais), et qui est le fonctionnement des smart contracts²⁸⁹. Elle est donc une candidate naturelle à une formalisation *on-chain* car elle comporte plusieurs des critères idéaux : les transferts peuvent porter sur des actifs tokenisables comme des sommes d'argent ou des titres financiers non cotés, sur lesquels des contrats de fiducie sont régulièrement constitués²⁹⁰, et les événements déclencheurs des transferts, ou de clôture de la fiducie, peuvent être objectifs, *on-chain* et produits par des programmes.

Ainsi, dans la fiducie à titre de gestion, la condition de délivrance des actifs est souvent temporelle.²⁹¹ Dans ce cas-là, le transfert d'actifs peut se faire sur la base d'une condition objective, et *on-chain*²⁹². Dans une fiducie à titre de transmission, l'événement déclencheur du transfert d'actif est un décès. Il s'agit d'une condition objective et quérable sur des bases de données²⁹³. Enfin, dans une fiducie constituée à titre de garantie, la condition déclenchant la remise de l'actif peut aussi être *on-chain* et

La fiducie, mécanisme très souple au service des praticiens, a vocation à accompagner de nombreuses situations de la vie des affaires ou des patrimoines. Parmi ces situations, le transfert d'actions ou de parts sociales peut utilement recourir à la fiducie pour sécuriser l'exécution de cessions impliquant un complément de prix (earn-out) ou une promesse de vente à terme (call option).

²⁸⁷ Reinhard Dammann et Vasile Rotaru. « La fiducie et le trust : une concurrence inégale », Recueil Dalloz, 2018, 1763.

Ainsi, pendant deux siècles, le droit français s'est montré réticent, pour des raisons essentiellement doctrinales, à l'idée d'introduire une institution analogue au trust. Ce n'est que la loi n° 2007-211 du 19 février 2007, complétée par la loi n° 2008-776 du 4 août 2008, qui a franchi le pas.

²⁸⁸ Escrow, American Trust. « What's the Difference Between an Escrow Account and a Trust Account? » American Trust Escrow (blog), 20 avril 2018. <https://americantrustescrow.com/2018/04/20/whats-the-difference-between-an-escrow-account-and-a-trust-account/>.

Although very different, a trust account operates in a similar way to an escrow account (...). Both operate like a bank account that holds funds to be dispersed to a designated party upon the completion or fulfillment of an agreement.

²⁸⁹ V., *infra*, §75

²⁹⁰ Virginie Corbet-Picard. « Fiducie sur titres ou sur les actifs sociaux ». Option Finance, La lettre des fusions-acquisitions et du private equity, 26 mars 2021. <https://www.optionfinance.fr/lettres-professionnelles/la-lettre-des-fusions-acquisition-et-du-private-equity/garanties-et-operations-de-fusion-acquisition/fiducie-sur-titres-ou-sur-les-actifs-sociaux.html>.

²⁹¹ Lorsque par exemple, un contrat prévoit qu'un fiduciaire gérera des titres financiers pour une certaine durée.

²⁹² V., *infra*, §61

²⁹³ V., *infra*, §66

objective (nous verrons comment²⁹⁴).

83. Fiducie exécuté par un smart contract. A l'instar du contrat de séquestre, des parties pourraient donc recourir à un *smart contract* pour automatiser la réception, gestion et délivrance des actifs placés en fiducie. Le programme agirait aussi comme l'outil du tiers fiduciaire. Il serait déployé et contrôlé par celui-ci, qui aurait pour seule tâche d'informer le *smart contract* de la réalisation ou non de la condition de délivrance de l'actif. Même dans ce rôle cantonné, le *smart contract* permettrait de rationaliser le processus de la fiducie et renforcer sa garantie d'exécution.

c) L'entiercement logiciel

84. Définition de l'entiercement logiciel. L'entiercement logiciel est la convention par laquelle un prestataire-fournisseur d'un logiciel confie les éléments techniques de celui-ci, comme le code source et les clefs d'accès, à un tiers afin qu'il les délivre à un client-bénéficiaire, généralement en cas de liquidation judiciaire du prestataire²⁹⁵. Le but de l'opération est de garantir au client-utilisateur d'un logiciel que, même en cas de disparition de son fournisseur, il dispose de la possibilité d'assurer lui-même, ou par un tiers, la continuité du service, grâce aux éléments techniques séquestrés²⁹⁶.

85. Convenance de l'entiercement logiciel pour une exécution par smart contract. Un entiercement logiciel consiste donc en un séquestre des éléments d'un logiciel, avec pour condition de délivrance la liquidation judiciaire du fournisseur de ce logiciel²⁹⁷. Le processus est donc un

²⁹⁴ V., *supra*, §98

²⁹⁵ APP - Agence pour la Protection des Programmes. « Entiercement - Gérer ses entiercements avec l'APP ». Consulté le 27 mars 2023. <https://www.app.asso.fr/nos-solutions/escrow-agreement>.

L'entiercement est un concept anglo-saxon (escrow agreement) qui consiste, pour le fournisseur d'un produit ou d'un service, à confier à un tiers séquestre des éléments essentiels (logiciels, bases de données, documents, etc.) à l'usage de ce produit ou à la réalisation de ce service. L'objectif est d'assurer à un tiers (client, partenaire, etc.) la possibilité d'y accéder, selon les dispositions prévues entre les parties, et notamment en cas de défaillance du fournisseur.

²⁹⁶ Ibid.

Lorsqu'un client investit dans une technologie ou un savoir-faire afin de l'utiliser pour les besoins de son activité, il n'a pas, le plus souvent, la maîtrise de la solution utilisée. Or, un lien de dépendance fort se crée au profit du prestataire qui vend cette solution ou tout au moins la met à disposition. L'entiercement permet de pallier cette difficulté en prévoyant un accès encadré aux éléments déposés. Il assure au client la possibilité de continuer à utiliser et le cas échéant de maintenir un produit ou service essentiel à son activité, en cas de défaillance du fournisseur (...)

²⁹⁷ Il s'agit d'une des conditions principales de délivrance d'un contrat d'entiercement logiciel ; mais le tiers séquestre

transfert d'actifs et l'évènement déclencheur de celui-ci est objectif et facilement quérable en ligne²⁹⁸. Toutefois les actifs ne résident pas *on-chain* : ici il s'agit de fichiers, certes immatériels, mais qui ne peuvent pas être stockés dans la *blockchain*²⁹⁹.

86. Entiercement logiciel exécuté par un *smart contract*. Un contrat d'entiercement logiciel pourrait donc être exécuté dans la *blockchain* de la manière suivante : les éléments du logiciel seraient stockés et chiffrés dans une solution de stockage décentralisée³⁰⁰; et un *smart contract* serait créé et connecté à ces derniers. Il prévoirait qu'en cas de liquidation judiciaire du fournisseur, le bénéficiaire serait automatiquement autorisé à déchiffrer et télécharger les éléments du logiciel.

Comme pour le contrat de séquestre et la fiducie, le *smart contract* pourrait assister la personne physique ou morale instituée en tant que séquestre : il déchiffrerait les éléments du logiciel seulement sur instruction du tiers. Cela est d'autant plus nécessaire qu'en pratique, les conditions de délivrance des éléments du logiciel dans un contrat d'entiercement peuvent être bien moins objectives que la liquidation judiciaire du prestataire : il peut s'agir de son absence de réponse aux sollicitations pendant un certain délai, de signes inquiétants de défaillance financière, etc³⁰¹. Toutefois, si les parties souhaitent ouvrir l'accès qu'en cas de procédure collective, les programmes peuvent totalement automatiser le processus du contrat : un logiciel peut être chargé de récupérer l'information sur la liquidation judiciaire, puis informer le *smart contract* qui autorisera à son tour le déchiffrement et téléchargement des éléments du logiciel. Dans cette hypothèse, il restera toutefois un élément

peut relâcher/délivrer ses éléments également en cas de défaut de maintenance.

André R. Bertrand. « Chapitre 202 – Logiciels », Dalloz action Droit d'auteur, 2010.

Enfin, il n'y a pas un jour où un distributeur de logiciels n'annonce son intention de ne plus maintenir certains de ses produits. Chaque jour, les utilisateurs de systèmes informatiques risquent donc d'être brutalement informés que, pour une raison ou une autre, leurs logiciels ne seront plus maintenus. Dans cette hypothèse, l'accès au programme source séquestré leur permettra de corriger, ou de faire corriger, ou de faire évoluer les logiciels pour lesquels ils auront légalement acquis un droit d'utilisation.

²⁹⁸ V., *infra*, §66

²⁹⁹ Nous aborderons ce sujet en détail plus-bas. Disons simplement qu'une *blockchain* n'est pas faite pour stocker des données lourdes, ce qui comprend les fichiers/documents qu'on trouve typiquement dans un ordinateur. Néanmoins, des solutions de stockage décentralisées, au fonctionnement similaire à ceux d'une *blockchain*, constituent des infrastructures faites pour les héberger.

³⁰⁰ V., *supra*, §415

³⁰¹ « Software Escrow: Ready for Quick Recovery | Codekeeper ». Consulté le 19 avril 2023.
<https://codekeeper.co/software-escrow.html>.

How prepared are you for the worst possible scenarios ? Scenario 1 Your supplier got sued and instantly went bankrupt. You lost thousands of lines of code! Scenario 2 You go into the office as usual, to find all your systems are locked by a ransomware attack. Scenario 3 Your developer decides he wants more money than agreed and holds your software hostage.

centralisant dans le processus : l'information sur la procédure collective. Fort heureusement, il existe de nombreuses sources en ligne mettant à disposition cette information³⁰², ce qui permettra aux parties de décentraliser, dans une certaine mesure, l'acheminement de cette information.

B – Les contrats aux opérations de garanties

87. Définition des contrats aux opérations de garanties. Au côté des contrats aux opérations de dépôts, se trouve ceux que nous nommons les contrats aux opérations de garanties. Ils consistent en des conventions mettant en œuvre des opérations dans lesquelles un tiers détient des actifs et les transfère à des bénéficiaires, à titre de garantie. Parmi ces contrats, nous pouvons compter le crédit-documentaire (a), la garantie autonome (b), le contrat de cautionnement (c) et la fiducie-sûreté (d).

a) Crédit documentaire

88. Définition du crédit documentaire. Le crédit documentaire, ou la lettre de crédit, est un engagement d'une banque, ou d'un autre tiers, en faveur d'un acheteur, à payer son vendeur dès lors que ce dernier lui aura remis certains documents justificatifs. Il s'agit d'une forme de garantie très utilisée dans le commerce international³⁰³, qui assure à l'importateur d'une marchandise de ne la payer que si la preuve de son exportation a été fournie à sa banque ; tandis que l'exportateur bénéficie de la garantie d'être payé par l'établissement bancaire de son client, dès lors qu'il fournit la preuve de l'expédition du bien.

Le mécanisme du crédit documentaire est donc tripartite et ressemble à celui d'une fiducie aux fins de sécurisation d'une transaction³⁰⁴ : un tiers détient une somme d'argent et la verse à un vendeur seulement lorsque ce dernier lui aura présenté des documents d'expéditions jugés conformes, comme

³⁰² Ici, il est pensé aux nombreux sites internet répertoriant des informations sur les sociétés comme infogreffe.com, pappers.fr, societe.com...

³⁰³ Gérard Hirigoyen, Alain Couret, et Jean Devèze. Lamy Droit du financement. Wolters Kluwers, 2022. Partie 7 - Titre 3 - Chapitre 3 - §5292

Le crédit documentaire est un moyen et une garantie de paiement créé par la pratique bancaire pour surmonter la méfiance entre partenaires commerciaux éloignés et n'ayant pas l'habitude de travailler ensemble.

³⁰⁴ V., *infra*, §81

un connaissance³⁰⁵. Généralement, les sommes versées par le tiers-bancaire ne proviennent pas de l'acheteur, et ne sont donc pas techniquement séquestrées, mais constituent une ligne de crédit de la banque en sa faveur³⁰⁶. Toutefois dans certaines formes de crédits documentaires, le prix de la marchandise est déjà payé par l'acheteur ; de sorte que le banquier ne se contente que de manipuler la somme que lui a fournie l'acheteur, comme dans un séquestre ou une fiducie³⁰⁷.

89. Différentes formes de crédits documentaires. Différentes formes de crédits documentaires peuvent être recensées :

- les crédits révocables et irrévocables, qui sont des lettres de crédits pouvant être annulées à tout moment par la banque ou au contraire constitutives d'engagements irrévocables pour elles³⁰⁸;
- les crédits avec paiement à vue³⁰⁹ ou différé³¹⁰ ;

³⁰⁵ Debard, Thierry, et Serge Guinchard. Lexique des termes juridiques 2020-2021 - 28e ed. Edition 2020-2021. Dalloz, 2020.

Titre de transport maritime de marchandises. Délivré par le représentant du transporteur, ce document constitue la preuve de la remise des marchandises à bord.

³⁰⁶ « Crédit documentaire - Fiches d'orientation - juillet 2022 | Dalloz ». Consulté le 28 mars 2023.

[Crédit documentaire :] Opération de banque, constitutive d'un crédit par signature, s'appliquant à des marchandises en voie d'acheminement, par laquelle l'acheteur (le donneur d'ordre) fait ouvrir un crédit permettant au vendeur (le bénéficiaire) d'obtenir, en échange du dessaisissement de documents justificatifs, représentatifs de la marchandise, le prix de celle-ci auprès d'une banque, après qu'elle a vérifié lesdits documents.

³⁰⁷ Julie YOUNG. « Understanding Fully Funded Documentary Letters of Credit (FFDLC) ». Investopedia (blog), 27 janvier 2023. <https://www.investopedia.com/terms/f/ffdlc.asp>.

A fully funded documentary letter of credit (FFDLC) is a documented letter of credit that serves as a written promise of payment provided by a buyer to a seller. With a fully funded letter of credit, the buyer's funds for the required payment are held in a separate account for use when needed, similar to the process for escrow.

³⁰⁸ Gérard Hirigoyen, Alain Couret, et Jean Devèze. « Lamy Droit du financement ». Wolters Kluwers, 2022. Partie 7 - Titre 3 - Chapitre 3 - §5299

Le plus souvent la banque s'engage personnellement à l'égard du bénéficiaire. Pour autant que les documents stipulés soient remis et les conditions du crédit respectées, le banquier est tenu de manière ferme, directe et autonome.

³⁰⁹ Gérard Hirigoyen, Alain Couret, et Jean Devèze. « Lamy Droit du financement ». Wolters Kluwers, 2022. Partie 7 - Titre 3 - Chapitre 3 - §5302

Paieement à vue : Le paieement du bénéficiaire par le banquier intervient contre remise des documents immédiatement après leur vérification.

³¹⁰ Ibid.

- les crédits transférables ou non³¹¹, c'est-à-dire ouverts pour un seul vendeur-bénéficiaire ou cessibles à d'autres;
- confirmés ou non³¹²; les crédits confirmés signifient qu'une deuxième banque est venue en garantie d'une première. Par exemple, la banque du vendeur garantit celle de l'acheteur-donneur d'ordre.

90. **Convenance du crédit documentaire pour une exécution par *smart contract*.**

Lorsqu'elle est couverte, le mécanisme de la lettre de crédit est donc opérationnellement le même que celui d'une fiducie; ce qui fait d'elle une autre excellente candidate à une exécution par *smart contract*. En effet, elle consiste en un tiers bancaire séquestrant une somme d'argent, qu'il versera à l'expéditeur, si celui-ci fournit la preuve de l'envoi de la marchandise. La condition déclenchante du transfert des fonds est particulièrement objective: ce n'est ni la satisfaction de l'acheteur ou la bonne conformité de la marchandise qui déclenche le paiement mais seulement la preuve documentaire, par l'exportateur, de son envoi.

En pratique cependant, il demeure souvent des disputes sur l'évaluation de la réalisation de cette condition; l'examen de la documentation à fournir peut être très rigoureux³¹³. Pour accroître l'objectivité de cette condition, diverses méthodes sont à la disposition des parties:

- des entreprises proposent des logiciels permettant d'analyser automatiquement les documents,

Paiement différé: L'engagement du banquier est à terme; le banquier paie à l'expiration d'un certain délai suivant, notamment, la date du document de transport. Concrètement, le paiement n'a souvent lieu qu'après réception des marchandises par le donneur d'ordre qui est alors tenté de s'opposer au paiement si la livraison ne lui convient pas.

³¹¹ Gérard Hirigoyen, Alain Couret, et Jean Devèze. « Lamy Droit du financement ». Wolters Kluwers, 2022. §5303

Le crédit peut être ouvert au profit d'un seul bénéficiaire désigné. Il peut également être transférable, à condition qu'il soit expressément qualifié de tel par la banque émettrice.

³¹² Article 2; Chambre de commerce internationale. Règles et usances uniformes de l'ICC relatives aux crédits documentaires: entrée en vigueur 1er juillet 2007, révision 2007. ICC Publications, 2007.

Confirmation signifie un engagement ferme de la banque confirmante, s'ajoutant à celui de la banque émettrice, d'honorer ou de négocier une présentation conforme.

³¹³ Larson, Dakota A. « Mitigating Risky Business: Modernizing Letters of Credit with Blockchain, Smart contracts, and the Internet of Things ». Law Review 2018, n° 4 (1 janvier 2019). <https://hcommons.org/deposits/item/hc:36187/>.

A letter of credit often specifies that the beneficiary must include an invoice, a bill of lading, an insurance certificate, and possibly a certificate of weight or quality.

comme les connaissances, afin de déterminer leur authenticité et conformité au contrat³¹⁴ ;

- il est possible de directement constituer le prestataire chargé du transport de la marchandise comme l'oracle informant du déclenchement du transfert des sommes ;
- d'autres ont avancé l'idée d'utiliser un GPS connecté au bien qui indiquerait de manière fiable sa position géographique ; le paiement serait déclenché dès lors qu'il aurait franchi une frontière³¹⁵.

En tout état de cause, la condition ne sera donc jamais *on-chain* et très décentralisée.

91. Crédit documentaire exécuté par un *smart contract*. Il n'y a pas de mécanisme de crédit documentaire reproduit dans le milieu *blockchain* à notre connaissance, même si nous faisons état d'expérimentations explorant ce cas d'usage.³¹⁶ Malgré cela, compte tenu de ce que nous venons d'évoquer, l'exécution *on-chain* du crédit documentaire ne poserait pas de grandes difficultés. A l'instar du séquestre et de la fiducie, le *smart contract* pourrait servir d'outil assistant le tiers bancaire. Il lui servirait à réceptionner les fonds pour ensuite les transférer à l'exportateur. Le tiers bancaire pourrait s'assister d'un oracle programmé pour informer le *smart contract* de la possibilité de transférer les sommes à l'exportateur : en connectant son *smart contract* à l'API de l'entreprise de transport, ou en utilisant un GPS connecté, etc. En tout état de cause, le recours à un *smart contract* et un oracle rationaliserait considérablement le processus de la lettre de crédit, même en tant que simple outil assistant un tiers-bancaire.

L'immutabilité du *smart contract* garantirait si bien le versement de la somme, que des parties

³¹⁴ Kiran Challapalli. « Trade Finance Workflow Automation Using AI ». IBM Digital Transformation Blog (blog), 5 mai 2021. <https://www.ibm.com/blogs/digital-transformation/in-en/blog/trade-finance-workflow-automation-using-ai/>.

It is not uncommon for the Banks to have thousands of people manually processing thousands of trade documents associated with each deal (...). There is a need for a smarter solution. An Artificial Intelligence (AI) powered solution can be exactly that!

³¹⁵ Emmanuel Netter. « II - L'idéal de rigueur dans l'exécution des contrats - Droit et numérique ». Consulté le 28 mars 2023. <https://enetter.fr/le-contrat/section-1-droit-commun-des-contrats/ii-lideal-de-rigueur-dans-lexecution-des-contrats/>.

Ainsi, dans le commerce maritime, plutôt que d'en passer par une banque à qui l'on enverra par fax ou email un document attestant d'une livraison par un navire, on pourrait équiper les conteneurs transportés de puces GPS, et déclencher le règlement par une blockchain dès que les coordonnées spatiales du port de destination sont atteintes.

³¹⁶ Par exemple, ce répertoire Github dans lequel un développeur met à disposition le code d'un prototype de lettre de crédit exécuté par un *smart contract* : <https://github.com/SabaunT/letter-of-credit>

pourraient choisir de se passer de la banque en tant que tiers. En effet, si cette dernière est l'intermédiaire par défaut dans une lettre de crédit, c'est en raison du fait qu'elle est une institution particulièrement fiable pour les parties : il y a peu de tiers présentant des risques aussi faibles de se défaire de leur engagement. Mais si un *smart contract* formalise le mécanisme d'une lettre de crédit couverte, la garantie d'exécution qu'il fournit pourrait suffire aux parties et les pousser à se passer de la banque, ce qui est déjà pratiqué³¹⁷.

b) La garantie autonome

92. Définition d'une garantie autonome. L'article 2321 du code civil définit une garantie autonome comme *un engagement par lequel le garant s'oblige, en considération d'une obligation souscrite par un tiers, à verser une somme soit à première demande, soit suivant des modalités convenues*. En général, l'opération mise en œuvre par le contrat est celle-ci : un donneur d'ordre demande à une banque de s'engager à verser une somme convenue à son créancier, dès lors que ce dernier en fera la demande³¹⁸. Le versement se fait en considération d'un contrat entre le donneur d'ordre et son créancier, mais la garantie de la banque est juridiquement autonome de ce rapport d'obligation³¹⁹.

En effet, bien que cette description fait apparaître la garantie autonome comme une caution³²⁰, elle en diffère par le fait que le bénéficiaire de la garantie n'est, en principe, soumis à aucun préalable pour activer le garant, d'où son autre nom de *garantie à première demande*. L'obligation du garant

³¹⁷ Jones, Stephen A. « Letter of Credit Non-Bank Issuer ». In *The Trade and Receivables Finance Companion: A Collection of Case Studies and Solutions*, édité par Stephen A. Jones, 95-103. Cham: Springer International Publishing, 2019. https://doi.org/10.1007/978-3-030-25139-0_7.

³¹⁸ « Garantie autonome - Fiches d'orientation - juillet 2022 | Dalloz ». Consulté le 28 mars 2023.

Le garant, en général un banquier, s'engage à la demande du donneur d'ordre, débiteur au titre du contrat de base (contrat de construction, par exemple), à payer le bénéficiaire, créancier du donneur d'ordre, une certaine somme d'argent dont le montant est déterminé à l'avance. Une fois le paiement effectué sur première demande – on parle ainsi parfois de garantie à première demande –, le garant, qui a en réalité consenti un crédit par signature au donneur d'ordre, dispose contre ce dernier d'un recours en remboursement de son avance.

³¹⁹ « Chapitre 312 - Garanties conventionnelles », Dalloz action - Droit et pratique des baux commerciaux, 2022 2021. 312.41

(...) ces garanties ont, comme le cautionnement, pour objet de faire payer les dettes du débiteur par un tiers, mais elles présentent l'avantage de constituer un contrat extérieur à la relation créancier-débiteur, et de ne pas être l'accessoire de l'obligation principale.

³²⁰ V., *supra*, §95

est autonome du contrat entre le donneur d'ordre et le bénéficiaire. Il ne peut opposer des exceptions ou vices issus de leur rapport³²¹ et se retrouve engagé dès lors qu'il est actionné, sauf en cas d'abus, fraudes manifestes ou collusion³²².

93. Convenance d'une garantie autonome pour une exécution par *smart contract*³²³.

Plusieurs aspects de la garantie à première demande attestent de sa convenance pour une exécution par *smart contract*. D'abord, la garantie consiste en une somme d'argent, qui est donc un actif tokenisable³²⁴. Ensuite elle est activée, par principe, dès lors que le bénéficiaire en fait la demande. Il s'agit d'une condition de déclenchement de transfert simple et objective. Parfois la demande peut être enfermée dans un certain délai. L'événement déclencheur du transfert peut donc être très objectif et *on-chain* : l'actionnement, dans le *smart contract*, de la garantie dans le délai prévu contractuellement.

94. Garantie autonome exécutée par un *smart contract*³²⁵. Le garant pourrait placer les sommes éventuelles à déboursier dans un *smart contract*, qui les tiendrait en séquestre. Le programme prévoirait qu'en cas d'actionnement du bénéficiaire dans le délai imparti, ce dernier recevrait la somme garantie. Le *smart contract* agirait alors comme l'outil opérationnel du garant. Là encore, comme pour le crédit documentaire, les propriétés du *smart contract* pourraient permettre à des tiers de se substituer à la banque dans le rôle de garant³²⁶.

c) Contrats de cautionnement

95. Définition du contrat de cautionnement. Le contrat de cautionnement est le contrat dans lequel une personne, la caution, s'oblige envers un créancier à payer la dette de son débiteur, en

³²¹ Article 2321 du code civil alinéa 3 : *Le garant ne peut opposer aucune exception tenant à l'obligation garantie.*

³²² Article 2321 du code civil alinéa 2 : *Le garant n'est pas tenu en cas d'abus ou de fraude manifestes du bénéficiaire ou de collusion de celui-ci avec le donneur d'ordre.*

³²³ Emmanuel Netter. « II - L'idéal de rigueur dans l'exécution des contrats - Droit et numérique ». Consulté le 28 mars 2023. <https://enetter.fr/le-contrat/section-1-droit-commun-des-contrats/ii-lideal-de-rigueur-dans-lexecution-des-contrats/>.

Les conventions les plus propres à être doublées d'un smart contract sont celles qui présentent, avant même le recours à l'informatique, le plus haut degré d'automatisme et d'élémentaire brutalité dans leur exécution. Les deux exemples les plus évidents sont alors le crédit documentaire et sa cousine, la garantie autonome à première demande.

³²⁴ V., *infra*, §51

³²⁵ Abdoulaye DIALLO. « Smart contract d'une garantie autonome au sens de l'article 2321 du code civil. » Village de la Justice (blog), 18 novembre 2018. <https://www.village-justice.com/articles/smart-contract-une-garantie-autonome-sens-article-2321-code-civil,29924.html>.

³²⁶ V., *infra*, §91

cas de défaillance de ce dernier³²⁷. Il est fait une distinction entre le cautionnement dit *simple* et *solidaire*³²⁸. Dans ce premier type de cautionnement, la caution peut exiger que le créancier discute préalablement des biens du débiteur et divise ses poursuites entre les éventuelles autres cautions³²⁹. Tandis que dans la caution solidaire, le créancier peut solliciter le garant dès la première défaillance du débiteur et lui demander de payer la part entière de la dette³³⁰. Contrairement à la garantie autonome, la caution est dépendante de l'obligation principale garantie : la nullité de cette dernière entraîne celle de la caution³³¹, elle-même ne saurait être engagée dans des conditions plus onéreuses que le débiteur principal³³², et enfin elle peut opposer au créancier toutes les exceptions personnelles ou inhérentes à la dette, qui appartiennent au débiteur³³³.

96. Convenance du cautionnement pour une exécution par *smart contract*.

L'opération mise en œuvre par un contrat de cautionnement consiste en un transfert de sommes d'argent, de la part de la caution, à raison de la réalisation d'un événement objectif et possiblement *on-chain* : la défaillance du débiteur. La condition est objective puisqu'elle consiste en l'appréciation de la réalisation ou non de l'obligation du débiteur : si l'obligation n'est pas exécutée par le débiteur, alors la caution est activée, sinon elle ne l'est pas. La condition peut également être *on-chain*, nous le verrons, si l'obligation est réalisée dans la *blockchain*³³⁴.

Ces éléments font donc du contrat de cautionnement, en théorie, une convention propice à une exécution par *smart contract*. Plus particulièrement le contrat de cautionnement solidaire, qui a une nature de fonctionnement encore plus automatique : activation de la caution dès la défaillance, possibilité de demander à l'une quelconque des cautions de payer l'intégralité de la dette. Néanmoins

³²⁷ Article 2288 du code civil : *Le cautionnement est le contrat par lequel une caution s'oblige envers le créancier à payer la dette du débiteur en cas de défaillance de celui-ci.*

³²⁸ Article 2290 du code civil alinéa 1 : *Le cautionnement est simple ou solidaire.*

³²⁹ Article 2305 du code civil alinéa 1 : *Le bénéfice de discussion permet à la caution d'obliger le créancier à poursuivre d'abord le débiteur principal.*

³³⁰ Article 2305 du code civil alinéa 2 : *Ne peut se prévaloir de ce bénéfice ni la caution tenue solidairement avec le débiteur, ni celle qui a renoncé à ce bénéfice, non plus que la caution judiciaire.*

³³¹ Article 2293 du code civil alinéa 1 : *Le cautionnement ne peut exister que sur une obligation valable.*

³³² Article 2296 du code civil : *Le cautionnement ne peut excéder ce qui est dû par le débiteur ni être contracté sous des conditions plus onéreuses, sous peine d'être réduit à la mesure de l'obligation garantie.*

³³³ Article 2298 du code civil : *La caution peut opposer au créancier toutes les exceptions, personnelles ou inhérentes à la dette, qui appartiennent au débiteur, sous réserve des dispositions du deuxième alinéa de l'article 2293.*

³³⁴ V., *supra*, §103

il faut noter que le cautionnement, même solidaire, a un fonctionnement par nature beaucoup moins automatique que celui de la garantie autonome. La caution a beaucoup plus de possibilités de contester le bien-fondé de son actionnement, malgré la défaillance du débiteur : elle peut arguer de la nullité du contrat ou de l'irrespect de son formalisme.

97. Contrat de cautionnement exécuté par un *smart contract*. Tout comme la garantie autonome, les parties souhaitant automatiser un contrat de cautionnement dans la *blockchain* pourront préalablement constituer le *smart contract* de l'obligation principale et le connecter à celui de la sûreté personnelle. Autrement dit, si le cautionnement garantit par exemple un contrat de prêt, le *smart contract* de ce dernier sera déployé dans la *blockchain* et la défaillance du prêteur pourra activer automatiquement la caution ; pareillement, si la caution garantit une dette locative³³⁵. Le *smart contract* tiendra en séquestre la somme garantie, et si la dette, de prêt ou de loyer, n'est pas payée à l'échéance, alors la somme séquestrée sera transférée au créancier.

d) Fiducie-sûreté

98. Définition de la fiducie-sûreté. La fiducie-sûreté est une forme de fiducie constituée aux fins de garantir un créancier contre la défaillance de son débiteur³³⁶. Dans la pratique, elle est très utilisée pour prémunir un prêteur contre l'incapacité de l'emprunteur à payer sa dette³³⁷. En effet, le contrat opère le transfert de propriété des biens de l'emprunteur dans le patrimoine affecté du tiers fiduciaire, qui les remettra au créancier si le débiteur n'a pas payé sa dette selon les modalités prévues au contrat. Cette sûreté est particulièrement efficace car les biens remis au fiduciaire résident dans son patrimoine propre ; à l'abri des créanciers du débiteur lors d'une éventuelle procédure collective³³⁸.

³³⁵ Elodie POULIQUEN. « Le cautionnement relatif à un bail d'habitation n'est pas soumis aux dispositions du Code de la consommation ! », La revue des Loyers, n° 1027 (1 mai 2022).

³³⁶ Vernières, Christophe. Guide de la rédaction des actes notariés: Actes courants - Immobilier, Famille - Patrimoine, Entreprise, Rural. Paris-La Défense: DEFRENOIS, 2022.

La fiducie est l'opération par laquelle un constituant transfère, à titre de garantie d'une créance, des biens, droits ou sûretés dont il est propriétaire, dans un patrimoine d'affectation géré par un fiduciaire.

³³⁷ Michel Collet et Alexandre Bordenave. « Prêteurs et emprunteurs : la fiducie, c'est maintenant ! » CMS Francis LEFEBVRE (blog), 22 octobre 2020.

Bien maîtrisée, le recours à la fiducie comme sûreté, c'est-à-dire pour garantir une dette du «constituant» (ou de son groupe) en transférant des biens ou droits dans un patrimoine fiduciaire, dans des contextes tendus est le plus souvent privilégié.

³³⁸ Ibid.

99. Convenance de la fiducie-sûreté pour une exécution par *smart contract*. Nous avons déjà évoqué comment la fiducie est un contrat dont les processus se prêtent bien à une exécution *on-chain*³³⁹. Les actifs garantis peuvent être matérialisés par des jetons et représenter des biens immatériels comme des créances ou des droits de propriété intellectuelle. La condition déclenchante du transfert d'actif peut avoir lieu dans la *blockchain* si l'obligation garantie est exécutée par des *smart contracts*. En effet, lorsque la fiducie-sûreté est constituée pour garantir un prêt, nous avons déjà évoqué que ce dernier peut être réalisé *on-chain*. Or si c'est le cas, l'information du non-remboursement du prêt peut être quérable dans la même infrastructure où réside le *smart contract* de la fiducie-sûreté.

100. Fiducie-sûreté exécutée par un *smart contract*. Classiquement le *smart contract*, sous l'administration d'un fiduciaire, tiendra en séquestre les actifs tokenisés. Lorsque le prêt ne sera pas remboursé à échéance dans le *smart contract*, l'actif immatériel séquestré pourra être automatiquement versé au créancier.

§ III - Les contrats modélisables en séquestre

101. Contrats n'ayant pas un fonctionnement de base en séquestre. D'autres contrats, qui consistent essentiellement en des transferts d'actifs aux conditions simples, forment d'excellents candidats à une exécution *on-chain*, malgré qu'ils n'aient pas leur structure de base en séquestre. Cette caractéristique ne nuit pas pour autant à l'opportunité de les exécuter par *smart contract* car ils peuvent tout même être modélisés en séquestre, sans que cela bouleverse leur nature. Nous pouvons diviser ces contrats entre ceux encadrant des opérations du domaine de la finance (A) et ceux encadrant d'autres opérations simples (B).

La fiducie résiste à la faillite du constituant. En effet, les biens ou droits transférés à la fiducie étant sortis du patrimoine de ce dernier, l'ouverture d'une procédure collective à son égard n'empêche normalement pas le bénéficiaire de réaliser la fiducie-sûreté.

³³⁹ V., *infra*, §82

A – Les contrats encadrant des opérations financières

102. Contrats de la finance. Parmi les contrats encadrant des opérations financières, mais n'ayant pas une structure en séquestre, nous pouvons trouver le contrat de prêt de sommes d'argent (a), le contrat de nantissement (b), le contrat d'assurance paramétrique (c) et les contrats dérivés (d).

a) Le contrat de prêt de sommes d'argent

103. Définition du contrat de prêt de sommes d'argent. L'article 1874 du code civil distingue le contrat de prêt à consommation du contrat de prêt à usage³⁴⁰. Le contrat de prêt à usage est défini comme *le contrat par lequel l'une des parties livre une chose à l'autre pour s'en servir, à charge pour l'emprunteur de la rendre après s'en être servi*³⁴¹. Le prêt de consommation est défini comme *le contrat par lequel l'une des parties livre à l'autre une certaine quantité de choses qui se consomment par l'usage, à charge pour cette dernière de lui en rendre autant de même espèce et qualité*.³⁴² Le prêt peut donc porter sur des objets matériels ou immatériels, consommables à l'usage ou non.

Dans le cas qui nous intéresse, le prêt est un prêt d'argent, donc un prêt de consommation³⁴³ consistant à mettre à disposition d'un emprunteur une somme, que celui-ci utilisera et restituera *en même espèce et qualité*, avec des intérêts, à une certaine date. Pour sécuriser ce prêt, le prêteur peut prévoir diverses garanties comme celles lui permettant de s'approprier un bien de l'emprunteur d'une valeur égale au montant emprunté, si ce dernier ne respecte pas son engagement³⁴⁴.

³⁴⁰ Article 1874 du code civil : *Il y a deux sortes de prêt : Celui des choses dont on peut user sans les détruire ; Et celui des choses qui se consomment par l'usage qu'on en fait. La première espèce s'appelle " prêt à usage ". La deuxième s'appelle " prêt de consommation ", ou simplement " prêt ".*

³⁴¹ Article 1875 du code civil : *Le prêt à usage est un contrat par lequel l'une des parties livre une chose à l'autre pour s'en servir, à la charge par le preneur de la rendre après s'en être servi.*

³⁴² Article 1892 du code civil : *Le prêt de consommation est un contrat par lequel l'une des parties livre à l'autre une certaine quantité de choses qui se consomment par l'usage, à la charge par cette dernière de lui en rendre autant de même espèce et qualité.*

³⁴³ Geneviève PIGNARRE. « Répertoire de droit civil - Prêt – Prêt de consommation - Chapitre 2 », janvier 2016.

L'origine du prêt de consommation est, on l'a vu, romaine. Il s'agit du mutuum qui permet à l'emprunteur de consommer la chose prêtée. Le code civil le traite de « simple prêt ». Forme de prêt par excellence, le prêt de consommation peut porter sur cette chose particulière qu'est l'argent.

³⁴⁴ Il s'agit d'une sûreté réelle.

Debard, Thierry, et Serge Guinchard. Lexique des termes juridiques 2020-2021 - 28e ed. Edition 2020-2021. Dalloz, 2020.

104. Convenance du contrat de prêt pour une exécution par *smart contract*. Le contrat de prêt de somme d'argent consiste opérationnellement en deux types de flux : le prêteur met à disposition des fonds à l'emprunteur, que ce dernier rembourse au prêteur avec des intérêts à l'échéance. Il s'agit donc de transferts d'actifs tokenisables dans une *blockchain*. La condition de remboursement d'un prêt est temporelle : le terme convenu³⁴⁵. Il s'agit donc d'une condition objective et *on-chain*. Cela fait du contrat de prêt, un contrat se prêtant bien à une exécution par *smart contract*. Enfin, l'opportunité de réaliser le *smart contract* d'un contrat de prêt, tient aussi au fait qu'il peut servir d'évènement déclencheur à d'autres contrats garanties *smart contractualisés*³⁴⁶.

105. Contrat de prêt exécuté par un *smart contract*. Comme pour la garantie autonome, il est possible de prévoir qu'un *smart contract* constitue une sorte de compte bancaire dans lequel son titulaire aurait préapprouvé le versement automatique de la somme à rembourser³⁴⁷. Dès lors que le terme de la date d'échéance serait échu, le *smart contract* transférerait cette somme au créancier. Mais dans le cas où il n'y aurait pas de solde, la défaillance de cette fonctionnalité pourrait déclencher l'actionnement du *smart contract* d'une sûreté comme une garantie autonome, une fiducie-sûreté, etc. Les articles 1892 et suivants du code civil n'imposent pas de conditions particulières quant aux modalités opérationnelles du prêt. Un prêt peut donc être réalisé dans une *blockchain* et se retrouver bien plus fiablement garanti qu'un contrat de prêt ordinaire de la manière qu'il vient d'être décrite³⁴⁸.

b) Nantissement

106. Définition du nantissement. D'après l'article 2355 du code civil, le nantissement est

[Sûreté réelle :] Lorsque certains biens du débiteur garantissent le paiement qu'en cas de défaillance, le produit de la vente de ces biens est remis au créancier bénéficiaire de la sûreté par préférence aux créanciers chirographaires.

³⁴⁵ Article 1899 du code civil : *Le prêteur ne peut pas redemander les choses prêtées avant le terme convenu.*

³⁴⁶ V., *infra*, §94

³⁴⁷ Andrew Beams, Catherine Gu, Srini Raghuraman, Mohsen Minaei, et Ranjit Kumaresan. « Visa Crypto Thought Leadership – Auto Payments ». Consulté le 2 avril 2023. <https://visa-signature.com/solutions/crypto/auto-payments-for-self-custodial-wallets.html>.

³⁴⁸ Capdeville, Jérôme Lasserre. « [Jurisprudence] De quelques précisions intéressant le bitcoin et le prêt de bitcoins ». La lettre juridique, mars 2020, 18 mars 2020. <https://www.lexbase.fr/article-juridique/57260658-jurisprudence-de-quelques-precisions-interessant-le-ibitcoin-i-et-le-pret-de-ibitcoins-i>.

Le bitcoin est un bien incorporel fongible et consommable. Les contrats de prêt de bitcoins sont alors des prêts de consommation. Le régime juridique de ces prêts a, par conséquent, vocation à s'appliquer en la matière, et notamment l'article 1902 du Code civil imposant à l'emprunteur de rendre les choses prêtées, en même quantité, et au terme convenu.

l'affectation, en garantie d'une obligation, d'un bien meuble incorporel ou d'un ensemble de biens meubles incorporels, présents ou futurs. Il s'agit d'un contrat solennel³⁴⁹. Contrairement au gage dont l'assiette porte sur les biens meubles corporels³⁵⁰, celle du nantissement porte sur des biens incorporels et peut comprendre des titres financiers, sociaux, des fonds de commerce ou des éléments de fonds de commerce, des créances et des droits de propriétés intellectuelles. En pratique, le nantissement garantit un prêt. Par exemple, une personne affecte en garantie d'une ligne de crédit auprès d'une banque des titres financiers³⁵¹. En cas de défaillance, la banque dispose de la faculté de saisir ces titres et de les vendre afin d'épurer la dette de son débiteur.

107. Convenance du nantissement pour une exécution par *smart contract*. Plusieurs éléments rendent la formalisation *on-chain* du nantissement opportune. Sa structure de flux d'actifs est simple : des biens en garantie sont transférés à un créancier si le débiteur est défaillant, c'est-à-dire s'il ne rembourse pas à échéance son emprunt. La condition déclenchant la saisie ensuite est possiblement objective et *on-chain* si l'obligation garantie est réalisée par *smart contract*.

Mais le nantissement n'implique pas de tiers : les biens nantis demeurent dans la possession du débiteur et c'est au moment de sa défaillance que le nantissement peut être réalisé. Malgré cela, l'opération peut être structurée de sorte à ce que les biens nantis soient conservés dans les mains d'un séquestre, afin de garantir son exécution. Cet élément abonde dans le fait que le nantissement peut être modelé dans un *smart contract* de séquestre³⁵². La convenance du nantissement pour une formalisation par *smart contract* est encore appuyée par le fait que la loi reconnaît explicitement la possibilité de l'exécuter en ayant recours à la *blockchain*. En effet, l'ordonnance autorisant

³⁴⁹ Article 2356 du code civil : *A peine de nullité, le nantissement de créance doit être conclu par écrit.*

³⁵⁰ Article 2333 du code civil : *Le gage est une convention par laquelle le constituant accorde à un créancier le droit de se faire payer par préférence à ses autres créanciers sur un bien mobilier ou un ensemble de biens mobiliers corporels, présents ou futurs.*

³⁵¹ Catherine Berlaud. « Garanties d'un prêt : nantissement et caution », Gazette du Palais, n° 41 (13 décembre 2022): P.24.

Cass. com., 30 nov. 2022, no 20-23554, M. X c/ Sté BNP Paribas, F-B (cassation CA Aix-en-Provence, 22 oct. 2020), M. Mollard, f.f. prés. ; SCP Waquet, Farge et Hazan, SCP Marc Lévis, av.

Une banque consent à une société un prêt in fine, destiné à financer partiellement l'acquisition d'actions, garanti par le nantissement des titres, objet du prêt, et par la cession de toutes les créances nées ou à naître au titre d'une promesse d'achat consentie par des sociétés tierces, débitrices cédées, dans le cadre d'un pacte d'actionnaires portant sur les actions cédées que la société emprunteuse détiendrait.

³⁵² Dominique LEGEAIS. Droit des sûretés et garanties du crédit. 15e éd. LGDJ, 2022.

l'inscription des titres financiers non cotés dans un dispositif électronique partagé³⁵³, régit aussi les modalités d'application du nantissement de compte titres sur celui-ci. Les articles D211-10 et suivants du code monétaire et financier, et en particulier l'article R211-14-1, décrivent les conditions devant être remplies lorsque sont nantis des titres représentés en jetons dans un dispositif électronique partagé^{354 355}.

108. Contrat de nantissement exécuté par un *smart contract*. Il est très courant de retrouver des mécanismes de nantissement exécutés sous forme de *smart contract*, dans le milieu de la *blockchain*. Le nantissement servant essentiellement à garantir un prêt, sa logique a été souvent mobilisée dans les smart contracts composant les protocoles *DeFi* proposant des services de prêt. Dans ceux-ci, les prêts sont quasiment tous garantis par des mécanismes de nantissement. Lorsqu'une personne souhaite emprunter des cryptomonnaies, elle doit préalablement mettre en garantie d'autres crypto-monnaies d'une valeur supérieure en moyenne 150% au montant emprunté. Si la valeur de l'actif affecté en garantie descend en dessous d'un seuil, celui-ci est automatiquement³⁵⁶ liquidé ou transféré au prêteur. Sans en avoir le nom, l'opération mise en œuvre est donc celle d'un nantissement : un actif immatériel garantit un prêt ; et démontre bien que les smart contracts peuvent automatiser une partie des processus de ces contrats³⁵⁷.

³⁵³ Ordonnance n° 2017-1674 du 8 décembre 2017 relative à l'utilisation d'un dispositif d'enregistrement électronique partagé pour la représentation et la transmission de titres financiers

³⁵⁴ Dispositions spécifiques au nantissement de titres inscrits dans un dispositif d'enregistrement électronique partagé (Article R211-14-1)

(...) II.-Pour l'application du IV de l'article L. 211-20, le créancier nanti définit avec le constituant du nantissement les conditions dans lesquelles ce dernier peut disposer des titres financiers nantis et des sommes en toute monnaie figurant dans le compte ouvert dans les livres d'un intermédiaire mentionné à l'article L. 211-3 ou d'un établissement de crédit mentionné au I.

³⁵⁵ Bruno Mathis. « Quel décret d'application pour les ordonnances blockchains ? », Lamy Droit de l'immatériel, n° 149 (1 juin 2018).

L'ordonnance n'interdit pas que la chaîne de blocs représente la propriété des titres avec des jetons. Ceux-ci observent le principe de non-double dépense : le contrôle de provision des titres chez le cédant est donc assuré par le protocole, sans besoin de développement additionnel. Ils se prêtent aussi à une fonction de séquestre, utile pour subordonner la livraison à une confirmation de la contrepartie, mais aussi pour procéder à un nantissement.

³⁵⁶ Il est pensé ici au protocole de prêt décentralisé MakerDAO.

Dominique LEGEAIS. « Marché financier - 3 QUESTIONS - De nouveaux développements pour la finance décentralisée - Veille par Dominique Legeais », La Semaine Juridique - Entreprise et affaires (JCP E), n° 52 (26 décembre 2019).

Dans le cas de MakerDAO, l'objet qui est mis en gage est un cryptoactif, et le prêt est exprimé dans une cryptomonnaie créée pour l'occasion et dont le cours ne fluctue pas, le DAI. L'entité qui conserve le cryptoactif mis en gage est un smart-contract appelé Collateralized Debt Position ou CDP, que seul le créancier a la possibilité de débloquent.

³⁵⁷ Pour preuve, il est possible de trouver de nombreuses descriptions de ces protocoles dans lesquelles le terme de nantissement est utilisé pour expliquer leur fonctionnement.

Ainsi, un véritable contrat de nantissement de compte titres peut être exécuté par *smart contract* dès lors qu'il respecte les dispositions du code monétaire et financier. Le *smart contract* peut être constitué pour jouer le rôle du compte-titre : il servirait à la fois de réceptacle des titres financiers représentés en jetons et de compte-séquestre saisissable par la banque lorsque les conditions du nantissement seraient réunies³⁵⁸. Mais les modalités de réalisation du nantissement imposées par le code monétaire et financier³⁵⁹ nécessiteront un certain contrôle sur le *smart contract* et donc une perte d'autonomie de celui-ci.

c) Assurance paramétrique

109. Définition de l'assurance paramétrique. Les contrats d'assurance paramétriques sont des produits d'assurances visant à garantir les assurés contre des risques très objectivement évaluable³⁶⁰ : par exemple le risque de sécheresse mesuré par le dépassement d'une certaine température³⁶¹, le retard d'un avion mesuré par son heure réelle d'arrivée en comparaison de son

« Maker : Le Pionnier de La Finance Décentralisée », 15 avril 2021.

<https://academy.youngplatform.com/fr/cryptomonnaies/maker-le-pionnier-de-la-finance-decentralisee/>.

Le protocole Maker permet d'obtenir des prêts en DAI de manière décentralisée, donc sans intermédiaire financier. Pour profiter de ce service, il est nécessaire de déposer un nantissement en cryptomonnaie dans un Maker Vault. Les Maker Vaults sont des smart contracts dédiés à la protection des nantissements des prêts DAI.

³⁵⁸ Dès lors qu'on conçoit que le nantissement de titres porte sur un compte, il faut se représenter/imaginer qu'un *smart contract* puisse représenter un compte dans lequel pourront entrer et sortir des titres.

Jean-François Adele et Didier Poracchia. « Nantissement de titres financiers enregistrés sur un DLT (blockchain) : reconnaissance d'une universalité fictive ». *Entreprise et expertise Juridique, Option Finance*, n° 1494 (21 janvier 2019).

La question se posait de savoir si ce dispositif de nantissement de titres en DLT produirait les mêmes effets que le régime de nantissement de compte-titres classique impliquant le nantissement "automatique" des titres venant en substitution ou en complément des titres initialement nantis, ainsi que de leurs fruits et produits. Bien que les dispositions du décret ne le précisent pas expressément, cette question doit recevoir une réponse positive.

³⁵⁹ Article D211-10 et suivants du code monétaire et financier.

³⁶⁰ Pierre-Grégoire Marly et Arnaud Sorel. « Assurance et nouvelles technologies - Les promesses de l'assurance paramétrique », *Responsabilité civile et assurances*, n° 3 (1 mars 2023).

(...) l'assurance paramétrique se signale par un mode original de règlement des sinistres : là où les assurances ordinaires de dommages subordonnent l'indemnisation de l'assuré à l'appréciation du préjudice qu'il a réellement subi, elle s'émancipe de celui-ci pour offrir un règlement à forfait dès qu'un seuil indiciel est atteint. En d'autres termes, le déclenchement comme le montant de ce règlement sont fonctions de critères objectifs et fixés d'avance.

³⁶¹ Roh, Jooyun. « Parametric Drought Insurance: Case Studies From Latin America & Europe ». Descartes. Consulté le 4 avril 2023. <https://www.descartesunderwriting.com/whitepaper/parametric-drought-insurance-case-studies-latin-america-europe/>.

heure prévue d'arrivée³⁶², la crue mesurée par le niveau de la mer³⁶³, etc. Lorsque ceux-ci sont réalisés, la compagnie d'assurance verse un montant forfaitaire prévu à l'avance au bénéficiaire, au lieu d'une indemnisation correspondant au dommage réellement subi. L'intérêt de ce type de produits est de fluidifier considérablement le processus de versement des indemnités d'assurances qui peuvent se déclencher de façon totalement automatisée³⁶⁴.

110. Convenance de l'assurance paramétrique pour une exécution par *smart contract*.

Ces contrats ont volontairement une nature automatique : ils mettent en place un transfert de sommes d'argent dépendant d'événements objectifs. Ils forment donc d'excellents candidats à une exécution par *smart contract*. Les actifs transférés sont aisément représentable en jetons et les événements sont objectifs car ils sont des données provenant de programmes : une température d'un thermomètre connecté, une API d'un aéroport pour l'heure d'arrivée d'un avion³⁶⁵, etc.

L'information sera en revanche rarement en provenance de la *blockchain*. Et elle nécessitera presque toujours la mise en place d'oracles qui la récupéreront de manière plus ou moins décentralisée. En effet, s'il s'agit d'une donnée très distribuée, l'assureur disposera de plusieurs sources d'informations qu'il pourra requêter afin de former un agrégat entre elles, et fournir au *smart contract* une donnée particulièrement fiable. En revanche pour des données ne provenant que d'une seule source, il faudra composer avec la faille de la centralisation³⁶⁶.

³⁶² Delphine CUNY. « Retard d'avion : Axa lance une assurance automatique sur la Blockchain ». La Tribune, 14 septembre 2017. <https://www.latribune.fr/entreprises-finance/banques-finance/retard-d-avion-axa-lance-une-assurance-automatique-sur-la-blockchain-750202.html>.

³⁶³ Chaperon, Meg. « Parametric Flood Insurance ». Descartes. Consulté le 4 avril 2023. <https://www.descartesunderwriting.com/whitepaper/parametric-flood-paper/>.

Descartes monitors the exposure with our new technologies, and when pre-agreed thresholds, such as river water levels, amount of rainfall, and wind speed are reached or exceeded, it directly leads to pay-out.

³⁶⁴ Pierre-Grégoire Marly et Arnaud Sorel. « Assurance et nouvelles technologies - Les promesses de l'assurance paramétrique », Responsabilité civile et assurances, n° 3 (1 mars 2023).

Ce modus operandi permet ainsi d'alléger considérablement la procédure d'indemnisation. Nul besoin d'exiger une déclaration de l'assuré ni l'éventuelle intervention d'un expert : l'identification et l'évaluation du sinistre s'accomplissent dans une même séquence, où l'événement déclencheur est instantanément décelé et, avec lui, le montant de la prestation automatiquement fixé selon le barème ou la formule établie dans la police.

³⁶⁵ Comme le propose depuis récemment LexisNexis qui utilise la solution d'oracle ChainLink.

Christopher Capot et Eric Lammerding. « LexisNexis Launches Flight Status Data Tracking Using Chainlink Node, Enabling Smart contract-Based Parametric Insurance Products », 29 juin 2022. <https://www.lexisnexis.com/community/pressroom/b/news/posts/lexisnexis-launches-flight-status-data-tracking-using-chainlink-node-enabling-smart-contract-based-parametric-insurance-products>.

³⁶⁶ V., *infra*, §71

111. Contrat d'assurance paramétrique exécuté par un smart contract. Un *smart contract* séquestre pourra facilement être constitué afin d'être le réceptacle du montant forfaitaire de l'indemnisation. Celui-ci sera versé au bénéficiaire dès lors que l'oracle aura fait parvenir l'information de la réalisation du risque assuré au *smart contract*. L'avantage de recourir à un *smart contract* pour automatiser ces conventions plutôt qu'une solution centralisée, est que le processus se retrouvera bien plus fiablement garanti. L'assuré aura également la certitude que son paiement sera réalisé si l'oracle informe bien le *smart contract* tandis que dans le cas d'une solution centralisée, il devra faire entièrement confiance à l'assurance³⁶⁷. Le but de ces produits étant aussi de garantir à l'assuré le versement de ces fonds, l'usage de *smart contract* renforce cette garantie de versement.

d) Les contrats dérivés

112. Définition d'un contrat dérivé. Un contrat dérivé est un type de contrat, extrêmement répandu dans le monde de la finance, dans lequel des parties s'entendent sur un échange d'actifs financiers à une date et un prix déterminés³⁶⁸. L'article D211-1 du code monétaire et financier liste ces contrats financiers, dont les plus populaires sont :

- les contrats d'option, dans lesquels une partie s'engage à vendre ou acheter à une autre un actif financier, qui peut être une action ou une matière première par exemple, à un prix, appelé « le prix d'exercice », majoré d'une prime, correspondant au prix de l'option, et une date déterminés. A la date d'exercice, le bénéficiaire de l'option a le choix entre lever l'option ou non³⁶⁹ ;

³⁶⁷ Courtecuisse, Matthieu, et Ronan Davit. « Les assurances paramétriques au cœur des smart contracts : une révolution pour l'assurance ». *Revue d'économie financière* 135, n° 3 (2019): 145-62. <https://doi.org/10.3917/ecofi.135.0145>.

Les avantages d'un smart contract [dans l'assurance paramétrique] sont évidents :- transparence dans les prestations servies définies ex ante et du niveau de primes associé à cette prestation ; - prestation servie plus rapidement à l'assuré en cas d'événement garanti ; - réduction des coûts de gestion ; - sécurisation des opérations d'assurance ; - gain d'image pour l'assureur.

³⁶⁸ Gilles Nejman. *Les Contrats de Produits Dérivés : Aspects Juridiques*. Larcier. Les Dossiers Du Journal Des Tribunaux, 1999.

Le produit dérivé est un actif financier consistant en un droit à terme ou conditionnel né d'un contrat ou d'une promesse de contrat, dont la valeur dépend de l'évolution de la valeur d'un ou de plusieurs actifs ou indices sous-jacents pendant la période séparant la conclusion du contrat de son dénouement.

³⁶⁹ Jean AULAGNIER, Laurent AYNÈS, et Jean-Pierre Bertrel. *Partie 2 - C à D - Étude 233 Contrats d'option - Section I - Présentation générale des contrats d'option - § 1 Définition – 233-5*. Lamy Patrimoine, 2015.

Une option est un droit pour son acquéreur d'acheter ou de vendre, selon le type d'option, un actif donné à un prix donné et, en fonction du mode d'exercice, à une échéance ou jusqu'à une échéance fixée à l'avance. En contrepartie, l'acquéreur de l'option verse, dès la conclusion du contrat, une prime au vendeur de l'option.

- les contrats à terme dans lesquels l'acheteur ou le vendeur prennent l'engagement ferme réciproque d'acheter ou céder un actif financier, sans possibilité d'option³⁷⁰ ;
- et les contrats d'échange, ou de *swap*, définis comme les contrats dans lesquels des parties s'accordent pour échanger un flux financier contre un autre, comme un taux d'intérêt, en fonction d'échéances et de conditions prévues à l'avance³⁷¹.

Les contrats dérivés peuvent être cédés sur des marchés réglementés, qui sont organisés par des intermédiaires régulateurs ou sur le marché dit “gré à gré”, c’est-à-dire directement entre un acheteur et un vendeur³⁷².

113. Convenance d'un contrat dérivé pour une exécution par *smart contract*. Plusieurs aspects des contrats dérivés plaident en faveur de leur convenance pour une exécution par *smart contract*³⁷³. D'abord, ils sont presque purement opérationnels dans la mesure où ils consistent essentiellement en des transferts d'actifs³⁷⁴. Ensuite, les actifs financiers objets des transactions sont

³⁷⁰ Jean AULAGNIER, Laurent AYNÈS, et Jean-Pierre Bertrel. Partie 2 - C à D - Étude 233 Contrats d'option - Section I - Présentation générale des contrats d'option - § 1 Définition – 233-6. Lamy Patrimoine, 2015.

Un contrat à terme correspond à un engagement ferme alors que le contrat d'option est un engagement conditionnel. L'acheteur de l'option n'est pas tenu d'exercer le contrat : il peut l'abandonner s'il juge qu'il n'est plus intéressant pour lui. Par contre le vendeur n'a pas cette possibilité : sa situation est soumise à la décision de l'acheteur.

³⁷¹ Jean DEVEZE. Partie 3 - Ressources de trésorerie de l'entreprise - Division 4 Couverture des risques de taux et conservation de la trésorerie - Chapitre 2 Swaps, FRA, caps, floors, collars, options de taux d'intérêt - Section 1 Swaps de taux d'intérêt - §1. Mécanisme de l'opération. Lamy Droit du financement, s. d.

Un swap de taux est une opération par laquelle deux parties ayant emprunté à des conditions différentes « acceptent de s'effectuer réciproquement des paiements équivalents aux montants des intérêts que l'autre partie doit à son prêteur initial ».

³⁷² Sébastien PRAICHEUX. « Instruments financiers à terme – Régime juridique des instruments financiers à terme ». Répertoire des sociétés, avril 2019.

Il s'ensuit notamment que les contrats financiers doivent être négociés sur un marché réglementé ou négociés de gré à gré et ne peuvent porter que sur certains sous-jacents.

³⁷³ Vitalik Buterin. « Ethereum Whitepaper ». ethereum.org. Consulté le 12 octobre 2021. <https://ethereum.org>.

Financial derivatives are the most common application of a "smart contract", and one of the simplest to implement in code.

³⁷⁴ ISDA. « ISDA Legal Guidelines For Smart Derivative Contracts : Introduction », janvier 2019. <https://www.isda.org/2019/01/30/legal-guidelines-for-smart-derivatives-contracts-introduction/>.

Derivatives are fertile territory for the application of smart contracts and distributed ledger technology (DLT) because their main payments and deliveries are heavily dependent on conditional logic.

d'excellents candidats à une tokenisation³⁷⁵. Enfin, les transactions sont déclenchées par des conditions très objectives : levée d'option dans un certain délai, ou à une certaine date ; bien que pas forcément produites *on-chain*³⁷⁶.

114. Contrat dérivé exécuté par un *smart contract*. Les contrats dérivés pourraient donc être formalisés en séquestre dans lesquels des smart contracts joueraient le même rôle que des intermédiaires dans les contrats de gré à gré qui sécurisent les transactions³⁷⁷. Dans le contrat d'option, par exemple, un actif est transféré à un bénéficiaire si celui-ci lève l'option dans un délai déterminé, ou à une date précise. Le vendeur pourrait déposer son actif dans un *smart contract* qui le tiendrait en séquestre, tandis que l'acheteur y déposerait également le prix de l'actif et le prix de l'option. Le programme délivrerait alors l'actif à l'acheteur que s'il actionne sa levée d'option dans le délai convenu et le vendeur pourrait, au même moment, récupérer le prix et la prime.

De nombreux protocoles DeFi proposent des smart contracts de produits dérivés³⁷⁸. Le milieu est si développé qu'on trouve même des produits n'ayant pas leur équivalent dans le monde réel³⁷⁹. Dès lors, il n'est pas étonnant que des acteurs institutionnels du monde des produits dérivés se soient penchés sur l'opportunité de recourir aux smart contracts pour automatiser ces contrats. Ce fut le cas de l'ISDA, par exemple, l'organisme de standardisation des pratiques de marchés de gré à gré sur les produits dérivés, qui a produit plusieurs articles détaillés sur la méthodologie à suivre pour formaliser des contrats dérivés³⁸⁰.

³⁷⁵ En particulier lorsqu'ils sont des actifs immatériels financiers (V., §51).

³⁷⁶ Fries, Christian, Peter Kohl-Landgraf, Björn Paffen, Stefanie Weddigen, Luca Del Re, Wilfried Schütte, David Bacher, et al. « Implementing a financial derivative as smart contract ». arXiv, 5 mars 2019. <http://arxiv.org/abs/1903.00067>.

Unlike traditional OTC derivatives, the valuation of the smart derivative contract is not based on each counterparty's internal valuation model but relies on an external valuation source, a so-called oracle. Thus, the underlying derivative's net present value is exogenously determined and contractually accepted by both parties. As a result of the oracle's valuation, the reset value is exchanged between the counterparties on each settlement date.

³⁷⁷ Chainlink Blog. « Bringing Trust to Derivatives Using Chainlink DeFi Smart contracts », 11 octobre 2019. <https://blog.chain.link/solving-deep-seated-trust-problems-in-derivatives-using-chainlink-enabled-smart-contracts/>.

A smart contract can digitally represent the operational clauses of a paper derivatives contract using boolean logic (if x happens, pay y). It holds funds in escrow as a custodian, executes the contract, keeps records of state changes, and redundantly stores the contract across the network with perfect uptime.

³⁷⁸ Par exemple : Synthetix, UMA, Hegic Option, Oryn, dYdX...

³⁷⁹ Comme par exemple, le concept d'options perpétuelles.

Leifke, Robert. « Perpetual Options for DeFi ». Numoen (blog), 20 septembre 2022. <https://medium.com/numoen/perpetual-options-for-defi-821351c0a24f>.

³⁸⁰ ISDA. « Smart contracts and Distributed Ledger – A Legal Perspective – International Swaps and Derivatives Association », août 2017. <https://www.isda.org/2017/08/03/smart-contracts-and-distributed-ledger-a-legal-perspective/>.

B – Les contrats encadrant d'autres opérations simples

115. Contrats simples. D'autres contrats, mettant en œuvre des opérations simples, mais qui n'ont pas une structure en séquestre, peuvent trouver un intérêt à être exécutés par des smart contracts. Parmi ceux-ci, peuvent se trouver le contrat de vente (a), le contrat de mandat de gestion (b), le contrat de niveau de services (c) et le contrat de louage de choses (d).

a) Le contrat de vente ou de cession

116. Définition du contrat de vente. Le contrat de vente est défini par l'article 1582 du code civil comme *la convention par laquelle un individu s'oblige à livrer une chose, et l'autre à la payer*. Une vente peut donc porter sur un bien matériel ou immatériel. Lorsqu'elle porte sur des droits personnels, le terme de cession est généralement employé³⁸¹. Les ventes peuvent avoir lieu sur place ou à distance. Dans ce dernier cas, elles seront encadrées par les articles L221-1 suivants du code de consommation lorsqu'elles ont lieu entre un particulier et un professionnel³⁸².

117. Convenance d'une vente pour une exécution par smart contract. Le mécanisme d'un contrat de vente consiste schématiquement en deux types de flux : le transfert du bien à l'acheteur, puis le transfert du prix du bien au vendeur. Comme évoqué, le bien peut être immatériel et à ce titre se prêter à une représentation en jeton. Dès lors, dans le cas d'une vente au schéma classique, il est constaté que le transfert d'actif peut également s'opérer selon des conditions objectives et *on-chain* : le bien, représenté par un NFT par exemple, est transféré à l'acheteur dès lors que celui-ci a payé son prix ; et le prix est transféré au vendeur dès le moment où l'acheteur est en possession du bien-NFT.

ISDA. « ISDA Legal Guidelines For Smart Derivative Contracts : Introduction », janvier 2019.
<https://www.isda.org/2019/01/30/legal-guidelines-for-smart-derivatives-contracts-introduction/>.

³⁸¹ Debard, Thierry, et Serge Guinchard. Lexique des termes juridiques 2020-2021 - 28e ed. Edition 2020-2021. Dalloz, 2020.

Lorsque le droit transféré est un droit personnel, on parle généralement de cession (exemple : cession de créance).

³⁸² Article L221-1 du code de la consommation : I. - Pour l'application du présent titre, sont considérés comme :

I° Contrat à distance : tout contrat conclu entre un professionnel et un consommateur, dans le cadre d'un système organisé de vente ou de prestation de services à distance, sans la présence physique simultanée du professionnel et du consommateur, par le recours exclusif à une ou plusieurs techniques de communication à distance jusqu'à la conclusion du contrat ; (...)

Si les actifs sont physiques, l'évènement déclencheur du transfert des sommes sera l'information de la prise de possession du bien par l'acheteur, ou du colis si la vente est à distance. Bien qu'il ne s'agisse pas d'une information *on-chain*, elle reste objective : le livreur peut être constitué en oracle informant que l'acheteur est en possession du bien³⁸³. Ce mécanisme fera perdre un peu des bénéfices de l'exécution par *smart contract* ; mais il faut noter qu'il existe, en pratique, des solutions pour neutraliser ce point de défaillance³⁸⁴.

118. Contrat de vente exécuté par un *smart contract*. Il existe de nombreux contrats de vente, portant souvent sur des NFT, exécutés complètement *on-chain* de la manière venant d'être décrite. Dans ces cas-là, les smart contracts jouent le rôle de tiers sécurisant la transaction : le bien est déposé dans le programme par le vendeur et le prix lui est également versé. Dès cet instant, si le montant versé est celui du prix convenu, l'acheteur pourra retirer le bien et le vendeur récupérer le prix³⁸⁵.

Un *smart contract* peut donc tout à fait être institué afin de sécuriser des contrats de vente. Il sera particulièrement utile pour les cessions portant sur des actifs matérialisables par des jetons : il est pensé aux cessions de créance³⁸⁶ et de licences³⁸⁷. Une difficulté pourra surgir du fait que le droit impose parfois un formalisme à respecter pour certaines cessions. Les parties pourront connecter leur smart contracts de sorte à ce que la délivrance de l'actif ne se fasse qu'après la complétion du formalisme légal³⁸⁸.

³⁸³ José, Fábio. « Building a Smart contract to Sell Goods ». Coinmonks (blog), 26 juin 2022. <https://medium.com/coinmonks/build-a-smart-contract-to-sell-goods-6cf73609d25>.

(...) 8. *The Courier, after delivery the order to the Retailer, marks the order as delivered on the Smart contract. The courier could be a robot, a drone. Think with me! Today we have many possibilities.* 9. *The Smart contract payout the Manufacturer for the order.*

³⁸⁴ Il est pensé, notamment, à des mécanismes d'arbitrage en ligne comme Kleros (kleros.io)

³⁸⁵ Il s'agit schématiquement de la manière dont fonctionne les ventes de NFT sur OpenSea.

³⁸⁶ Ce qui est rendu possible par la « tokenisation » des créances. Hou, Jing, Burak Kazaz, et Fasheng Xu. « Invoice Tokenization for Deep-Tier Payables Finance ». SSRN Scholarly Paper. Rochester, NY, 17 février 2023. <https://doi.org/10.2139/ssrn.4362566>.

³⁸⁷ Ce qui est rendu possible par la « tokenisation » des licences. Matulionyte, Rita. « Can Copyright Be Tokenized? » SSRN Scholarly Paper. Rochester, NY, 24 octobre 2019. <https://doi.org/10.2139/ssrn.3475214>.

³⁸⁸ V., *supra*, §451

b) Contrat de mandat de gestion

119. Définition du mandat de gestion. Le contrat de mandat de gestion est un contrat par lequel une personne, le mandant, donne à une autre personne, le mandataire, le pouvoir de gérer tout ou partie de son patrimoine, en son nom et pour son compte. Cette forme de contrat est celle encadrant l'opération dite de gestion de portefeuilles dans laquelle un individu confie à un tiers spécialiste le soin de gérer ses instruments financiers dans un objectif de rendement. Ce tiers peut être une banque, un conseiller de gestion patrimonial, une société privée de gestion de portefeuille, etc³⁸⁹

120. Convenance du mandat de gestion pour une exécution par smart contract. Les actifs dans un mandat de gestion financière sont le plus souvent immatériels³⁹⁰, et sont donc à ce titre aisément représentables par des jetons dans la *blockchain*³⁹¹. Ils sont, en premier lieu, mis à disposition du mandataire afin d'être gérés, puis éventuellement transférés vers d'autres lieux au cours de la gestion, et enfin restitués au mandant à la fin du mandat. Si la gestion consiste à investir les actifs dans des protocoles de finance décentralisée, alors les événements déclencheurs de ces flux sont tous quérables dans la *blockchain*.

121. Exécution d'un mandat de gestion par un smart contract. Les opérations de gestion financière sont déjà régulièrement mises en œuvre dans le milieu crypto. Il est, en effet, très courant que soient confiés à des smart contracts, des crypto-monnaies et/ou des NFT afin qu'ils soient gérés et génèrent des plus-values pour leurs propriétaires³⁹². Ces smart contracts peuvent gérer les actifs

³⁸⁹ AMF. « Le mandat de gestion ». Consulté le 11 avril 2023. <https://www.amf-france.org/fr/espace-epargnants/comprendre-les-produits-financiers/supports-dinvestissement/mandat-de-gestion>.

Le mandat de gestion est un contrat écrit sur papier ou autre support durable, par lequel un client (le mandant) donne pouvoir à un gérant (le mandataire) de gérer un portefeuille incluant un ou plusieurs instruments financiers (actions, obligations, fonds et sicav...), en fonction de ses objectifs d'investissement, y compris sa tolérance au risque, de ses connaissances et son expérience et de sa situation financière, y compris sa capacité à subir des pertes.

³⁹⁰ Ibid.

Sauf demande particulière du mandant, les catégories d'instruments financiers autorisés par l'AMF sont : - les instruments financiers négociés sur un marché réglementé (par exemple, en France, les actions cotées sur Euronext) ou sur un marché étranger de titres financiers reconnu, mentionné à l'article L. 423-I du code monétaire et financier; (...)

³⁹¹ V., *infra*, §51

³⁹² Directorate-General for Financial Stability, Financial Services and Capital Markets Union (European Commission). Decentralized Finance: Information Frictions and Public Policies: Approaching the Regulation and Supervision of Decentralized Finance. LU: Publications Office of the European Union, 2022. <https://data.europa.eu/doi/10.2874/444494>.

The main DeFi services are currently structured around the following products: Asset management services allowing customers to integrate pools of assets - so-called 'vaults' - governed and managed by predetermined rules encoded publicly into smart contracts. Major protocols include Set Protocol and PieDAO.

selon des stratégies pré-configurées ou être sous l'administration de véritables individus. Dans le cas d'un mandat de gestion financière, un *smart contract* pourra être constitué afin de réceptionner les actifs du mandant et gérer son investissement sous le contrôle du mandataire. Le mandant disposera de la faculté de retirer ses actifs, dans les modalités convenues en actionnant le *smart contract* ; tandis que le mandataire pourra être rémunéré par une commission prélevée automatiquement sur la plus-value réalisée.

Les avantages du recours à un *smart contract* pour la gestion seraient multiples :

- en premier lieu, les rendements dans les protocoles de finance décentralisée peuvent être bien plus intéressants que ceux trouvés dans la finance traditionnelle³⁹³,
- la gestion assistée d'un *smart contract* facilite le travail du mandataire à qui il peut donner une relative autonomie en codant les stratégies d'investissement³⁹⁴,
- la gestion hérite des propriétés de la *blockchain* : elle est auditable et le mandant peut garder toute souveraineté sur les actifs confiés³⁹⁵.

c) Contrat de niveau de services

122. Définition d'un contrat de niveau de services. Un accord de niveau de service, dit aussi *SLA* en anglais, est un contrat souvent prévu en annexe de celui de fourniture d'un logiciel

³⁹³ « Why Decentralised Finance (DeFi) Matters and the Policy Implications ». Paris: OCED, 19 janvier 2022.

Borrowing and lending rates in DeFi lending protocols are much higher to those available for traditional financial products and are driven by supply and demand dynamics for each of the crypto-asset transacted (...). Such high rates have been an important driver of DeFi activity especially in a prolonged low-rate environment and a consequent search for yield by retail and institutional investors.

³⁹⁴ Directorate-General for Financial Stability, Financial Services and Capital Markets Union (European Commission). Decentralized Finance: Information Frictions and Public Policies : Approaching the Regulation and Supervision of Decentralized Finance. LU: Publications Office of the European Union, 2022. <https://data.europa.eu/doi/10.2874/444494>.

Asset management services allowing customers to integrate pools of assets - so-called 'vaults' - governed and managed by predetermined rules encoded publicly into smart contracts.

³⁹⁵ Ibid.

Non-custodial services: Holders of crypto-assets in a DeFi process have full control over the treatment of their assets once they are associated with holders' public addresses. This feature contrasts with the traditional use of custodial services by financial intermediaries to manage their clients' portfolios.

SaaS³⁹⁶. Il vise à engager un prestataire sur des niveaux de services minimaux quant à la disponibilité, la performance et la qualité de service du logiciel mis à disposition. Concrètement ces conventions prévoient des indicateurs clés de performance, comme une obligation de 99% de disponibilité d'un logiciel, et si ceux-ci ne sont pas atteints, le prestataire s'obligera à payer une indemnité au client³⁹⁷. Même si ces contrats sont souvent utilisés pour les services informatiques, ils peuvent se retrouver dans d'autres domaines tels que les services de télécommunications³⁹⁸.

123. Convenance d'une convention de niveau de services pour une exécution par smart contract. Une convention de niveaux services prend opérationnellement la forme d'un transfert de sommes d'argent au client, si le prestataire n'atteint pas ce qu'on nomme un indicateur clé de performance (dits aussi *KPI* ³⁹⁹). Il s'agit donc d'un transfert d'actifs dépendant de conditions objectives. Les sommes d'argent sont représentables par des jetons et les conditions de leurs transferts, bien que se trouvant en dehors de la *blockchain*, sont objectives et produites par des

³⁹⁶ Vivant, Michel. Partie 6 - Guide - Titre 3 Comment négocier, rédiger et gérer les contrats et projets informatiques ? - Chapitre 2 Les grands types de contrats informatiques et leurs spécificités - Section 1 Les caractéristiques propres à certains types de contrats informatiques - § 5. CONTRATS ET « CLOUD COMPUTING ». Editions Lamy, 2020. <https://hal-sciencespo.archives-ouvertes.fr/hal-03397686>.

Ce contrat [de cloud computing] ayant pour objet la fourniture d'un service, il conviendra de préciser les niveaux de service attendus par le client tant en termes de performances (temps de réponse, temps de transmission des données...) qu'en termes de disponibilité des applications (horaires d'ouverture et de fermeture, périodes d'indisponibilité...). A ce titre, un certain nombre de seuils, de taux, d'indicateurs devront être fournis et précisés.

³⁹⁷ Thibault Verbiest. « Le Service Level Agreement dans les contrats informatiques ». Droit & Technologies (blog), 11 novembre 2003. <https://www.droit-technologie.org/actualites/le-service-level-agreement-dans-les-contrats-informatiques/>.

Le cœur du SLA réside dans les clauses fixant le niveau des services attendus, à savoir le taux de disponibilité et de fiabilité du service (par exemple : les heures et les jours pendant lesquels le service sera disponible). Sera également précisé le temps de réponse qui devra être octroyé au fournisseur en cas de plainte pour dysfonctionnement (par exemple, prévoir que 95% des problèmes seront résolus après une heure à compter de la plainte).

³⁹⁸ Décision n° 2014-1102 du 30 septembre 2014 portant sur la définition des marchés pertinents de la téléphonie fixe, la désignation d'opérateurs exerçant une influence significative sur ces marchés et les obligations imposées à ce titre - IV.2.7.2. Obligation de prendre des engagements de niveau de qualité de service. Consulté le 12 avril 2023. <https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000030136148>.

Afin de refléter l'importance particulière de la qualité de service comme facteur de compétitivité, notamment sur le segment non résidentiel, et permettre la reproduction des engagements de qualité de service (« service level agreement ») couramment pratiqués sur le marché de détail, en particulier par Orange, il est justifié et proportionné de demander des engagements contractuels renforcés de la part d'Orange, incluant un système de pénalités, notamment financières, suffisamment incitatifs à la fourniture d'une qualité de service satisfaisante. Ce système de pénalités doit également être équilibré et juste au regard de celui appliqué aux opérateurs clients en cas de manquement à leurs propres engagements contractuels auprès de leurs clients de détail.

³⁹⁹ Sisense. « Service Level Agreement (SLA) KPI ». Consulté le 12 avril 2023. <https://www.sisense.com/kpis/it-kpi/service-level-agreements-slals/>.

Service Level Agreement KPIs track how your organization is performing relative to the baseline services established in your company's SLA. The term can include a variety of smaller metrics that tie into an organization's SLA commitments.

programmes. Dès lors que les parties disposent d'un moyen fiable de mesurer ces événements⁴⁰⁰ et de les connecter à un *smart contract* alors, l'opération d'ensemble se prêtera bien à une exécution par *smart contract*⁴⁰¹.

124. Exécution par *smart contract* d'un contrat de niveaux de services. Les parties peuvent utiliser un *smart contract* afin qu'il soit chargé de tenir en séquestre les sommes à verser au client en cas d'irrespect des KPI. Dès lors que le *smart contract* est informé d'une violation de la convention de niveaux de services, le client pourrait récupérer la somme séquestrée dans la mesure convenue⁴⁰². L'inconvénient de ce schéma est que l'information de l'irrespect du KPI a des bonnes chances d'être mesurable et fournie par le prestataire, ce qui pose un problème de neutralité. Les parties peuvent tenter de solutionner cet aspect en ajoutant autant qu'elles peuvent d'autres personnes approvisionnant d'informations le *smart contract*.

L'intérêt d'exécuter *on-chain* cette convention demeure malgré ces éléments car la mise en œuvre d'une convention de niveaux de services dépend, en pratique, grandement de la seule volonté du prestataire, qui n'a pas intérêt à la respecter⁴⁰³. L'utilisation du *smart contract* permet donc de garantir

⁴⁰⁰ En recourant à des oracles qui seront chargés d'informer le *smart contract* de l'atteinte ou non du KPI.

Uriarte, Rafael Brundo, Huan Zhou, Kyriakos Kritikos, Zeshun Shi, Zhiming Zhao, et Rocco De Nicola. « Distributed service-level agreement management with smart contracts and blockchain ». *Concurrency and Computation: Practice and Experience* 33 (28 avril 2020). <https://doi.org/10.1002/cpe.5800>.

An oracle implementation could rely on retrieving data directly from trusted services, such as Provable4, Town Crier, and TLS-N. These, however, are centralized services that suffer from the well-known single point of failure (SPoF) problem and deviate from the discipline of decentralization, which is widely adopted in the blockchain. ChainLink works on distributed oracles that carry the data onto the chain or can trigger transactions only when an agreement is achieved among all the oracles. However, neither ChainLink nor any of the previously mentioned solutions support any kind of QoS monitoring.

⁴⁰¹ On trouve de nombreux exemples de contrats de niveaux de services exécutés par des smart contracts. Par exemple :

John Scheid, Eder, Bruno Rodrigues, Lisandro Granville, et Burkhard Stiller. « Enabling Dynamic SLA Compensation Using Blockchain-based Smart contracts », 2019.

⁴⁰² Neidhardt, Nils, Carsten Köhler, et Markus Nüttgens. « Cloud Service Billing and Service Level Agreement Monitoring based on Blockchain », 2018. <https://www.semanticscholar.org/paper/Cloud-Service-Billing-and-Service-Level-Agreement-Neidhardt-K%C3%B6hler/ea136e5539bd8d3c967f78736c29b3a36c1673a7>.

The service billing smart contract bills a customer based on the service coins that he used(...) This could either be done after receiving the bill, or by locking up Ether in advance, which would automatically be used by the smart contract. In this scenario the smart contract would act as a decentralised escrow party.

⁴⁰³ Uriarte, Rafael Brundo, Huan Zhou, Kyriakos Kritikos, Zeshun Shi, Zhiming Zhao, et Rocco De Nicola. « Distributed service-level agreement management with smart contracts and blockchain ». *Concurrency and Computation: Practice and Experience* 33 (28 avril 2020). <https://doi.org/10.1002/cpe.5800>.

The current service-level agreement (SLA) management solutions cannot easily guarantee a trustworthy, distributed SLA adaptation due to the centralized authority of the cloud provider who could also misbehave to pursue individual goals.

au client le respect des engagements du prestataire sur les niveaux de services⁴⁰⁴. Cela explique pourquoi ce contrat est souvent cité dans la littérature académique relative aux smart contracts comme un beau candidat à une smart contractualisation.

d) Contrat de louage de choses

125. Définition du contrat de louage de choses. Le louage des choses est un contrat par lequel une personne s'oblige à faire jouir à une autre une chose pendant un certain temps et moyennant un certain prix⁴⁰⁵. La nature de la chose louée varie : elle peut être mobilière ou immobilière, matérielle ou immatérielle⁴⁰⁶. Les modalités de versement du prix peuvent également être variées : il peut être versé selon un échéancier, à mesure de la jouissance de la chose⁴⁰⁷, ou encore sous forme de redevances sur les revenus perçus par cette chose.⁴⁰⁸

126. Convenance du contrat de louage pour une exécution par *smart contract*⁴⁰⁹. Un contrat de louage de chose consiste opérationnellement en deux types de flux : la mise à disposition et le retrait du bien au locataire, et le transfert de la rémunération de la location du bien au bailleur. Le bien loué, lorsqu'il est immatériel, et la monnaie pour payer la location sont représentables par des jetons. Ensuite, les conditions déclenchant les transferts sont objectives et quérables dans la *blockchain* si l'opération de location est réalisée en son sein. Ils correspondent en général:

⁴⁰⁴ Hamdi, Nawel, Chiraz El Hog, R. Djemaa, et Layth Sliman. « A Survey on SLA Management Using Blockchain Based Smart contracts », 1425-33, 2022. https://doi.org/10.1007/978-3-030-96308-8_132.

In fact, trust between the consumer and the service provider (SP) is a real issue, especially when there is a violation of the SLA. Usually, the consumer is unable to prove or determine a violation. Therefore, traditional ways to ensure trustworthiness are no longer effective. Recent studies have shown that Blockchain technology and smart contracts (SCs) are an effective solution to address this issue.

⁴⁰⁵ Article 1709 du code civil : *Le louage des choses est un contrat par lequel l'une des parties s'oblige à faire jouir l'autre d'une chose pendant un certain temps, et moyennant un certain prix que celle-ci s'oblige de lui payer.*

⁴⁰⁶ Article 1713 du code civil : *On peut louer toutes sortes de biens meubles ou immeubles.*

⁴⁰⁷ TESTU François-Xavier. Dalloz référence Contrats d'affaires Chapitre 95 – Garanties dans le contrat de louage – SECTION 1 - Notion de louage de chose 95.01. Du louage. 2010^e-2011^e éd. Dalloz, s. d.

[Le contrat de louage] Il s'agit du contrat à exécution successive par lequel une personne (le loueur, le bailleur, le concédant...), s'engage à fournir à une autre (le locataire, le preneur, le licencié...), la jouissance temporaire d'une chose (...), moyennant le paiement d'un prix (le loyer, les redevances...) dû au fur et à mesure du temps d'exécution du contrat – du moins est-ce la pratique normale, car il n'y a pas de nécessité juridique à cet régularité dans le paiement des loyers.

⁴⁰⁸ Comme c'est le cas dans le contrat de location-gérance (article L144-1 du code de commerce et suivants) dans le lequel le locataire-gérant verse une partie des revenus issus de son exploitation du fonds de commerce au bailleur.

⁴⁰⁹ Svtowa, Lynet, et Samukeliso Mabarani. Rental Lease Agreement on Blockchain, 2021.

- à la date de début de location du bien et la date de fin, qui sont des éléments temporels. Ce peut être aussi la cessation du paiement par le locataire. Ces événements vont déclencher les transferts du bien loué ;
- la jouissance du bien est la condition déclenchant ou faisant se poursuivre le transfert des sommes à payer. Si le bien est un NFT, alors cette information peut être aisément renseignée dans le *smart contract*.

127. Contrat de louage exécuté par un *smart contract*. Compte tenu de ces considérations, un *smart contract* pourrait, une nouvelle fois, être utilisé dans le rôle de séquestre afin de sécuriser et mettre en œuvre la location d’une chose. A l’instar des locations de NFT ayant lieu dans le milieu de la *blockchain*⁴¹⁰, les NFT représentant les biens pourraient être déposés dans un *smart contract* et « marqués » comme loués à un locataire. Cette information serait alors transmise à un système *off-chain* qui autoriserait alors la jouissance effective du bien⁴¹¹. Le *smart contract* réceptionnerait aussi le paiement du locataire, que le bailleur pourrait retirer à tout moment. Dès lors que le prix ne serait pas versé selon les modalités convenues et inscrites dans le *smart contract*, le NFT serait remis au bailleur, ce qui indiquerait au système auquel le *smart contract* est connecté que la jouissance a pris fin.

D’autres configurations sont possibles : le bailleur peut choisir de se faire payer en prenant une commission sur les revenus générés par le bien loué. Imaginons que les NFT représentent des biens virtuels qu’un locataire peut exploiter⁴¹². Les revenus tirés de la location peuvent être simplement réceptionnés par un *smart contract* et retirés que dans la proportion convenue et codée dans le

⁴¹⁰ Prudence, Jérémy. NFT pour les Débutants: Achetez, Vendez et Créez Vos Propres NFT Étape Par Étape. Gagnez de l’argent avec l’art numérique, devenez un expert en NFT, en objets de collection cryptographiques. Jérémy Prudence, 2022.

Ce n'est pas grave si vous ne voulez pas vendre vos jetons non fongibles. Pourquoi ? Parce qu'il n'y aura des gens qui auront besoin d'accéder à certains de vos NFTs mais qui ne souhaitent pas les acheter. Tout ce que ce groupe d'individus doit faire est de louer votre NFT, et c'est là que vous pouvez faire de l'argent passif supplémentaire avec vos NFTs sans les vendre.

⁴¹¹ Comme un actif dans un jeu vidéo, une voiture connectée, etc.

⁴¹² Tan, Cindy. « NFT Landlords Are Making Big Bucks from Renting Out Blockchain Game NFTs ». NFTgators (blog), 25 janvier 2022. <https://www.nftgators.com/nft-landlords-are-making-big-bucks-from-renting-out-blockchain-game-nfts/>.

NFT lending is done through a game scholarship programme, originally introduced by the Axie Infinity player community. The scholarships allow players to rent NFTs of in-game tools, skins or creatures, giving gamers the chance to participate in play-to-earn games without having to cough up capital upfront. Lenders then take a cut of the crypto profit from gamers.

programme. Par exemple : 10% pour le bailleur et 90% pour le locataire. L'opportunité demeure aussi pour des biens physiques. Les parties peuvent alors recourir aux objets connectés et les interfacer à leur smart contracts. Ce dernier ne sera utilisé que pour le paiement de la location. En cas de non-paiement, il pourrait, par exemple, déclencher la fermeture d'une serrure du bien loué⁴¹³.

128. Conclusion de la section I. Parmi les contrats étant les plus opportuns à être exécutés par *smart contract* figurent donc ceux qui sont essentiellement constitués de transferts d'actifs et ayant une structure en séquestre ou modélisables en séquestre. En effet, ces conventions forment d'excellentes candidates car elles recourent à la *blockchain* pour ce qu'elle fait de mieux : le transfert de valeur ; et également car leurs structures épousent celles des smart contracts qui ont une fonctionnement rappelant celui du séquestre opérationnel.

Section II - Les contrats partiellement constitués de transferts d'actifs

129. Contrats n'ayant pas une structure en séquestre. La majorité des contrats à titre onéreux ne consistent pas essentiellement en des transferts d'actifs et n'ont pas une structure en séquestre. L'opportunité de leur exécution par *smart contract* n'est alors pas évidente et les parties devront mener une étude approfondie sur la pertinence d'exécuter les processus de ces contrats dans la *blockchain*. Dans le cas où elles en prendraient toutefois la décision, seules certaines clauses de ces conventions se prêteront à cette démarche. Sans prétendre, une nouvelle fois, à l'exhaustivité, nous faisons le constat que les clauses relatives aux paiements (§2) et celles trouvées dans les contrats dits *corporate* (§1) forment d'excellentes candidates à une exécution *on-chain*.

§ I - Les clauses des contrats *corporate*

130. Définition des contrats *corporate*. Nous appelons contrats *corporate*, les conventions qui ont pour thème l'organisation des sociétés et/ou des rapports entre associés d'une société. Les statuts de sociétés (A) et les pactes d'actionnaires (B) figurent parmi ces contrats, qui contiennent plusieurs clauses convenant à une exécution dans la *blockchain*.

⁴¹³ Elise GUILHAUDIS. « Comprendre la blockchain à travers l'étude d'un cas pratique : Le covoiturage "BlockCar" », Lamy Droit de l'immatériel, n° 143 (1 décembre 2017).

A - Les clauses statutaires

131. Définition des statuts de société. Les statuts d'une société⁴¹⁴ sont un contrat visant à organiser le fonctionnement d'une société ainsi que les rapports avec les tiers et entre associés⁴¹⁵. Dans ce document constitutif, se trouve autant de clauses non opérationnelles⁴¹⁶ que de clauses opérationnelles⁴¹⁷ prescrivant des actions concrètes à l'entreprise ou aux associés⁴¹⁸. Or depuis l'ordonnance 2017-1674 du 8 décembre 2017, il est devenu possible de représenter des actions de société par actions par des jetons et ainsi de formaliser des processus statutaires relatifs aux transferts de ces actifs⁴¹⁹.

a) La clause de modalités de vote

132. Définition de la clause de modalités de vote. Dans une société par actions, les décisions sur la société sont prises par vote des associés lors d'assemblées générales où chacun d'eux a droit de faire entendre sa voix lors de délibérations. Le nombre de votes de chaque actionnaire est généralement proportionnel à la quantité d'actions qu'il détient dans le capital social⁴²⁰. Ces décisions peuvent porter sur l'approbation des comptes annuels de la société, la répartition des bénéfices ou

⁴¹⁴ Article 1835 du code civil : *Les statuts doivent être établis par écrit. Ils déterminent, outre les apports de chaque associé, la forme, l'objet, l'appellation, le siège social, le capital social, la durée de la société et les modalités de son fonctionnement. Les statuts peuvent préciser une raison d'être, constituée des principes dont la société se dote et pour le respect desquels elle entend affecter des moyens dans la réalisation de son activité.*

⁴¹⁵ « Statuts de société et actes annexes - Fiches d'orientation - août 2022 | Dalloz ». Consulté le 13 avril 2023.

Les statuts consistent en l'acte constitutif d'une société ou d'une association rédigé par écrit comportant un certain nombre de mentions obligatoires qui posent les objectifs ainsi que les règles de fonctionnement du groupement.

⁴¹⁶ Comme celles relatives à la dénomination de la société, la forme, l'objet, le siège, etc.

⁴¹⁷ V., *infra*, §38

⁴¹⁸ Comme celles relatives aux modalités de versement de dividendes, de cession des actions, etc.

⁴¹⁹ Dominique LEGEAIS. « Blockchain - Blockchain et droit des sociétés - Quelles perspectives ? Quelle incidence véritable de la technologie ? », *Droit des sociétés*, n° 2 (1 février 2022).

On peut envisager que la blockchain facilite la création de la société. Il faut alors envisager la transformation du registre des sociétés en Deep ce qui est techniquement concevable. La blockchain peut aussi transformer la vie de la société en donnant un nouvel essor au phénomène de digitalisation. On peut d'abord utiliser la blockchain pour l'envoi des documents sociaux et les convocations des actionnaires.

⁴²⁰ Article L225-122 du code de commerce : *I.-Sous réserve des dispositions des articles L. 225-10, L. 225-123, L. 225-124, L. 225-125, L. 22-10-46, L. 22-10-47 et L. 22-10-48, le droit de vote attaché aux actions de capital ou de jouissance est proportionnel à la quotité de capital qu'elles représentent et chaque action donne droit à une voix au moins. Toute clause contraire est réputée non écrite(...).*

encore la nomination de nouveaux dirigeants⁴²¹. Dans la société par actions simplifiée (SAS), les associés bénéficient d'une certaine liberté contractuelle⁴²² : ainsi l'article L227-9 du code de commerce leur permet de prévoir librement les modalités de leurs prises de décisions ; elles pourraient donc s'appuyer sur la *blockchain*⁴²³.

133. Convenance de clause de modalités de vote pour une exécution par *smart contract*.

Bien que le vote ne soit pas un transfert d'actifs, il reste un processus très opportun à formaliser dans une *blockchain*. En effet, il est souvent l'évènement déclencheur de transferts d'actifs stipulés dans les statuts de société puisque ce sont régulièrement les décisions des associés qui conditionnent le versement de sommes d'argent ou de mouvement d'actions⁴²⁴.

134. **Vote exécuté par un *smart contract*.** Or un vote peut très simplement être exprimé à l'aide de *smart contract*. On dénombre de nombreuses organisations *on-chain* qui permettent à leurs membres de voter par ce biais : ces derniers sont identifiés par leurs adresses et détiennent des jetons qui représentent leur participation dans le « capital » de l'organisation ; ils peuvent exprimer leur vote en interagissant avec un *smart contract* qui vérifie s'ils ont les droits et enregistre leur choix, à proportion de leur poids de participation⁴²⁵. Une SAS peut donc s'appuyer sur ces mécanismes existants pour permettre à ses associés de voter à l'aide de *smart contract*.

Le vote ainsi réalisé permettrait d'éviter les problèmes de l'hybridation⁴²⁶, en exécutant un processus statutaire purement *on-chain*, de sorte à retenir toutes les propriétés bénéfiques de la *blockchain* : les

⁴²¹ Article L227-9 du code de commerce alinéa 1 : *Les statuts déterminent les décisions qui doivent être prises collectivement par les associés dans les formes et conditions qu'ils prévoient.*

⁴²² Schlumberger, Edmond. « Réflexions sur la liberté contractuelle dans la SAS ». Mélanges offerts à Michel Germain, 2015, 767.

⁴²³ Elizabeth Guégan. Blockchain et assemblées d'actionnaires. Dalloz. Blockchain et droit des sociétés, 2019.

⁴²⁴ Article L227-9 du code de commerce alinéa 2 : *Toutefois, les attributions dévolues aux assemblées générales extraordinaires et ordinaires des sociétés anonymes, en matière d'augmentation, d'amortissement ou de réduction de capital, de fusion, de scission, de dissolution, de transformation en une société d'une autre forme, de nomination de commissaires aux comptes, de comptes annuels et de bénéfices sont, dans les conditions prévues par les statuts, exercées collectivement par les associés.*

⁴²⁵ Florence Guillaume et Sven Riva. « Libres propos - DAO, code et loi : le régime technologique et juridique de la decentralized autonomous organization », Revue de droit international d'Assas, n° 4 (13 décembre 2022).

Une DAO peut être décrite très simplement comme étant une entité constituée et opérant sur une blockchain qui est gérée collectivement par ses membres détenant des droits de gouvernance (tokens de gouvernance). Le processus décisionnel, notamment quant à l'utilisation des fonds de la DAO, implique la participation des membres qui votent en ligne sur les propositions de décision qui sont soumises à la communauté de la DAO (proposals).

⁴²⁶ V., *infra*, §68

associés pourraient alors exprimer leur vote sur l'agrément préalable à une cession de titres, le versement des dividendes, l'exclusion de l'un d'entre eux, la nomination d'un nouveau dirigeant, en restant uniquement dans la *blockchain* et de manière légale.

b) La clause de versement des dividendes

135. Définition de la clause de versement des dividendes. Nous appelons clause de versement des dividendes celle qui détermine les modalités de distribution des bénéfices de la société aux actionnaires. Elle peut prévoir différentes conditions pour le versement des dividendes, tel que le montant minimum de bénéfices nécessaire avant qu'un dividende ne soit versé, la fréquence à laquelle les dividendes seront versés, et la manière dont le montant des dividendes sera déterminé. La décision de verser ou non des dividendes revient aux actionnaires, qui l'exprime par vote généralement après que ceux-ci aient approuvé les comptes annuels⁴²⁷.

136. Conenance de la clause de versement de dividendes pour une exécution par *smart contract*. L'opération mise en œuvre par cette clause consiste en un simple transfert de sommes d'argent à un associé, dans une proportion correspondant à sa part de détention dans le capital de la société. Il s'agit d'actifs représentables en jetons. Ce transfert a lieu, en principe, après le vote décidant la distribution des bénéfices. Nous avons vu précédemment que cet événement peut-être *on-chain*. L'information sur la participation dans le capital de l'associé provient aussi de la *blockchain* si l'actionnariat est enregistré dans un registre en son sein. Nous avons donc un transfert de sommes d'argent dépendant de conditions quérables dans la *blockchain* ; soit un processus entièrement formalisable par *smart contract*⁴²⁸.

137. Versement des bénéfices exécuté par un *smart contract*. Dès lors que les associés auraient exprimé leur approbation des comptes par vote, le *smart contract* de la société détenant en séquestre le montant de ses bénéfices pourrait automatiquement être distribué aux associés à proportion de leurs droits.

⁴²⁷ Article L232-12 du code de commerce : *Après approbation des comptes annuels et constatation de l'existence de sommes distribuables, l'assemblée générale détermine la part attribuée aux associés sous forme de dividendes.*

⁴²⁸ Mustapha Mekki. « If code is law, then code is justice ? Droits et algorithmes ». Gazette du Palais, n° GPL297k2 (27 juin 2017): 10.

Enfin, la blockchain permet la mise en place de smart contracts (...) Les applications possibles sont nombreuses : exécution d'un pacte d'actionnaires, royalties, distribution de dividendes...

c) La clause relative aux apports

138. Définition de la clause relative aux apports. Le statut d'un actionnaire d'une société par actions est conditionné par son apport en nature, en numéraire ou en industrie à la société⁴²⁹. L'associé reçoit alors un nombre d'actions correspondant à la part de ses apports dans le capital social de la société⁴³⁰.

139. Convenance de la clause relative aux apports pour une exécution par *smart contract*. Le processus d'apport peut être exécuté *on-chain* si l'objet apporté est lui-même représentable *on-chain*. Or qu'il s'agisse d'un apport en numéraire ou en nature, ces actifs transférés à la société peuvent être matérialisés par des jetons dans la *blockchain*⁴³¹ : un NFT pour un apport en nature ou des *stablecoin* pour un apport en numéraire. L'élément déclencheur du transfert est le dépôt de l'actif par l'aspirant-associé.

140. Exécution par *smart contract* du processus d'apport. Lorsqu'un *smart contract* représentant la société est constitué, celui-ci peut très bien réceptionner les apports des actionnaires en son sein qu'il tiendra en séquestre, et leur rétribuer automatiquement ensuite des jetons représentant les actions, en fonction de la part de leur apport dans le capital social de la société.

d) La clause d'agrément

141. Définition de la clause d'agrément. Une clause d'agrément est une clause subordonnant la cession de titres d'un associé vers un tiers, ou un autre associé, à l'agrément préalable des actionnaires⁴³². Le but de ce mécanisme est de maîtriser l'arrivée de nouveaux arrivants dans la

⁴²⁹ Article 1832 du code civil : *La société est instituée par deux ou plusieurs personnes qui conviennent par un contrat d'affecter à une entreprise commune des biens ou leur industrie en vue de partager le bénéfice ou de profiter de l'économie qui pourra en résulter.*

⁴³⁰ Article 1843-2 du code civil : *Les droits de chaque associé dans le capital social sont proportionnels à ses apports lors de la constitution de la société ou au cours de l'existence de celle-ci.*

⁴³¹ Yanis-Said Khadiri. « Comment faire un apport en nature de crypto-actif au capital d'une société ? », Village-Justice, 17 février 2022. <https://www.village-justice.com/articles/comment-faire-apport-nature-crypto-actif-capital-une-societe,41704.html>.

Les cryptomonnaies deviennent un instrument de financement des projets d'entreprises. A ce titre, elles peuvent être apportées par les investisseurs ou actionnaires au capital social d'une société.

⁴³² L227-14 code de commerce : *Les statuts peuvent soumettre toute cession d'actions à l'agrément préalable de la société.*

société et la répartition de l'actionnariat⁴³³.

142. Convenance de la clause d'agrément pour une exécution par *smart contract*. Le mécanisme d'une clause d'agrément consiste en un simple flux dépendant d'une condition objective : un transfert de titres d'un associé vers un tiers, si les associés de la société ont agréé ce dernier, c'est-à-dire donner leur aval à la cession. Dès lors que l'expression de l'agrément se fait par vote, ce qui est généralement le cas⁴³⁴, alors la condition de transfert peut-être *on-chain*. Il en résulte un processus très enclin à une exécution par *smart contract*.

143. Exécution de la clause d'agrément par *smart contract*. Ici un *smart contract* pourrait classiquement être utilisé dans un rôle de séquestre. Dès lors qu'un associé souhaiterait vendre ses titres, il les déposerait préalablement dans un *smart contract* séquestre ; de même que l'acheteur déposerait le prix dans ce même programme. Le *smart contract* délivrerait ensuite au tiers cessionnaire les titres si les associés de la société ont exprimé leur agrément à cette cession par l'intermédiaire du même *smart contract* et selon les modalités convenues de la clause. A cet instant, le cessionnaire pourra retirer les titres tokenisés dans le *smart contract* et l'associé le prix de cession.

e) La clause d'exclusion

144. Définition de la clause d'exclusion. Il est courant de prévoir dans des statuts de société les conditions d'exclusion d'un associé ; autrement dit, les événements déclencheurs de la cession forcée des actions d'un associé⁴³⁵. Il peut, par exemple, être prévu que l'ouverture d'une procédure collective à l'encontre d'un associé soit une cause de perte de sa qualité⁴³⁶.

145. Convenance de la clause d'exclusion pour une exécution par *smart contract*.

⁴³³ « Clause d'agrément - Fiches d'orientation - juillet 2022 | Dalloz ». Consulté le 14 avril 2023.

Cette clause permet aux actionnaires de se préserver contre l'intrusion d'un tiers dans la société, mais également (...), de maintenir un équilibre entre les actionnaires existants.

⁴³⁴ Auparavant l'article L227-19 du code de commerce subordonnait la cession des actions à l'accord unanime des associés dans les SAS. Désormais, depuis la loi SAPIN II, il n'y a seul que l'article L227-14 du code de commerce qui prévoit que les statuts peuvent soumettre toute cession d'actions à l'agrément préalable de la société. Autrement dit, cet agrément peut être exprimé par un vote majoritaire.

⁴³⁵ Article L227-16 du code de commerce : *Dans les conditions qu'ils déterminent, les statuts peuvent prévoir qu'un associé peut être tenu de céder ses actions.*

⁴³⁶ Cour de Cassation, Chambre commerciale, du 8 mars 2005, 02-17.692, Publié au bulletin

L'opérationnalité de cette clause est évidente : elle met en œuvre un transfert d'actions si la cause d'exclusion est réalisée. La nature de cette cause est objective puisqu'il s'agit de l'ouverture de la procédure collective⁴³⁷. Même si elle n'est pas *on-chain*, elle est aisément quérable de manière relativement décentralisée sur le web⁴³⁸.

146. Exécution de la clause d'exclusion par un *smart contract*. Si un *smart contract* de la société est déjà constitué, celui-ci peut tenir le registre des actions-jetons de chacun des associés. Dès lors qu'un associé est indiqué par un oracle être en procédure collective, alors le *smart contract* de la société pourra automatiquement transférer les actions-jetons de la société vers elle-même⁴³⁹.

B - Les clauses des pactes d'actionnaires

147. Définition du pacte d'actionnaire. Le pacte d'actionnaire est un contrat conclu entre des associés d'une société, en marge des statuts de société, dont le but est d'organiser certaines relations entre eux à l'abri du regard du public⁴⁴⁰. Beaucoup de clauses des pactes d'actionnaires ont trait au transfert de titres et ont donc une nature très opérationnelle. La différence avec ces clauses qui peuvent aussi être retrouvées dans les statuts de société est que leur irrespect n'est en principe sanctionné que par des dommages et intérêts⁴⁴¹ ; ce qui amoindrit leur efficacité. Les clauses de pactes

(...) Mais attendu qu'après avoir énoncé qu'il est possible et licite de prévoir dans les statuts, qui constituent le contrat accepté par les parties et fixant leurs droits et obligations, que le redressement judiciaire de l'un des associés lui fera perdre cette qualité, dès lors que lui est due la valeur des droits dont il est ainsi privé pour un motif qui est en l'occurrence conforme à l'intérêt de la société et à l'ordre public, l'arrêt relève qu'en vertu de cette clause, la perte des droits d'associés s'opère de plein droit par l'effet du redressement judiciaire de l'associé...

⁴³⁷ V., *infra*, §68

⁴³⁸ Comme déjà évoqué, il est aisément possible de récupérer ces informations en sollicitant les API de sites internet comme InfoGreffé ou societe.com

⁴³⁹ Pour un exemple de clause d'exclusion transformée en *smart contract*, voir la sélection des clauses dans la librairie de *smart contract* à destination des professionnels du droit et de la justice.

Amélie FAVREAU. « Smart contracts - La première librairie européenne et ouverte de smart contracts à destination des professionnels du droit et de la justice ». Mission de recherche droit et justice, s. d.

⁴⁴⁰ « Pacte d'actionnaires - Fiches d'orientation - août 2022 | Dalloz ».

Les statuts sont parfois complétés par un pacte extrastatutaire qui lie tous les actionnaires ou associés, ou certains d'entre eux seulement. Il est destiné à régir certaines questions relatives à leurs relations : exercice du pouvoir au sein de la société (le pacte peut ainsi prévoir une concertation avant toute prise de décision, dans le but d'essayer d'adopter une position commune), droits et obligations des signataires lorsque l'un d'entre eux envisage de céder ses titres.

⁴⁴¹ Ibid.

L'inexécution du pacte ne donne normalement lieu qu'à l'allocation de dommages-intérêts, l'exécution forcée n'étant

d'actionnaires trouvent donc un grand intérêt à être exécutées par des smart contracts⁴⁴² ; c'est notamment le cas de la clause de préemption (a), de *buy or sell* (b) et d'inaliénabilité (c).

a) Clause de préemption

148. Définition de la clause de préemption. Une clause de préemption est, au même titre qu'une clause d'agrément, une clause imposant une condition à l'actionnaire souhaitant céder ses actions à un tiers : celle de les proposer en priorité à ses coassociés⁴⁴³. Comme l'agrément, elle constitue un moyen de contrôle des associés sur l'actionnariat. Concrètement, la clause impose à l'associé souhaitant céder de faire une notification préalable à ses coassociés, qui disposent alors d'un délai pour se prononcer en faveur ou non de l'achat des titres⁴⁴⁴. La clause peut prévoir que seul un quorum minimum d'associés souhaitant exercer leur droit de préemption peut faire échec au projet de vente initial. Ou alors que, nonobstant le nombre de voix final exprimé en faveur de l'achat, une quote-part des titres sera cédée à ceux ayant souhaité acquérir les actions.

149. Convenance de la clause de préemption pour une exécution par *smart contract*. Le fonctionnement d'une clause de préemption de titres est très opérationnel : il consiste en un transfert d'actions vers un individu, le cessionnaire, ou un groupe d'individus, les coassociés, selon que ce dernier a exprimé ou non sa volonté d'en être les acquéreurs. Dès lors que les actions sont représentées par des jetons et que le vote exprimant la décision de préemption est valablement effectué dans la *blockchain*, le processus de la clause de préemption peut être entièrement formalisé par *smart contract* comme la clause d'agrément.

admise que dans l'hypothèse exceptionnelle de la fraude.

⁴⁴² Roda, Jean-Christophe. « Smart contracts, dumb contracts ? » Dalloz IP/IT, n° 07-08 (4 juillet 2018): 397.

Des auteurs français ont également avancé l'idée que la technologie smart contract puisse être utilisée dans la vie des affaires, par exemple dans le cadre de la mise en œuvre de pactes d'actionnaires comportant des droits de préemption.

⁴⁴³ Il s'agit juridiquement d'un pacte de préférence. Article 1123 du code civil : *Le pacte de préférence est le contrat par lequel une partie s'engage à proposer prioritairement à son bénéficiaire de traiter avec lui pour le cas où elle déciderait de contracter.*

⁴⁴⁴ Maria-Beatriz Salgado. « Fasc. Q-30 : SOCIÉTÉS ANONYMES. – Pactes d'actionnaires et clauses de préemption non statutaires ». In JurisClasseur Sociétés Formulaire, s. d.

Les clauses de préemption insérées dans un pacte d'actionnaires ont pour objet de contraindre un actionnaire, lorsqu'il souhaite céder ses titres, à les offrir en priorité aux autres associés signataires du pacte. C'est seulement si le bénéficiaire n'exerce son droit de préemption dans le délai fixé, que le cédant peut proposer ses titres à un tiers.

150. Clause de préemption exécutée par smart contract. Nous faisons état de certains protocoles mettant en place des mécanismes de préemption automatisés dans la *blockchain*⁴⁴⁵. Dans notre contexte, un mécanisme de préemption peut-être implémenté à l'aide d'un *smart contract* contrôlé par la société ; celui-ci détiendrait les actions et les céderait aux cessionnaires ou aux coassociés selon le résultat du vote, comme dans la clause d'agrément. En cas de préemption « réussie », les associés pourraient recevoir les titres à proportion de leur quote-part dans le capital social de la société.

b) Clause de *buy or sell*

151. Définition de la clause de buy or sell. La clause de *buy or sell* est une clause courante des pactes d'actionnaires permettant à un associé de contraindre un autre de lui acheter ses titres à un prix déterminé ou, dans le cas où ce dernier refuserait, de le forcer à vendre ses propres titres au même prix. Cette clause vise à débloquent des situations de conflits dans la gouvernance d'une société en forçant un des deux protagonistes à, *in fine*, quitter la société⁴⁴⁶.

152. Convenance de la clause de buy or sell pour une exécution par smart contract. L'opération mise en œuvre par cette clause consiste en des transferts d'actifs dépendant de conditions objectives : les actions des associés sont vendues selon que l'un accepte ou refuse d'acheter celles de celui actionnant le mécanisme. Il s'agit de processus qui peuvent être mises en œuvre à l'aide de *smart contract*.

153. Exécution d'une clause de buy or sell par un smart contract. Le processus exécuté à l'aide d'un *smart contract* séquestre pourrait prendre cette forme : dès lors que la clause serait valablement actionnée, les titres de la personne l'actionnant (actionnaire A) ainsi que celle contre qui

⁴⁴⁵ Dans ce site internet, il est possible de créer le smart contract d'un pacte d'actionnaire qui met en œuvre le mécanisme d'un droit de préemption : <https://s3.us-east-2.amazonaws.com/liquidity-prototype/index.html>

Dans la librairie européenne de *smart contract*, la clause de préemption fait partie des premières clauses proposées.

Amélie FAVREAU. « Smart contracts - La première librairie européenne et ouverte de smart contracts à destination des professionnels du droit et de la justice ». Mission de recherche droit et justice, s. d.

⁴⁴⁶ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires, Clause de buy or sell. Les Intégrales. LGDJ, 2018.

La clause buy or sell, littéralement achetez ou vendez, permet de résoudre des situations de blocage, notamment en présence d'un conflit entre associés(...). Dans une première approche, cette clause peut être définie comme la clause permettant à un actionnaire de proposer la vente de ses titres à un prix déterminé à un autre actionnaire ou à défaut d'acquiescer les siens au prix auquel il était prêt à les lui céder.

elle est actionnée (actionnaire B) seraient transportés dans le *smart contract* séquestre, avec le prix d'achat des actions versé par l'actionnaire A.

Si après l'écoulement du délai prévu dans la clause, l'actionnaire B n'a toujours pas versé le prix des actions de l'actionnaire A dans le programme, alors ses actions seraient transférées à l'actionnaire A ; tandis que l'actionnaire B disposerait de la faculté de retirer le prix des actions versé par l'actionnaire A. Inversement, si l'actionnaire B paye dans les modalités convenues, alors les actions de l'actionnaire A lui reviendraient tandis que l'actionnaire A pourrait récupérer le prix de la vente⁴⁴⁷.

c) Clause d'inaliénabilité

154. Définition de la clause d'inaliénabilité. La clause d'inaliénabilité est une clause visant à interdire à des associés de vendre leurs actions pendant un certain délai⁴⁴⁸. Elle peut figurer dans les statuts comme dans un pacte d'actionnaires. Elle a souvent pour but de contraindre certains types d'associés, comme l'équipe dirigeante typiquement, à demeurer dans l'entreprise afin de rassurer des investisseurs de l'engagement sur le long terme des associés⁴⁴⁹.

155. Convenance de la clause d'inaliénabilité pour une exécution par *smart contract*. Le fonctionnement de la clause d'inaliénabilité semble peu opérationnel puisqu'il impose une inaction. Mais techniquement cette inaliénabilité se traduit simplement par un transfert d'actifs (les actions représentées en jetons) bloqué. Dès lors que les actifs sont *on-chain*, il est tout à fait possible

⁴⁴⁷ Dans la librairie de *smart contract* du projet de recherche droit et justice, cette clause figure parmi celles sélectionnées.

Amélie FAVREAU. « Smart contracts - La première librairie européenne et ouverte de smart contracts à destination des professionnels du droit et de la justice ». Mission de recherche droit et justice, s. d.

⁴⁴⁸ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires. Les Intégrales. LGDJ, 2018.

[En droit des sociétés], Il est fréquent également de prévoir l'inaliénabilité temporaire des droits sociaux. La clause peut figurer soit dans les statuts soit dans un pacte extra-statutaire.

⁴⁴⁹ Ibid.

L'idée est de conserver durablement un associé, de stabiliser l'actionnariat, de plafonner la participation des associés pour figer durablement les pouvoirs des uns et des autres ou, indirectement, de disposer d'un veto sur la cession des participation.

de prévoir un mécanisme empêchant leur transfert pendant un certain délai⁴⁵⁰.

156. Clause d'inaliénabilité exécutée par un *smart contract*. Pour exécuter une telle clause à l'aide d'un *smart contract* séquestre, il suffirait de prévoir simplement que le programme ne pourra pas faire de nouvelle assignation pendant un certain délai. Ce mécanisme est déjà largement mis en œuvre, dans le milieu de la *blockchain*, à travers la technique du *vesting*⁴⁵¹; il pourrait donc aisément être répliqué pour exécuter une clause d'inaliénabilité d'une SAS.

§ II - Les clauses relatives au paiement de sommes d'argent

157. Définition de clauses relatives au paiement de sommes d'argent. Nous appelons clauses relatives au paiement les clauses qui prévoient les modalités de versement d'une somme d'argent à un individu dans un contrat. Nous avons vu que le transfert de ces actifs se prêtent bien à une formalisation en *smart contract*⁴⁵². Il est donc possible de proposer plusieurs clauses implémentant ces processus comme de bonnes candidats à une exécution dans une *blockchain* ; elles peuvent être classées entre celles ayant une architecture proche du séquestre (A) et celles qui sont modélisables ainsi (B).

A – Les clauses de paiement sous forme de séquestre

158. Clauses de paiement aux mécanismes de séquestre. Parmi les clauses de paiement ayant un fonctionnement étant celui du séquestre, ou pouvant s'approcher du séquestre, nous pouvons identifier au moins deux clauses issues du monde de la fusion-acquisition : la clause de *earn-out* (a) et la clause de garantie de passif (b).

⁴⁵⁰ Arnaud LECOURT. « Droit des sociétés et numérique – Chapitre 2 Numérique et fonctionnement de la société - Section 2 Les opérations sur les titres », Répertoire IP/IT et Communication, novembre 2020.

Par exemple, la présence dans les statuts d'une clause d'inaliénabilité des titres pour 5 ans interdit mécaniquement, en présence de ce protocole informatique, de transférer ou d'effectuer une quelconque opération sur les titres concernés avant l'expiration de cette durée.

⁴⁵¹ Ledger. « Vesting », 21 février 2023. <https://www.ledger.com/academy/glossary/vesting>.

Vesting in crypto involves setting aside some of a coin's total supply and releasing them into the market after certain conditions have been met. The period in which the tokens are locked up is known as the vesting period or token lock up, and investors cannot transact or trade those specific tokens during this time. This helps reduce market manipulation and token dumping to improve a project's stability.

⁴⁵² V., *infra*, §51

a) Clause de *earn-out*

159. Définition de la clause *earn-out*. Une clause de *earn-out* est une clause qu'on trouve typiquement dans les contrats de fusions-acquisitions. Elle vise à indexer une partie du prix de cession d'une société à ses résultats futurs⁴⁵³. L'acheteur s'entend avec le vendeur pour lui verser le prix de cession en deux fois :

- une partie fixe immédiatement après la conclusion du contrat,
- et une autre qui correspond à un pourcentage du chiffre d'affaires ou de l'ebitda fiscal de la société vendue, un certain nombre d'années après la cession⁴⁵⁴.

De cette sorte, l'acheteur et le vendeur seront assurés d'acheter et de vendre la société proche de sa véritable valeur ; car une partie de son prix aura été indexée sur ses résultats financiers⁴⁵⁵. Il arrive que la clause de *earn-out* soit mise en œuvre à l'aide d'un tiers-fiduciaire. Celui-ci est alors chargé de sécuriser l'engagement de l'acheteur en séquestrant le prix maximal de la cession et en versant la juste somme au vendeur selon les modalités convenues⁴⁵⁶.

⁴⁵³ Lamy Sociétés Commerciales - Partie 1 Règles communes à tous les types de sociétés - Titre 3 Les acteurs de la vie sociétaire : dirigeants — associés — pactes d'associés - Division 2 Les associés : droits, obligations et parts sociales (évaluation, détention, cession, transmission, saisie) - Chapitre 5 Cession des parts ou actions de l'associé : conditions générales du contrat - Section 2 Les exigences permanentes - § 3. Le contenu - B. — Le prix

Les parties peuvent convenir qu'un complément de prix viendra s'ajouter au prix versé lors de la cession [d'une société]. Ce versement sera généralement prévu sous réserve de la réalisation de conditions que les parties détermineront ensemble. ...). En pratique, les compléments de prix seront souvent d'un montant variable qui sera déterminé en fonction des résultats futurs de la société : par exemple, les résultats sur les deux prochaines années.

⁴⁵⁴ Cécile Sommelet. « Les modalités optimales de l'*earn-out* ». Option Finance, La lettre des fusions-acquisitions et du private equity, 1 octobre 2019.

*La détermination du complément de prix éventuellement versé par l'acquéreur au vendeur est le point clé du mécanisme de la clause d'*earn-out*. (...). Les agrégats financiers retenus (marge brute, EBITDA, résultat opérationnel, etc.) devront être définis précisément et accompagnés d'un exemple de calcul chiffré.*

⁴⁵⁵ Lamy Sociétés Commerciales - Partie 1 Règles communes à tous les types de sociétés - Titre 3 Les acteurs de la vie sociétaire : dirigeants — associés — pactes d'associés - Division 2 Les associés : droits, obligations et parts sociales (évaluation, détention, cession, transmission, saisie) - Chapitre 5 Cession des parts ou actions de l'associé : conditions générales du contrat - Section 2 Les exigences permanentes - § 3. Le contenu - B. — Le prix

*(...) Cette clause dite d'*earn out* ou d'intéressement peut présenter l'avantage « de sécuriser un investissement, notamment lorsque celui-ci est réalisé dans une société dont le résultat risque de baisser »..*

⁴⁵⁶ Arnold Rouah. « La fiducie au service du M&A ». Village de la Justice (blog), 21 novembre 2015. <https://www.village-justice.com/articles/fiducie-service,40749.html>.

*(...) Le transfert d'actions ou de parts sociales peut utilement recourir à la fiducie pour sécuriser l'exécution de cessions impliquant un complément de prix (*earn-out*) ou une promesse de vente à terme (*call option*). Quelle partie et son conseil ne se sont pas posés la question angoissante du paiement de son *earn-out* ou de la délivrance des actions promises et n'ont pas craint la lourdeur judiciaire ou arbitrale lorsque survient un conflit ?*

160. Convenance de la clause de *earn-out* pour une exécution par *smart contract*.

Plusieurs experts dans le domaine des fusions-acquisitions ont identifié la clause de *earn-out* comme une des clauses de leur secteur convenant le mieux à une exécution dans la *blockchain*⁴⁵⁷. En effet, elle consiste en un transfert de sommes d'argent pouvant dépendre de conditions très objectives : comme le montant du chiffre d'affaire de l'entreprise ou son *ebidta*. Cette donnée est facilement quérable sur des bases de données publics et relativement distribuée⁴⁵⁸. Le processus consiste donc en un transfert d'actifs dépendant d'une condition objective.

161. Clause de *earn-out* exécutée par *smart contract*. Un *smart contract* séquestre peut donc être constitué pour servir de réceptacle de la somme maximale à verser, et la délivrer qu'après l'écoulement du délai convenu et dans la proportion en rapport avec le chiffre d'affaire ou l'*ebidta* de l'entreprise cédée. Un oracle sera nécessaire pour fournir cette donnée au *smart contract* qui pourra être récupérée sur le web ou auprès d'un tiers déterminé comme un expert-comptable⁴⁵⁹.

b) Clause de garantie de passif

162. Définition de la clause de garantie de passif. Toujours dans le domaine des fusions-acquisitions, il est également fréquent de trouver des clauses de garantie de passifs dans les cessions d'entreprise. Celles-ci visent à prémunir l'acquéreur d'une société contre la hausse de son passif, découverte après la cession, suite à un événement survenu antérieurement à la conclusion de l'acte de cession⁴⁶⁰. Il s'agit donc d'une clause protectrice de l'acquéreur car elle requiert du vendeur de

⁴⁵⁷ Bissegger, Mark. « Smart contract Applications in M&A: Earn-Outs ». Deal Law Wire, 22 novembre 2017. <https://www.deallawwire.com/2017/11/22/smart-contract-applications-in-ma-earn-outs/>.

Inherent in the traditional earn-out contract is a significant counter-party risk. This risk is manifested by the need for each party to trust that the other will behave according to the rules of the contract. One way to address this trust deficit is to build the conditions of the earn-out into a smart contract so that the seller is automatically paid in accordance with the rules of the contract if certain conditions are met.

⁴⁵⁸ Le chiffre d'affaire et/ou de l'EBITDA d'une entreprise sont des données aisément quérables sur le site Pappers.fr par exemple.

⁴⁵⁹ Alexis MARCHAND. « Décrypter les enjeux d'une cession ou d'une reprise pour le cédant ou le repreneur afin de donner des conseils pratiques », Lamy Droit des affaires, n° 116 (1 juin 2016).

Les enjeux d'une cession d'entreprise pour le cédant et le repreneur sont naturellement différents, parfois même contradictoires, mais doivent nécessairement se conjuguer pour atteindre l'objectif recherché. Compte tenu de la complexité de ces enjeux, il est utile, si ce n'est indispensable, que chaque partie se fasse accompagner par des conseils professionnels, principalement experts-comptables, conseils financiers et avocats, et s'adjoignent l'expertise d'autres conseils selon les spécificités de l'activité de l'entreprise.

⁴⁶⁰ « Clause de garantie de passif - Fiches d'orientation - juillet 2022 | Dalloz ». Consulté le 16 avril 2023.

s'engager à l'indemniser si des dettes sont découvertes un certain temps après la cession.

En pratique, cette clause est parfois mise en œuvre à l'aide de tiers-fiduciaires. Comme pour la clause de *earn-out*, ces derniers sont alors chargés de séquestrer le montant maximal que peut atteindre la garantie et verser la somme appropriée, correspondant au montant du passif, à l'acheteur si des dettes sont effectivement révélées dans les conditions prévues au contrat⁴⁶¹.

163. Convenance de clause de garantie de passif pour une exécution par *smart contract*. Il s'agit d'un mécanisme qui institue un transfert de sommes d'argent dépendant d'une condition relativement objective. En effet, dans une garantie de passif, c'est une dette révélée de l'entreprise cédée qui déclenche un transfert des sommes d'argent. Bien que l'évaluation et la mesure de cet événement nécessitent un jugement expert, la nature de l'événement en soi est objective. Cet élément, en plus du fait que la clause soit structurée en séquestre maintiennent sa convenance pour une formalisation *on-chain*.

164. Clause de garantie de passif exécutée par des *smart contracts*. Un tiers désigné par des parties pourrait donc être assisté par un *smart contract* séquestre et se contenter d'informer celui-ci de la révélation d'une dette d'une part et du montant devant être délivré au bénéficiaire d'autre part. La réception, gestion et délivrance seraient effectués par le programme ; ce qui rationaliserait le processus d'exécution de cette clause comme pour la fiducie⁴⁶².

B - Les clauses de paiement modélisables en séquestre

165. Clauses de paiement n'ayant pas une structure en séquestre. D'autres clauses, mettant en œuvre un versement de sommes d'argent peuvent être opportunément exécutées par des *smart contracts*, malgré qu'elles n'aient pas une architecture modélisable en séquestre ; c'est notamment le cas de la clause pénale (a), et de celle relative au versement de rémunération (b).

En vertu d'une clause de garantie de passif, le cédant d'un nombre important de parts sociales ou d'actions s'engage à prendre à sa charge tout ou partie des dettes sociales existantes antérieurement à la cession et qui se révéleraient postérieurement à celle-ci.

⁴⁶¹ Thierry Granier. « Fiducie sûreté et fiducie gestion, les premiers pas... », RTDF, n° 4-2010, p.98-102.

Même s'il faut tenir compte des pré-requis de la loi, il apparaît que la fiducie peut servir dans de nombreuses situations. Ainsi, elle pourrait accompagner le mécanisme de garantie de passif.

⁴⁶² V., *infra*, §83

a) La clause pénale

166. Définition de la clause pénale. La clause pénale est visée à l'article 1231-5 du code civil⁴⁶³. Elle a pour but d'imposer à celui qui a manqué une obligation contractuelle de payer une somme d'argent à titre de dommages et intérêts. Si les parties sont libres de fixer la somme forfaitaire à payer, l'alinéa 2 de l'article 1231-5 du code civil dispose que le juge peut la moduler si elle est *manifestement excessive ou dérisoire*. Elle peut se retrouver dans des contrats provenant de divers secteurs : contrat de bail⁴⁶⁴, contrat informatique⁴⁶⁵, contrat de vente⁴⁶⁶...

167. Convenance de la clause pénale pour une exécution par *smart contract*. Cette clause est souvent citée en exemple de celles les plus indiquées pour une formalisation *on-chain*⁴⁶⁷. Pourtant, son mécanisme ne fonctionne pas comme celui d'un séquestre. Il s'agit, en effet, d'un engagement du débiteur d'une obligation de verser une somme d'argent s'il n'exécute pas celle-ci. Néanmoins, si l'obligation prend lieu *on-chain* ou si l'information de sa réalisation ou non peut facilement être renseignée à un *smart contract*, le processus reste indiqué pour une formalisation en *smart contract*.

En effet, il consiste en un transfert de sommes d'argent en cas d'inexécution de la part du débiteur. Cet événement peut être objectif et produit *on-chain* ou par un programme. Exemple : l'obligation pour un fournisseur d'accès internet de produire un débit internet minimum⁴⁶⁸ ou à défaut de diminuer

⁴⁶³ Article 1231-5 du code civil : *Lorsque le contrat stipule que celui qui manquera de l'exécuter paiera une certaine somme à titre de dommages et intérêts, il ne peut être alloué à l'autre partie une somme plus forte ni moindre.*

⁴⁶⁴ Rose-Noëlle SCHÜTZ. « § 2 - Indemnité de résiliation 272. La nature de l'indemnité de résiliation : une clause pénale. », Répertoire de droit civil Crédit-bail, octobre 2015.

⁴⁶⁵ Olivier DORCHIES. « La clause pénale dans les contrats informatiques et télécoms », Communication commerce électronique, 2014, p. 14.

⁴⁶⁶ Emilie, Cambournac. « Clause pénale et vente immobilière, un gage de loyauté dans la conduite des pourparlers ». Village de la Justice (blog), 20 février 2015. <https://www.village-justice.com/articles/clause-penale-matiere-vente-immobiliere-gage-loyaute-dans-conduite-des,33814.html>.

⁴⁶⁷ Lipton, Alex, et Stuart Levi. « An Introduction to Smart contracts and Their Potential and Inherent Limitations ». The Harvard Law School Forum on Corporate Governance (blog), 26 mai 2018. <https://corpgov.law.harvard.edu/2018/05/26/an-introduction-to-smart-contracts-and-their-potential-and-inherent-limitations/>.

Smart contracts are presently best suited to execute automatically two types of "transactions" found in many contracts: ensuring the payment of funds upon certain triggering events and imposing financial penalties if certain objective conditions are not satisfied.

⁴⁶⁸ V., *infra*, §122

la facture du client d'un montant correspondant à cette perte de débit. Ou encore, l'obligation pour un transporteur de livrer une marchandise connectée à une certaine date⁴⁶⁹.

168. Clauses pénales exécutées par des smart contracts. Il est observé un certain nombre d'expérimentations illustrant le potentiel des smart contracts à travers l'exécution de clauses pénales⁴⁷⁰. Un de ceux-là est celui d'une clause pénale exécutée avec un thermomètre connecté : un transporteur conduit une marchandise dans un camion frigorifié, avec l'obligation de maintenir une certaine température. Si la température du camion descend en dessous d'un seuil, il est reconnu une violation de l'obligation du transporteur et le versement d'une indemnité forfaitaire⁴⁷¹.

Dans cette configuration, la somme de la clause pénale serait préalablement séquestrée dans un *smart contract* et récupérée ensuite par le créancier si l'oracle-thermomètre indique l'irrespect de l'obligation. La clause trouverait ainsi une toute nouvelle vigueur grâce à son exécution par *smart contract*. De plus, son régime légal pourrait très bien s'accommoder d'une exécution par *smart contract* : le juge peut, a posteriori, moduler le montant de la clause pénale si elle est jugée excessive ou dérisoire⁴⁷².

⁴⁶⁹ Abdoulaye DIALLO. « Smart contract d'une clause pénale. » Village de la Justice (blog), 19 mars 2021. <https://www.village-justice.com/articles/smart-contract-une-clause-penale.30899.html>.

⁴⁷⁰ « Example: Late Delivery Clause · Accord Project ». Consulté le 16 avril 2023. <https://docs.accordproject.org/index.html>.

In the rest of this specification, we will use the Late Delivery And Penalty legal clause as an example. It is a common clause in a legal contract related to the delivery of good or services, and in some circumstances may be amenable to automation.

⁴⁷¹ Pour des exemples, voir :

Baygin, Mehmet, Orhan Yaman, Nursena Baygin, et Mehmet Karakose. « A Blockchain-Based Approach to Smart Cargo Transportation Using UHF RFID ». *Expert Systems with Applications* 188 (1 février 2022): 116030. <https://doi.org/10.1016/j.eswa.2021.116030>.

(...) there is a GPS device that allows one to track the location of the transport vehicle and a thermometer that measures the temperature of the package's environment in the vehicle. The information obtained from these sensors is constantly transferred to a cloud server. (...) the cloud server is used to confirm whether or not the terms of the smart contract are met.

Kumar, S., A. Murugan, B. Muruganantham, et B. Sriman. « IoT-smart contracts in data trusted exchange supplied chain based on block chain ». *International Journal of Electrical and Computer Engineering (IJECE)* 10 (1 février 2020): 438. <https://doi.org/10.11591/ijece.v10i1.pp438-446>.

Blockchain enable us to have a distributed, digital ledger. IoT (Internet of Things) sensor devices (zigbee) utilizing blockchain technology to assert public availability of temperature records, tracking location shipment, humidity, preventing damage, data immutability. The sensor devices looking the temperature, location, damage of each parcel during the shipment to completely guarantee directions.

⁴⁷² Article 1231-5 du code civil alinéa 2 : Néanmoins, le juge peut, même d'office, modérer ou augmenter la pénalité ainsi convenue si elle est manifestement excessive ou dérisoire.

b) Clauses relatives au versement de rémunération

169. Définition des clauses relatives au versement de rémunération . Ici, nous visons toutes les clauses qui ont trait à la rémunération d'un travail accompli. Cela peut concerner les rémunérations des prestataires dans des contrats d'entreprise⁴⁷³ ou des salariés dans les contrats de travail⁴⁷⁴.

170. Convenance d'une clause relative au versement de rémunération pour une exécution par *smart contract*. Ce sont des clauses qui sont par nature très opérationnelles car elles consistent en des transferts de sommes d'argent. Ces dernières peuvent être représentées par des jetons et l'évènement déclenchant leur mouvement peut être très objectif et *on-chain*. Pour un contrat de travail, la condition déclenchante est principalement temporelle : le salarié est payé chaque mois pour les jours où il était présent. Il s'agit donc d'une condition relativement objective et *off-chain*. Pour un contrat de prestation de service, la condition de déclenchement des transferts sera rarement réalisée *on-chain*, mais peut être objective et facilement quérable en ligne selon la nature de la prestation.

171. Clause de rémunération exécutée par des *smart contracts*. Il est dénombré de nombreux *smart contracts* exécutant des clauses de rémunération dans le milieu de la *blockchain*. Certaines applications utilisent même des propriétés innovantes de la *blockchain* pour améliorer la manière dont ces clauses ont toujours été exécutées. Ainsi, les protocoles *Sablier* ou *Superfluid*⁴⁷⁵ permettent de rémunérer des individus en "temps réel", c'est-à-dire littéralement à mesure que le temps passe⁴⁷⁶. Par exemple, il est possible de configurer un *smart contract* afin qu'il verse une somme de dix centimes d'euro, en *stablecoin*, par minute pendant deux heures. Et la personne recevra

⁴⁷³ Qui est un contrat dit aussi de louage d'ouvrage (article 1710 du code civil) dans lequel l'une des parties s'engage à faire quelque chose pour l'autre, moyennant un prix convenu entre elles.

⁴⁷⁴ Cour de cassation du 22 juillet 1954, Bull. civ. IV, no 576 : *Le contrat de travail est une convention par laquelle une personne s'engage à travailler pour le compte d'une autre et sous sa subordination moyennant une rémunération.*

⁴⁷⁵ Ce sont tous deux des protocoles DeFi qu'on trouve dans la Blockchain Ethereum (*superfluid.finance* et *sablier.finance*)

⁴⁷⁶ Brady DALE. « Bitcoin Will Change Money Like the Internet Changed Video ». Observer (blog), 17 janvier 2017. <https://observer.com/2017/01/bitcoin-lightning-network-andreas-antonopoulos/>.

If I can do payments in a millionth of a second frequency and are as low as a satoshi, why not get your salary every minute?"he asks. "When you can make micropayments over milliseconds, cash flow takes on a whole new meaning.

effectivement dix centimes d’euro dans son *wallet*⁴⁷⁷, chaque minute qui passe, pendant deux heures. L’architecture de ces smart contracts fonctionne toujours en séquestre : un *smart contract* détient les jetons et les rend disponibles une fois la condition remplie⁴⁷⁸. Pour des contrats de prestation de services ou de travail, un client ou un employeur pourrait déposer les sommes dans un *smart contract* et configurer leur délivrance selon ce qui a été convenu. Très souvent, les parties devront recourir à un oracle : un système de badgeage connecté au *smart contract* pourra être mis en place pour déclencher le versement de la paie aux salariés ; ou encore le recours à une solution d’arbitrage *on-chain* dans les contrats de prestation de services en cas de dispute au moment de récupérer les sommes séquestrées⁴⁷⁹.

172. Conclusion du chapitre. Il existe bien d’autres contrats et clauses comportant tout ou partie des processus évoqués au précédent chapitre qui forment des bons candidats à une exécution par *smart contract*. Nous estimons pourtant que ce sont en priorité ceux composés essentiellement de transferts d’actifs et ayant une architecture en séquestre qui conviennent le mieux à une exécution dans la *blockchain*. Toutefois, certains contrats et clauses qui n’ont pas cette forme mais qui mettent en œuvre des transferts d’actifs, peuvent être modulées de sorte à fonctionner également en séquestre sans que soit altérée leur nature. Ceux-là constituent aussi des bons candidats à une formalisation *on-chain*.

⁴⁷⁷ V., *supra*, §408

⁴⁷⁸ Sablier. « Sablier ». Consulté le 18 avril 2023. <https://www.sablier.finance>.

After a one-time deposit, the Sablier smart contracts will start "streaming" the tokens towards the recipient, without you lifting a finger again.

⁴⁷⁹ Comme le permet par exemple, la solution d’arbitrage *on-chain* Kleros (<https://kleros.io/escrow/>).

173. L'étape de la délimitation du domaine formel. Une fois décidé ce qu'il convient d'exécuter dans la *blockchain*, les parties peuvent passer à l'étape de la détermination de la forme de leur contrat intelligent. Quelle doit être la nature de l'*instrumentum*⁴⁸⁰ d'un contrat intelligent ? Doit-il être représenté que par le code informatique des programmes l'exécutant, par un texte écrit en langage naturel, ou un mélange des deux ? Cette étape est cruciale, car c'est à partir de l'*instrumentum* du contrat que pourra être décelée la volonté des parties. Parmi les différents *instrumentum* possibles des contrats intelligents (Chapitre I), nous estimons que le plus adéquat et sécurisant pour les parties est celui formé du texte écrit en langage naturel et de code informatique, dans lequel le texte tient lieu de principale manifestation de leur volonté (Chapitre II).

Chapitre I – Les différents *instrumentum* des contrats intelligents

174. Panorama des différents *instrumentum* de contrats intelligents. Quelles sont les différentes natures d'*instrumentum* des contrats intelligents actuellement observables et que valent-elles ? Nous pouvons distinguer celle où il n'existe que le code des smart contracts pour matérialiser l'accord (Section I), de celles qui sont additionnellement composées d'un texte écrit en langage naturel (Section II).

Section I – Le code informatique comme seul acte instrumentaire

175. L'absence de formalisation écrite en langage naturel. La plupart des contrats intelligents identifiables dans le milieu de la *blockchain* sont dépourvus de textes écrits en langage naturel pour exprimer la volonté des parties (§1). Cette forme de contrat présente, selon nous, plus

⁴⁸⁰ Debard, Thierry, et Serge Guinchard. Lexique des termes juridiques 2020-2021 - 28e ed. Médiation conventionnelle. Edition 2020-2021. Dalloz, 2020.

(...) écrit authentique ou sous signature privée contenant la substance de l'acte juridique qu'il constate. En cas de non concordance entre le *negotiatum* et l'*instrumentum*, c'est le *negotium* qui l'emporte car il correspond aux volontés réelles.

d'inconvénients que d'avantages (§2).

§1 – Le choix le plus répandu

176. Une forme répandue car conforme à l'esprit du milieu. Cette forme de contrat intelligent trouve de nombreuses manifestations dans le milieu de la *blockchain* (B), sans doute car elle constitue la manière la plus fidèle à l'ethos *cypherpunk*⁴⁸¹ de créer des contrats (A).

A – Un choix dans l'esprit du milieu de la *blockchain*

177. Le code est le droit. *Code is law* est une expression⁴⁸² populaire dans le milieu de la *blockchain* qui exprime l'idée que le monde de la cryptographie serait un espace autonome dans lequel ses « habitants » ne compteraient pas sur les institutions du monde *fiat*⁴⁸³ pour garantir leurs droits⁴⁸⁴. Dans l'univers de la *blockchain*, les individus ont toute souveraineté sur leurs actifs et ne s'appuient que sur les outils fournis par la cryptographie (*code*) pour faire régner la loi. Il est aisé de déceler l'idéologie anarcho-libertarienne⁴⁸⁵ derrière cette vision ; et bien que l'usage de la cryptomonnaie s'étende au-delà de ce premier cercle politique, cette pensée libertaire dont elle est historiquement issue reste encore très prégnante et constituera longtemps un des traits saillants de

⁴⁸¹ V., *infra*, §26

⁴⁸² Elle provient de l'article du juriste américain Lawrence Lessig.

Lessig, Lawrence. « Code Is Law ». Harvard Magazine, 1 janvier 2000.
<https://www.harvardmagazine.com/2000/01/code-is-law.html>.

⁴⁸³ Le monde réel/ordinaire par opposition au monde crypto.

⁴⁸⁴ « Dossier : Enjeux et défis de la blockchain en propriété intellectuelle ». Dalloz IP/IT, 2018, p. 530.

La blockchain suscite aussi des peurs, dont la plus grande pour le juriste est celle d'être évincée, selon la formule magique « Code is law ». Cette technologie décentralisée dans le monde entier, en dehors de tout cadre général et de toute institution publique, suscite donc l'intérêt du juriste.

⁴⁸⁵ Rose, Julie L. Review of Review of Libertarian Anarchy: Against the State, par Gerard Casey, 19 juin 2013.
<https://ndpr.nd.edu/reviews/libertarian-anarchy-against-the-state/>.

The standard libertarian anarchist position, classically espoused by Rothbard, is to argue that a state is not required to protect individual property rights because that function could instead be performed by competing, non-monopolistic protection agencies.

cette technologie⁴⁸⁶.

En pratique, *code is law* signifie que recourir à un *smart contract* vaut une sorte d'acceptation tacite par ses utilisateurs de se soumettre à ses propriétés pour le pire comme pour le meilleur. En cas de défaillance ou de comportement inattendu de celui-ci, ils acceptent les conséquences de l'immutabilité du code, qu'elles leur soient favorables ou non. Et même quand il est techniquement possible de revenir sur cette immutabilité, le faire revient à rompre cette règle tacite⁴⁸⁷. Ainsi lors du piratage de the DAO en 2016, le *fork* de la *blockchain* Ethereum pour remédier à ses conséquences néfastes a été perçu par une grande partie de la communauté comme tout à fait contraire à l'ethos de la *blockchain*⁴⁸⁸.

178. Code is law dans le contexte des contrats intelligents. Pour des parties d'un contrat intelligent adeptes de cette philosophie, le recours à un *smart contract* afin d'exécuter un accord signifie qu'elles font le choix de ne dépendre que de la *blockchain* pour le faire respecter. Dans le meilleur des scénarios, le *smart contract* exécute leur convention tel qu'elles l'entendaient. Dans le pire, il dysfonctionne et les parties se soumettent tout de même à son résultat⁴⁸⁹.

Dans ce paradigme, il n'y a donc pas besoin d'un document écrit en langage naturel pour retranscrire les obligations des parties car celui-ci sert à prouver et rendre conforme le contrat à l'ordre juridique

⁴⁸⁶ Hughes, Eric. « A Cypherpunk's Manifesto », 9 septembre 1993. <https://www.activism.net/cypherpunk/manifesto.html>.

⁴⁸⁷ Dans un article, un informaticien célèbre de l'écosystème Ethereum appelle cette philosophie Szabo's Law. En référence à Nick Szabo qui serait, selon lui, la personne qui a théorisé et popularisé cette pensée.

Zamfir, Vlad. « Against Szabo's Law, For A New Crypto Legal System ». Crypto Law Review (blog), 29 janvier 2019. <https://medium.com/cryptolawreview/against-szabos-law-for-a-new-crypto-legal-system-d00d0f3d3827>.

(...) I'm naming this crypto law after Nick Szabo, since I am pretty convinced that he created it, popularized it, and brought it into crypto law. I'm sorry if I'm missing anyone else who deserves credit for it, but I'm just going to assume that Nick is responsible so I can keep my sentences short. Szabo's law is simple: Do not implement changes to the blockchain protocol unless the changes are required for the purpose of technical maintenance.

⁴⁸⁸ Falkon, Samuel. « The Story of the DAO — Its History and Consequences ». The Startup (blog), 12 août 2018. <https://medium.com/swlh/the-story-of-the-dao-its-history-and-consequences-71e6a8a551ee>.

Unsurprisingly, the hack was the beginning of the end for the DAO. The hack itself was contested by many Ethereum users, who argued that the hard fork violated the basic tenets of blockchain technology.

⁴⁸⁹ Quinn Emanuel Urquhart. « Client Alert: Code is Law ». Consulté le 25 avril 2023. <https://www.quinnemanuel.com/the-firm/publications/code-is-law/>.

Under a most expansive view of "code is law," if the code of a smart contract permits something, then it is "legal." This theory holds that code shall prevail, whether or not it conflicts with anything else. Those who accept his literal meaning of "code is law" contend that, even in the event of a bug or glitch in the code, that same code still governs. Since algorithmic law is unambiguous, they argue that it reduces the subjectivity inherent in the traditional legal and judicial systems.

du monde *fiat*, afin que ses institutions puissent garantir son exécution. Or, dans la philosophie *code is law*, cet ordre est méprisé au profit de la cryptographie. Le code fait office de support unique et parfait représentant la volonté des parties.

Ce sont pour ces raisons notamment que les contrats intelligents dépourvus d'écrit en langage naturel sont nombreux. Leurs parties sont insensibles à la sécurité juridique conférée par un document textuel ; ils préfèrent utiliser les smart contracts afin de se défaire précisément de la nécessité de compter sur l'office du juge ou d'autres tiers pour faire respecter leur contrat. Pourtant en vertu du principe du consensualisme, ces accords peuvent former des contrats légaux au regard du droit français. En effet ce dernier n'impose, en principe, qu'un accord de volontés⁴⁹⁰ entre au moins deux individus pour que soit formé un contrat : dès lors qu'une offre comportant ses éléments essentiels⁴⁹¹, a été acceptée de manière ferme et univoque⁴⁹² un contrat peut être créé. Pour que celui-ci soit valide, les parties doivent avoir été en capacité de contracter, avoir consenti de manière libre et éclairée et le contenu doit être licite et certain⁴⁹³. Ainsi, malgré leur forme sans texte écrit en langage naturel, nombre de ces accords exécutés dans la *blockchain* reçoivent tout de même une qualification juridique précise dans le code civil français.

B – Exemples de contrats intelligents sans texte écrit en langage naturel

179. Les contrats de vente. Les accords de vente font partie de ces contrats exécutés sans texte écrits en langage naturel dans la *blockchain* et pourtant susceptibles d'avoir une réception juridique précise en droit français. Ainsi, il est courant que dans certaines *marketplaces* web3 des détenteurs de NFT proposent à la vente leurs créations à des clients potentiels, avec qui le site se

⁴⁹⁰ Article 1101 du code civil : *Le contrat est un accord de volontés entre deux ou plusieurs personnes destiné à créer, modifier, transmettre ou éteindre des obligations.*

⁴⁹¹ Article 1114 du code civil : *L'offre, faite à personne déterminée ou indéterminée, comprend les éléments essentiels du contrat envisagé et exprime la volonté de son auteur d'être lié en cas d'acceptation. A défaut, il y a seulement invitation à entrer en négociation.*

⁴⁹² Article 1118 du code civil : *L'acceptation est la manifestation de volonté de son auteur d'être lié dans les termes de l'offre. Tant que l'acceptation n'est pas parvenue à l'offrant, elle peut être librement rétractée, pourvu que la rétractation parvienne à l'offrant avant l'acceptation. L'acceptation non conforme à l'offre est dépourvue d'effet, sauf à constituer une offre nouvelle.*

⁴⁹³ Article 1128 du code civil : *Sont nécessaires à la validité d'un contrat : 1° Le consentement des parties ; 2° Leur capacité de contracter ; 3° Un contenu licite et certain.*

charge de les mettre en relation⁴⁹⁴. Le vendeur renseigne dans les métadonnées du NFT ses caractéristiques principales et fixe sur la plateforme son prix d'achat. L'opération de vente est ensuite entièrement réalisée par un *smart contract* et à aucun moment ne figure un accord écrit en langage naturel cadrant la relation entre le vendeur et l'acheteur. Il figure bien des conditions générales dans le site permettant de référencer ces NFT, mais ces dernières cadrent la relation entre les utilisateurs et la plateforme⁴⁹⁵.

Ainsi, l'acheteur et le vendeur, s'ils sont bien en capacité de contracter, ont conclu un contrat de vente valide, au sens de l'article 1583 du code civil⁴⁹⁶. Celui-ci disposant, en effet, qu'un contrat de vente est parfait dès lors que les parties se sont accordées sur le prix et la chose. Nul besoin d'un écrit, sinon à titre de preuve.⁴⁹⁷

180. Les contrats de louage. Il est possible de qualifier également des contrats de louage dans ce même milieu des NFT. Le développement des jeux vidéo web3 a fait émerger une activité consistant à louer des NFT à des joueurs pour qu'ils puissent jouer à des jeux les rémunérant en cryptomonnaie. La location est exécutée exclusivement à l'aide de *smart contract* : le NFT est placé en séquestre dans un programme qui l'identifie comme loué à une adresse et les crypto-monnaies remportées grâce à cet actif sont ensuite partagées entre le joueur et le loueur selon ce qui a été convenu et programmé entre eux⁴⁹⁸.

⁴⁹⁴ Nous pouvons citer la plateforme OpenSea (<https://opensea.io/>), qui est la plus connue.

⁴⁹⁵ OpenSea. « Terms Of Service ». OpenSea. Consulté le 29 septembre 2023. <https://opensea.io/tos>.

Introduction. (...) OpenSea is not party to any agreement between any users. You bear full responsibility for verifying the identity, legitimacy, and authenticity of NFTs that you purchase from third-party sellers using the Service and we make no claims, guarantees, or recommendations about the identity, legitimacy, functionality, or authenticity of users or NFTs (and any content associated with such NFTs) visible on the Service.

⁴⁹⁶ Article 1589 du code civil : [La Vente] est parfaite entre les parties, et la propriété est acquise de droit à l'acheteur à l'égard du vendeur, dès qu'on est convenu de la chose et du prix, quoique la chose n'ait pas encore été livrée ni le prix payé.

⁴⁹⁷ Article 1359 du code civil : L'acte juridique portant sur une somme ou une valeur excédant un montant fixé par décret doit être prouvé par écrit sous signature privée ou authentique. Il ne peut être prouvé outre ou contre un écrit établissant un acte juridique, même si la somme ou la valeur n'excède pas ce montant, que par un autre écrit sous signature privée ou authentique. Celui dont la créance excède le seuil mentionné au premier alinéa ne peut pas être dispensé de la preuve par écrit en restreignant sa demande. Il en est de même de celui dont la demande, même inférieure à ce montant, porte sur le solde ou sur une partie d'une créance supérieure à ce montant.

⁴⁹⁸ Tan, Cindy. « NFT Landlords Are Making Big Bucks from Renting Out Blockchain Game NFTs ». NFTgators (blog), 25 janvier 2022. <https://www.nftgators.com/nft-landlords-are-making-big-bucks-from-renting-out-blockchain-game-nfts/>.

Cette relation pourrait correspondre à un louage de choses au sens de l'article 1709 du code civil, qui le définit comme *un contrat par lequel l'une des parties s'oblige à faire jouir l'autre d'une chose pendant un certain temps, et moyennant un certain prix que celle-ci s'oblige de lui payer*. Or en principe, le louage de chose est un contrat consensuel. Aussi, dès lors que les protagonistes de cette opération satisfassent les conditions de validité visées à l'article 1128 du code civil, ces locations de NFT par *smart contract* peuvent constituer des contrats de louage au sens du droit civil français, nonobstant la présence d'un écrit.

181. Les contrats de prêt. Il est dénombré également plusieurs protocoles de finance décentralisée permettant de prêter et d'emprunter des jetons. Il pourrait même être argué que cette activité constitue encore le produit phare de ce milieu⁴⁹⁹. Or ces services sont entièrement et uniquement exécutés à l'aide de *smart contract* : une personne dépose en garantie un montant de jetons ou des NFT dans un programme et reçoit en contrepartie la somme en jetons stables qu'il souhaite emprunter.

Dans la plupart de ces protocoles de prêt, il est malaisé de qualifier une relation contractuelle entre prêteurs et emprunteurs, car leur fonctionnement ressemble beaucoup à celui de banques ordinaires : l'emprunteur retire le montant emprunté d'une réserve de liquidités constituée des dépôts d'usagers souhaitant générer des intérêts sur leurs économies. Il n'y a donc pas de relation directe entre les premiers et derniers, comme il n'y a pas de relation directe entre ceux qui empruntent auprès d'une banque et ceux qui déposent leurs épargnes⁵⁰⁰.

Toutefois il existe d'autres protocoles proposant des services prêt en pair-à-pair : soit la possibilité pour un emprunteur de choisir l'offre de prêt à laquelle il souhaite souscrire parmi une liste présentée sur une plateforme. Chacune des offres est une proposition provenant d'un individu en particulier, avec ses propres termes. L'opération de prêt est réalisée de la même manière que dans les protocoles de prêts classiques : uniquement à l'aide de *smart contract*, l'emprunteur dépose un crypto-actif en garantie et reçoit en contrepartie la somme empruntée en *stablecoin*⁵⁰¹. Ces relations-là peuvent

NFT lending is done through a game scholarship programme, originally introduced by the Axie Infinity player community. The scholarships allow players to rent NFTs of in-game tools, skins or creatures, giving gamers the chance to participate in play-to-earn games without having to cough up capital upfront. Lenders then take a cut of the crypto profit from gamers.

⁴⁹⁹ ACPR. « Finance « décentralisée » ou « désintermédiée » : quelle réponse réglementaire ? », 3 avril 2023.

Le prêt-emprunt collatéralisé (lending) : il s'agit de la principale activité, en TVL « nette », au sein de la DeFi.

⁵⁰⁰ Nous pensons particulièrement au protocole de prêt AAVE (<https://aave.com/>) qui est le plus populaire dans le milieu crypto.

⁵⁰¹ Les plateformes NFTfi (<https://nftfi.com/>) et PWN (<https://pwn.xyz/>) permettent ces prêts en pair-à-pair exécutés uniquement à l'aide de *smart contract*.

caractériser un contrat de prêt à consommation au sens de l'article 1892 du code civil ; ce dernier étant formé par la remise de choses fongibles et consommables, que peuvent être des *stablecoin*,⁵⁰² de manière consensuelle.

§ II – Un choix pratique, mais risqué juridiquement

182. Une commodité porteuse de risque. Les bénéfices de la praticité (A) apportées par l'absence de texte écrit en langage naturel ne suffisent pas à surmonter ses inconvénients dus à l'insécurité juridique que cette absence crée (B).

A – La praticité du code comme seul *instrumentum*

183. Gain de productivité des contrats formalisés que par du code. Le premier avantage de cette forme est, très naturellement, l'économie de coûts et de temps que procure le fait de ne pas prendre la peine de rédiger un document textuel. Dans certains contextes, les parties peuvent en effet estimer que l'effort de produire un texte écrit est dispensable. Par exemple, si la valeur de leur contrat est faible et que les smart contracts utilisés pour l'exécuter sont particulièrement éprouvés et audités par une large communauté d'utilisateurs et de développeurs. Dans cette situation, elles peuvent être tentées de se satisfaire de la sécurité technique de leurs programmes sans y rajouter la sécurité juridique conférée par un document rédigé. Au-delà de toute considération idéologique ou philosophique, il faut rappeler qu'une *blockchain* sert à garantir l'exécution des conventions de manière commode : dans cet univers, il ne suffit que de coder et déployer correctement un programme pour s'assurer de l'exécution de son contrat⁵⁰³.

184. Débridement de la forme sans texte écrit en langage naturel. Rédiger un écrit pour

⁵⁰² Au même titre que les bitcoins : voir en ce sens la décision du Tribunal de commerce de Nanterre, 6ème Chambre, 26 février 2020, n° 2018F00466 :

(...) *Que le BTC étant fongible et consommable, la qualification juridique des 3 contrats de prêt de BTC signés entre les parties les 1er septembre 2014, 11 janvier 2016 et 23 juin 2016 est donc bien celle de prêt de consommation.*

⁵⁰³ Alharby, Maher, et Aad van Moorsel. « Blockchain-based Smart contracts: A Systematic Mapping Study ». In *Computer Science & Information Technology (CS & IT)*, 125-40, 2017. <https://doi.org/10.5121/csit.2017.71011>.

A smart contract is executable code that runs on top of the blockchain to facilitate, execute and enforce an agreement between untrusted parties without the involvement of a trusted third party.

des parties procèderait également d'une volonté de se soumettre à notre ordre juridique⁵⁰⁴. Or certaines relations permises par la *blockchain* s'accommodent mal des qualifications et régimes juridiques de notre Droit⁵⁰⁵. Essayer de se conformer au droit positif peut alors conduire à brider les fonctionnalités des contrats intelligents et sous-exploiter leur potentiel. Ce recul d'opportunité se justifie d'autant moins que des parties peuvent vouloir, comme évoqué, ne pas du tout compter sur le Droit pour garantir l'exécution de leurs contrats⁵⁰⁶. En se déchargeant de la peine de rédiger un écrit, et plus généralement de chercher à se conformer au Droit, les parties peuvent plus commodément et librement exploiter le potentiel des smart contracts. Les inconvénients portés par cette forme nous paraissent toutefois bien plus importants que ses bénéfices.

B – L'insécurité juridique liée au code comme seul *instrumentum*

185. Perte de la maîtrise de la qualification et du régime applicable lié au code comme seul *instrumentum*. L'absence document écrit en langage naturel peut entraîner la perte de la maîtrise de la qualification du contrat et donc de son régime juridique. En effet, même lorsque des parties se réclament de la philosophie *code is law*, elles ne peuvent légalement se soustraire à l'application d'un droit étatique⁵⁰⁷. Ainsi, le considérant 13 du règlement Rome I prévoit que des parties à un contrat international ne sont autorisées, seulement, qu'à faire référence dans leur contrat à un droit non étatique⁵⁰⁸. Elles ne peuvent s'arroger le pouvoir d'être régi uniquement par leurs propres normes⁵⁰⁹. Cette position est partagée en droit français par une jurisprudence constante de la Cour de cassation⁵¹⁰.

⁵⁰⁴ Lorsqu'un contrat est formalisé par écrit, cela est fait soit à titre *ad probationem* (afin de prouver son contenu), soit à titre *ad validatem* (afin de le rendre valide). Dans tous les cas, il est compté sur l'ordre juridique pour rendre efficace la convention.

⁵⁰⁵ Par exemple, le paiement de salaire en *streaming* comme il se fait régulièrement dans la DeFi ne peut pas être implémenté en Droit Français en vertu de l'article L3242-1 du code du travail.

⁵⁰⁶ V., *infra*, §177

⁵⁰⁷ Leveneur, Claire. « Les smart contracts : étude de droit des contrats à l'aune de la blockchain ». §22, p. 26. These de doctorat, Université Paris-Panthéon-Assas, 2022. <https://www.theses.fr/2022ASSA0063>.

En définitive, il n'est pas raisonnable de penser pouvoir faire échapper la blockchain et ses applications au droit. La blockchain existe dans un monde gouverné par le droit (...). La blockchain est donc, dans ses applications, soumise au droit. Le code informatique, qui structure la blockchain, ne peut remplacer la loi.

⁵⁰⁸ *Le présent règlement n'interdit pas aux parties d'intégrer par référence dans leur contrat un droit non étatique ou une convention internationale (...).*

⁵⁰⁹ Bureau D. et Muir Watt H., Droit international privé. Partie spéciale, t. II, PUF, coll. Thémis, 4e éd, 2017, p. 411, spéc. p. 415

⁵¹⁰ Cass. com., 21 juin 1950, D. 1951 p. 749, note Hamel

(...) Tout contrat international est nécessairement rattaché à la loi d'un État.

Les parties qui n'auraient donc pas pris soin de prévoir quoi que ce soit par écrit se verraient appliquer par défaut un corps de normes potentiellement incommodes pour elles.

Imaginons des parties créant l'équivalent d'une société par actions simplifiée dans la *blockchain*, en constituant une DAO⁵¹¹. Cette organisation pourrait être dotée de toutes les caractéristiques de cette forme société : mécanisme de capital social avec apport en crypto-monnaie et rétribution en jetons-actions, distribution des bénéfices et vote *on-chain*... mais sans avoir la personnalité juridique d'une SAS. En dépit de la volonté des parties de ne dépendre que de la *lex cryptographia*⁵¹² pour gérer leur organisation, celle-ci recevrait immanquablement une réception en droit français. A défaut d'incorporation légale, elle pourrait être considérée comme une société de fait⁵¹³. A ce titre, elle exposerait l'ensemble de ses membres à un engagement personnel vis-à-vis des contractants de la DAO⁵¹⁴, contrairement à une SAS où la responsabilité des associés est limitée⁵¹⁵ et le dirigeant contracte, en principe, au nom et pour le compte de la société⁵¹⁶.

186. Preuve du contrat. Un des intérêts fondamentaux de la production d'un document

⁵¹¹ Une DAO pour *decentralized autonomous organisation* est une organisation administrée à l'aide de *smart contract* dans la *blockchain*.

⁵¹² C'est-à-dire le code des smart contracts que les parties ont voulu institué pour seule loi.

Becker, Katrin. « Blockchain Matters—Lex Cryptographia and the Displacement of Legal Symbolics and Imaginaries ». *Law and Critique* 33, n° 2 (1 juillet 2022): 113-30. <https://doi.org/10.1007/s10978-021-09317-8>.

Lex cryptographia is a law that is no longer legitimized by a culturally established symbolic referent which it no longer needs to be as there is no longer a need for recognition or belief: by programming the code, the parties to a smart contract are making law, implying—or rather coding—the values they take to be fundamental, and initiating the law's automatic execution: legal basis, law writing, law and its enforcement fall into one.

⁵¹³ Article 1873 du code civil : *Les dispositions du présent chapitre sont applicables aux sociétés créées de fait.*

BOFiP BOI-BIC-CHAMP-70-20-60 - 12/09/2012

L'appellation de société créée de fait recouvre les sociétés qui (...) bien que réunissant les éléments de fond du contrat de société (mise en commun d'apports, participation aux bénéfices et aux pertes, intention de s'associer sur un pied d'égalité quant au contrôle de l'affaire) n'ont donné lieu à aucune des formalités prescrites par la loi et les règlements pour leur constitution régulière.

⁵¹⁴ Article 1872-1 du code civil : *Chaque associé contracte en son nom personnel et est seul engagé à l'égard des tiers. Toutefois, si les participants agissent en qualité d'associés au vu et au su des tiers, chacun d'eux est tenu à l'égard de ceux-ci des obligations nées des actes accomplis en cette qualité par l'un des autres, avec solidarité, si la société est commerciale, sans solidarité dans les autres cas.*

⁵¹⁵ Article L227-1 du code de commerce : *Une société par actions simplifiée peut être instituée par une ou plusieurs personnes qui ne supportent les pertes qu'à concurrence de leur apport.*

⁵¹⁶ Article L227-6 du code de commerce : *La société est représentée à l'égard des tiers par un président désigné dans les conditions prévues par les statuts. Le président est investi des pouvoirs les plus étendus pour agir en toute circonstance au nom de la société dans la limite de l'objet social.*

écrit est de clarifier et prouver le contenu des obligations entre des parties⁵¹⁷. Lorsque ces dernières adoptent la forme *code is law*, elles choisissent de se priver de ce moyen de preuve pour ne s'appuyer que sur le code de leur smart contracts comme indicateur de leur volonté écrite. Or celui-ci constitue un moyen de preuve médiocre sur le plan légal. En effet, l'article 1358 du code civil permet, en principe, la preuve par tous moyens des faits et actes juridiques⁵¹⁸ mais l'article 1359 du code civil impose que la preuve de tout acte juridique dont la valeur dépasse 1500 soit faite par écrit⁵¹⁹.

L'écrit, selon l'article 1366 du code civil, peut être manuscrit ou électronique. Ce dernier dispose de la même force probante que l'écrit manuscrit à la condition qu'*il puisse dûment identifier la personne dont il émane et qu'il soit établi et conservé dans des conditions de nature à en garantir l'intégrité*⁵²⁰. Le code d'un smart contract peut satisfaire cette condition d'intégrité, mais échoue à la condition d'identification⁵²¹. En effet, dans un écrit électronique, l'auteur doit être identifié par une signature électronique, qui elle-même doit satisfaire les conditions de la signature électronique qualifiée du règlement eIDAS⁵²².

Ce dernier impose, entre autres, le recours à un tiers détenteur d'un *dispositif de création de signature*

⁵¹⁷ François Xavier Testu. « Chapitre 31 – Liberté contractuelle, écrit et intitulé – Section 1 Question préliminaire : écrit ou pas écrit ? - §3 Ecrit comme exigence pratique - 31.18. Utilité de l'écrit », Dalloz référence Contrats d'affaires, 2010.

Quelles que soient les règles applicables, l'écrit est nécessaire en fait pour des raisons de sécurité juridique, car son existence donne aux parties l'occasion de clarifier leur volonté commune, et il évite un grand nombre de litiges. Il s'impose dès que l'opération économique en cause atteint le moindre degré de complexité ou suppose la stipulation de conditions qui ne sont pas d'usage.

⁵¹⁸ Article 1358 du code civil : *Hors les cas où la loi en dispose autrement, la preuve peut être apportée par tout moyen.*

⁵¹⁹ Article 1359 du code civil : *L'acte juridique portant sur une somme ou une valeur excédant un montant fixé par décret doit être prouvé par écrit sous signature privée ou authentique.*

⁵²⁰ Article 1366 du code civil : *L'écrit électronique a la même force probante que l'écrit sur support papier, sous réserve que puisse être dûment identifiée la personne dont il émane et qu'il soit établi et conservé dans des conditions de nature à en garantir l'intégrité.*

⁵²¹ Barbet-Massin, Alice. « Le droit de la preuve à l'aune de la blockchain ». Phd thesis, Université de Lille, 2020. <https://theses.hal.science/tel-03124881>.

La problématique de l'identification dans la blockchain est latente, ce qui a tendance à compromettre la qualification d'écrit électronique pour les transactions de la blockchain. L'intégrité des données garantie par l'immuabilité du registre qui trouve sa justification technique par l'empreinte numérique [le hash] des blocs de transactions est néanmoins satisfaite.

⁵²² Décret n° 2017-1416 du 28 septembre 2017 relatif à la signature électronique, 2017-1416 § (2017) – Article 1

La fiabilité d'un procédé de signature électronique est présumée, jusqu'à preuve du contraire, lorsque ce procédé met en œuvre une signature électronique qualifiée.

Règlement (UE) N° 910/2014 du parlement Européen et du conseil du 23 juillet 2014 sur l'identification électronique et les services de confiance pour les transactions électroniques au sein du marché intérieur et abrogeant la directive 1999/93/CE - Article 32 Exigences applicables à la validation des signatures électroniques qualifiées

électronique qualifiée, reposant sur un *certificat qualifié de signature électronique*⁵²³. Ce qui est, en pratique, irréalisable dans une *blockchain* publique⁵²⁴. En d'autres termes, le code d'un *smart contract* ne peut avoir la même valeur probante juridique qu'un document contractuel rédigé classiquement en langage naturel, c'est-à-dire soit manuellement ou soit électroniquement dans les conditions satisfaisant l'article 1366 du code civil.

Au-delà de l'aspect probatoire, le code d'un *smart contract* constitue également un médiocre instrument de preuve du rapport d'obligations, car il n'est pas destiné à cet usage. Il consiste en des instructions informatiques ayant vocation à être compilées par une machine virtuelle⁵²⁵. Il n'a pas pour rôle d'éclairer sur le contexte de conclusion du contrat ou l'étendue des obligations des parties, et donc constituera toujours à ce titre un pauvre support pour indiquer la volonté des parties comparé à un véritable document écrit en langage naturel⁵²⁶.

187. Exposition face aux risques de bugs et piratages. Enfin cette absence de texte peut particulièrement exposer les parties aux risques de bugs et piratages des smart contracts. Dans notre contexte, un tel incident signifie que le programme a mal appliqué la volonté véritable des parties. Son comportement a dévié de l'accord qu'elles avaient conclu, certes consensuellement. Il n'existe alors que le code du programme pour connaître leur volonté réelle et éventuellement revenir sur les agissements erronés du *smart contract*. Pour des tenants de la philosophie "*code is law*", cela est un compromis acceptable mais pour ceux ayant fait recours à cette forme par simple commodité, ce n'est

⁵²³ Ibid – Article 3.12 :

«signature électronique qualifiée», une signature électronique avancée qui est créée à l'aide d'un dispositif de création de signature électronique qualifiée, et qui repose sur un certificat qualifié de signature électronique.

⁵²⁴ Barbet-Massin, Alice. « Le droit de la preuve à l'aune de la blockchain ». Phd thesis, Université de Lille, 2020. <https://theses.hal.science/tel-03124881>.

L'absence de vérification de signature par un PSCQ attestée par la délivrance d'un certificat ne permet pas au procédé de signature de la blockchain de bénéficier de la qualification de signature électronique qualifiée. La nécessaire ré-intermédiation du PSCQ dans la signature électronique qualifiée est tout à fait critiquable dans le contexte de la blockchain. L'alourdissement du processus probatoire est exactement ce que la technologie cherche à abolir en théorie. Partant, l'intervention de ce tiers est un obstacle à l'assimilation de la signature électronique qualifiée pour une signature blockchain.

⁵²⁵ V., *supra*, §417

⁵²⁶ Sklaroff Jeremy. « Smart contracts and the Cost of Inflexibility ». University of Pennsylvania Law Review 166, n° 1 (1 janvier 2017): 263.

While such contracts [written contracts] are not the only type of enforceable agreement, they are the most efficient way to ensure that the court correctly understands what parties were willing to exchange under their deal. That understanding can be essential when the court needs to supplement or correct the agreement. And, as we will see, these documents provide parties with important tools to manage uncertainties inherent in the agreement process and responses if the agreement goes wrong.

pas le cas.

188. Conclusion de la Section I. Le code comme seul acte instrumentaire est une forme de contrat intelligent dans laquelle ce dernier n'est manifesté que par le code des smart contracts qui l'exécutent. Elle est la forme la plus commode à mettre en œuvre car elle ne nécessite pas la production d'un document écrit en langage naturel détaillant le rapport contractuel. Cela explique sa popularité avec le fait qu'elle est en concordance avec la philosophie prégnant l'univers de la *blockchain*. Pourtant cette forme est marquée d'une grande insécurité juridique, puisque le code d'un *smart contract* constitue un médiocre instrument de preuve et renseignements sur la volonté réelle des parties.

Section II – Le texte écrit en langage naturel comme acte instrumentaire

189. Les contrats intelligents comportant un texte écrit en langage naturel. Au côté des contrats intelligents seulement pourvus du code pour matérialiser les intentions des parties, co-existent ceux qui sont en plus dotés de stipulations rédigées dans un langage courant. Il s'agit de contrats exécutés par des smart contracts mais qui disposent, classiquement, d'un document écrit ordinaire pour cadrer la relation contractuelle. Au sein de cette catégorie, il est encore possible de faire la distinction entre les contrats dans lesquels le code est inféodé à ce document, qui est le premier support de la volonté des parties (§1), et ceux où le code a un rôle égal ou concurrent à ce dernier (§2).

§ I – Le texte en tant *qu'instrumentum* prioritaire sur le code

190. Architecture du contrat intelligent donnant priorité au texte en langage naturel . Ces contrats ont l'architecture et la composition classiques des conventions automatisées par des programmes ordinaires⁵²⁷ :

- un document, souvent électronique, consistant en un texte écrit en langage naturel qui

⁵²⁷ Ces contrats sont nommés par Harry Surden des « contrats computationnels ». Ils incluent les contrats intelligents mais généralement toutes sortes de contrats exécutés par des programmes, qui fonctionnent sur blockchain ou non.

Surden, Harry. « Computable Contracts ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network,

gouverne toute la relation contractuelle ;

- une suite de programmes conçus pour exécuter certains des processus stipulés dans la convention écrite et n'ayant pas ou très peu de valeur juridique.

Cette forme de contrat est répandue dans le monde *fiat*, à défaut d'être très utilisée dans celui de la *blockchain* (A). Son principal avantage tient à la sécurité juridique qu'elle confère aux parties malgré sa relative lourdeur (B)

A – Un choix minoritaire dans le monde de la *blockchain*

Cette architecture de contrat automatisé par des programmes est très fréquente (a) en dehors de l'univers de la *blockchain* (b).

a) Une forme répandue dans le monde *fiat*

191. Contrats automatisés « classiquement ». Nous faisons couramment l'expérience de contrats exécutés par des programmes dans notre quotidien. Seulement pour l'extrême majorité d'entre eux, cette exécution n'a pas lieu dans la *blockchain*. Ce sont des programmes résidant sur les serveurs privés d'une partie qui sont en charge de l'exécution de l'accord⁵²⁸. Ces contrats sont formalisés par un accord écrit en langage naturel et les programmes exécutant les stipulations de cet

2012. <https://papers.ssrn.com/abstract=2216866>.

The basic idea behind a computable contract term is to create a series of actionable, computer-processable instructions that approximate what it is that the parties are intending to do in their contractual arrangement. In certain contexts, computer systems can be instructed how to assess contract terms in a way that mirrors the parties' intentions. Further, the parties can sometimes provide the computer with data that is relevant to making determinations of conformance with specified contract terms.

⁵²⁸ GOSSA Julien. « Les blockchainss et smart contracts pour les juristes ». Dalloz IP/IT n°7-8 (2018), p. 393-97.

En réalité, nous faisons l'expérience quotidienne des contrats auto-exécutés, par exemple, lorsque nous accédons à un service en ligne après avoir effectué un paiement, ou lorsque nous sommes automatiquement facturés et prélevés suite à l'utilisation d'un service. Dans ces cas, ce sont des machines qui vérifient les conditions du contrat et en exécutent les clauses. Mais si nous confions ces tâches à ces machines, c'est seulement parce que nous faisons confiance à leurs propriétaires.

accord y sont strictement inféodés, au point même qu'ils font rarement partie du corpus contractuel⁵²⁹.

Prenons l'exemple d'une convention de compte courant entre un client et sa banque⁵³⁰. Ces contrats prévoient souvent une clause de découvert autorisant le client à dépenser plus que ce dont il est créancier à l'égard de la banque. Cette clause, en pratique, est exécutée de façon particulièrement automatique : lorsqu'un client paie un bien ou un service alors que son solde créditeur est à 0, il lui est automatiquement alloué la somme dont il a besoin dans son compte, sans aucune friction et dans la limite du montant prévu dans la convention de compte courant⁵³¹. Cette opération s'analyse pourtant en un prêt de consommation⁵³². Et le client a donc le devoir de rembourser la somme qui lui a été prêtée dans un certain délai et en s'acquittant d'intérêts. Ce remboursement peut avoir lieu lorsque le client « recrédite » son compte qui était débiteur, ou « à découvert » ; il est alors automatiquement prélevé les intérêts applicables et le prêt est clos.

Cette clause de découvert, dans la convention de compte courant, ne fait pourtant même pas mention du programme qui l'exécute car celui-ci n'importe pas juridiquement : il n'est qu'une modalité d'exécution de ce qui a été convenu. Seules comptent les obligations du prêt⁵³³ ; la manière dont il sera exécuté appartient à un autre plan qui n'a pas forcément besoin de figurer dans le contrat. Les logiciels développés par la banque pour mettre en œuvre cette clause ne se contentent donc de

⁵²⁹ Cohney, Shaanan, et David A. Hoffman. « Transactional Scripts in Contract Stacks ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 2020. <https://doi.org/10.2139/ssrn.3523515>.

As Jason Allen has recently argued, transactional scripts [programs] are the latest in a series of "contractware", i.e., technological artefacts designed to embody and perform contracts. "A chip in a credit card embodies the concept well. There is a natural language credit agreement between you and your card company, updated and modified at the issuers' will against the background of regulation, which define circumstances under which credit may be extended...

⁵³⁰ Anne-Marie TOLEDO-WOLFSOHN. « Répertoire de droit civil - Compensation et compte-courant - Article 3 - §4 », avril 2017.

La convention de compte courant est celle « par laquelle deux personnes décident de porter réciproquement en compte toutes les opérations juridiques qui s'effectueront entre elles, de manière à ce qu'il y ait des compensations réciproques, et de ne procéder en principe au règlement qu'à la clôture du compte par le paiement du solde. Elle est généralement le support d'une opération de crédit et cela en deux sens : de par sa structure, le compte courant comporte l'octroi d'un délai de paiement ; en outre, s'appuie souvent sur le compte courant l'octroi d'une facilité de crédit par la banque (...)

⁵³¹ « Modèle de convention de compte courant la Banque Populaire Méditerranée », 22 juin 2021. https://web.archive.org/web/20210622101835/https://www.mediterranee.banquepopulaire.fr/portailinternet/Editorial/Lists/DocEditoList/Convention_de_compte_courant-BPMED-01-12-2020.pdf.

7.1.2. Autorisation de découvert ; a) Octroi et fonctionnement ; La BANQUE peut accorder expressément au CLIENT une autorisation de découvert dont les conditions, notamment de montant et de taux applicables (...) En ce cas, la BANQUE perçoit des intérêts au taux nominal conventionnel. Le taux d'intérêt conventionnel est indiqué dans les Conditions Particulières/Contractuelles ou dans une convention spécifique.

⁵³² V., *infra*, §103

⁵³³ Comme l'échéance de remboursement, le montant maximal autorisé à être emprunté, le taux d'intérêt applicable..

performer ce qui a été convenu, sans qu'ils ne détiennent eux même une quelconque valeur juridique. L'écrit en langage naturel a toute primauté sur le programme et est le seul indicateur de la volonté des parties.

b) Une forme discrète dans le milieu de la *blockchain*

192. Rareté des contrats intelligents donnant primauté au texte écrit en langage naturel . Bien qu'existante, cette forme de contrat donnant primauté au document écrit en langage naturel est relativement rare dans le milieu de la *blockchain*. On dénombre, en effet, seulement quelques initiatives de juristes qui ont estimé que les smart contracts étaient avant tout des programmes pouvant être utilisés de la manière dont ils le sont habituellement dans l'exécution automatisée de contrats : en étant strictement soumis aux stipulations écrites qu'ils exécutent.

193. L'initiative Accord Project. Parmi les initiatives les plus connues implémentant cette forme, figure celle de l'entreprise *Clause*, qui fait partie des sociétés ayant reconnu très tôt l'opportunité de formaliser des véritables contrats dans la *blockchain*⁵³⁴. Elle propose une plateforme permettant la réalisation de *smart agreement* : des contrats qui sont connectés au Cloud et/ou à une *blockchain*, sur lesquels résident des programmes destinés à les exécuter⁵³⁵.

Elle a estimé que pour favoriser l'adoption de ces types de contrats, il était nécessaire de faire en sorte que l'industrie s'accorde sur des standards. Elle créa alors une association nommée *Accord Project*, chargée de développer et proposer au public des outils open-source permettant de réaliser des *smart agreement*.⁵³⁶ Le but étant que la communauté de juristes et développeurs s'emparent de cette technologie afin qu'elle devienne, *de facto*, la norme de réalisation de ces nouveaux types de

⁵³⁴ Clause. « Clause Joins Hyperledger ». Clause (blog), 22 mai 2017. <https://medium.com/clause-blog/clause-joins-hyperledger-38a10f8f3ea5>.

Clause is a NYC-based technology startup building software that will permanently transform the very nature of legal contracts into living, breathing, and integrated components of the digital world.(...). Distributed ledger technology is one piece of the puzzle when it comes to the future of legal contracting. A relatively new piece at that.

⁵³⁵ Kamal Hathi. « Taking the Next Step in Our Smart Agreement Journey », 27 mai 2021. <https://www.docuSign.com/blog/clause-docuSign-smart-agreement-journey>.

(...) "smart agreements", they incorporate computer code and smart clauses that effectively bring agreements to life—connecting different computer systems together, enabling actions to be taken automatically, and helping business to get done faster, more cost-efficiently, and with lower risk.

⁵³⁶ <https://accordproject.org/>

The Accord Project is a non-profit, collaborative, initiative developing an ecosystem and open source tools specifically for smart legal contracts.

contrats⁵³⁷.

L'outil proposé par *Accord Project*, et donc utilisé par *Clause*, consiste en un logiciel permettant de générer un contrat intelligent composé de trois éléments :

- un texte écrit en langage naturel correspondant au contrat écrit,
- ce qu'ils nomment un "modèle" qui est un programme contenant les variables du logiciel exécutant les contrats,
- un programme, codé dans un langage de programmation spécifique, exécutant la logique opérationnelle du contrat⁵³⁸.

Dans cet ensemble formant leur contrat intelligent, il y a une séparation nette entre le texte écrit, se situant en haut de la hiérarchie contractuelle, et les programmes. Le texte est l'élément juridique central du *smart agreement*, car c'est lui qui détaille le contenu du contrat et régit la relation. Il ne fait aucune mention des programmes chargés de rendre la convention connectée et auto-exécutante. Ces derniers ne sont qu'une modalité d'exécution au service de l'accord décrit en langage naturel.

194. *OpenLaw*. Une autre entreprise ayant adopté la forme de "primauté au texte écrit" est *OpenLaw*, une startup spécialisée dans l'édition de contrats sur la *blockchain Ethereum*⁵³⁹. Elle propose, comme *Accord Project*, un outil permettant la génération de contrats composés à la fois d'un document écrit en langage naturel⁵⁴⁰ et d'un *smart contract* associé, déployé sur *Ethereum*, et destiné à mettre en œuvre certaines stipulations du document.

Dans les contrats de *OpenLaw*, le texte écrit fait explicitement mention du *smart contract* comme un instrument d'exécution⁵⁴¹. Là aussi, il n'est question que d'un outil à l'aide duquel les parties

⁵³⁷ « FAQ · Accord Project ». Consulté le 27 avril 2023. <https://docs.accordproject.org/index.html>.

The purpose of the Accord Project is to establish and maintain a common and consistent legal and technical foundation for smart legal contracts.

⁵³⁸ « Accord Project · Documentation ». Consulté le 27 avril 2023. <https://docs.accordproject.org/index.html>.

Accord Project Templates are composed of three elements: the Text (the natural language), the Model (the data model), and the Logic (the executable business logic). When combined these three elements allow Accord Project templates to be both human-readable and machine-executable.

⁵³⁹ « OpenLaw ». Consulté le 27 avril 2023. <https://openlaw.io/>.

OpenLaw makes it easy to create legal agreements that work with Ethereum.

⁵⁴⁰ Plus exactement un modèle rempli via un formulaire.

⁵⁴¹ OpenLaw. « The Smart contract Stack ». Medium (blog), 24 septembre 2019.

exécuteront leurs contrats dans la *blockchain*. Le document textuel peut conférer des effets juridiques aux opérations dans la *blockchain*, mais il reste le référent principal de l'accord. En cas de contrariété entre le comportement du programme et ce qui est stipulé dans l'écrit en langage naturel, la volonté des parties n'est réputée prioritairement matérialisée que par ce dernier.

B – Un choix sécurisé mais porteur d'une certaine lourdeur

195. La préférence assumée de la sécurité. Cette forme de contrat a l'avantage d'être éprouvée et sécurisante (a), même si elle est porteuse d'une certaine lourdeur (b).

a) Les avantages de la forme donnant primauté au texte écrit

196. Une forme de contrat éprouvée. Cette forme de contrat a le premier avantage d'être éprouvée : elle est une manière de concevoir des conventions exécutées par des programmes qui existe depuis longtemps⁵⁴². Elle est donc non seulement soutenue par une pratique expérimentée et documentée mais elle est aussi connue des acteurs du monde du droit. Ces éléments participent à la rendre plus sécurisée⁵⁴³ et facilite son implémentation. L'expertise et l'expérience accumulées autour d'elle peuvent, en effet, être plus facilement mobilisées par des juristes et informaticiens qui ont l'habitude de réaliser ce genre de contrats. En sus, sa réception par des juges ou arbitres sera plus

<https://medium.com/@OpenLawOfficial/the-smart-contract-stack-5566ea368a74>.

Using OpenLaw, you can create a human-readable contract where both parties are able to understand the arrangement they are executing and have a signed record of the legal terms. (...). For example, think of a derivative interest swap agreement. Parties can cryptographically sign a derivative contract using OpenLaw's tools, trigger smart contracts on signature that pull digital assets into a smart contract-based escrow system that incorporates outside data related to interest rates to trigger transfers of payments between parties and post escrow into a smart contract.

⁵⁴² Surden, Harry. « Computable Contracts ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 2012. <https://papers.ssrn.com/abstract=2216866>.

⁵⁴³ François Xavier Testu. « Chapitre 31 – Liberté contractuelle, écrit et intitulé – Section 1 Question préliminaire : écrit ou pas écrit ? - §3 Ecrit comme exigence pratique - 31.18. Utilité de l'écrit », Dalloz référence Contrats d'affaires, 2010.

Quelles que soient les règles applicables, l'écrit est nécessaire en fait pour des raisons de sécurité juridique, car son existence donne aux parties l'occasion de clarifier leur volonté commune, et il évite un grand nombre de litiges. Il s'impose dès que l'opération économique en cause atteint le moindre degré de complexité ou suppose la stipulation de conditions qui ne sont pas d'usage.

aisée et prévisible⁵⁴⁴.

197. Une forme plus complète que la forme dépourvue de texte écrit. Surtout, dans cette forme, le texte écrit pourra combler tous les manques de la forme *code is law*. En effet, d’abord, il pourra servir à qualifier le contrat et préciser le régime juridique, en tant qu’instrument destiné à cet effet⁵⁴⁵. Puis il renseignera bien plus exhaustivement la volonté des parties ; qui aura toute primauté sur le programme. Autrement dit, si le code retranscrit mal la volonté des parties, cela aura une incidence limitée car le contrat n’est formalisé que par le texte écrit : les parties suivront sa lettre et modifieront, s’il le faut, le *smart contract* afin qu’il concorde avec lui⁵⁴⁶. Ainsi, la convention écrite constituera une sécurité additionnelle bienvenue face au risque de bogues et de piratage d’un *smart contract*.

b) Les inconvénients de la forme donnant primauté au texte écrit

198. Un document pouvant être perçu comme superflu. Comme évoqué, l’un des buts de la production d’un texte écrit est de conformer un contrat à un ordre juridique existant afin notamment de le rendre exécutable. Ainsi, l’écrit prouve clairement la relation et peut donc être présenté devant un juge qui pourra s’appuyer dessus pour forcer une des parties à respecter son engagement, ou engager leur responsabilité. Mais un *smart contract*, en tant qu’outil performatif, rend déjà exécutable toute convention à laquelle il est associé. Aussi, pour des parties y faisant recours avec cette conscience, rédiger un document écrit peut apparaître comme un effort superflu, surtout si la loi autorise le consensualisme et que le contrat est à faible valeur. En somme, ce peut être un effort supplémentaire incommode pour des parties se satisfaisant de la sécurité conférée par les smart

⁵⁴⁴ Cohnen, Shaanan, et David A. Hoffman. « Transactional Scripts in Contract Stacks ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 2020. <https://doi.org/10.2139/ssrn.3523515>.

(...), most judges are going to have a natural affinity for text that they can read without the aid of an expert translator. They will argue that “no one reads smart contracts.”

⁵⁴⁵ François Xavier Testu. « Chapitre 31 – Liberté contractuelle, écrit et intitulé – Section 1 Question préliminaire : écrit ou pas écrit ? - §3 Ecrit comme exigence pratique - 31.18. Utilité de l’écrit », Dalloz référence Contrats d’affaires, 2010.

Quelles que soient les règles applicables, l’écrit est nécessaire en fait pour des raisons de sécurité juridique, car son existence donne aux parties l’occasion de clarifier leur volonté commune, et il évite un grand nombre de litiges.

⁵⁴⁶ Cohnen, Shaanan, et David A. Hoffman. « Transactional Scripts in Contract Stacks ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 2020. <https://doi.org/10.2139/ssrn.3523515>.

All contractware has this property: it is wrapped within an ordinary, legal contract. Absent a legitimate connection to those contracts, use of a credit card can still affect the world—your account will be debited, the merchant’s credited—but such changes can be quickly reversed.

contracts.

199. Un recours pouvant équivaloir à une perte d'opportunité. De manière générale, la forme donnant primauté au texte est porteuse d'une certaine lourdeur qui peut remettre en question l'opportunité de recourir à des smart contracts. En effet dans celle-ci, les parties doivent s'assurer que leurs programmes soient le reflet des stipulations écrites, sans quoi elles devront les faire évoluer pour qu'ils soient conformes à ce qui a été écrit⁵⁴⁷. Cette situation est bien plus lourde à mettre en œuvre que si une certaine autonomie leur était laissée pour gérer toute la relation, comme dans les contrats représentés que par du code ou ceux concurremment matérialisés par du texte et du code que nous verrons ultérieurement⁵⁴⁸.

Dans ces dernières, les parties n'ont besoin de produire que le code du programme pour garantir l'exécution de leur contrats et ne se soumettent qu'à lui pour régir leur relation. Elles l'utilisent ainsi au maximum de son potentiel. Contrairement à la forme où le texte écrit prime, les parties ne peuvent pas constamment contester ses résultats du programme et donc nuire à l'opportunité d'y recourir. Tandis qu'en utilisant les smart contracts comme des simples programmes constituant une modalité comme une autre d'exécution d'un accord, elles se privent, il pourrait être argué, de leur atout principal⁵⁴⁹.

§ II - Le texte en tant qu'instrumentum complémentaire avec le code

200. Code and law. Dans certaines formes de contrats intelligents, le texte écrit cadrant la

⁵⁴⁷ Blycha, Natasha, et Ariane Garside. « Smart Legal Contracts - A Model for the Integration of Machine Capabilities into Contracts ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 7 décembre 2020. <https://papers.ssrn.com/abstract=3743932>.

For example, for one particular obligation the contracting parties may agree that the traditional natural language term contains the primary legal obligation. In this case, when the corresponding Incorporated Code fails, it is not to be considered a breach of contract as the contracting parties explicitly agree that in the case that the code fails, the contracting parties should use alternative methods to ensure the legal obligation held in the natural language term is fulfilled.

⁵⁴⁸ V., *supra*, §200

⁵⁴⁹ Hinkes, Andrew. « The Limits of Code Deference ». SSRN Scholarly Paper. Rochester, NY, 19 juillet 2021. <https://papers.ssrn.com/abstract=3889630>.

(...) given their flexibility, smart contracts enable perhaps the purest expression of private law. Smart contracts provide a hyper-flexible framework for agreement with a fully integrated, alegal enforcement mechanism., Smart contracts thus provide the tools necessary to implement an independent private system of order.

relation contractuelle donne également au *smart contract* le rôle de régir entièrement ou quelques aspects de la relation. Il en résulte une convention gouvernée à la fois par de la prose juridique, écrite en langage naturel, et du code informatique. Nous appelons cette forme *code and law*⁵⁵⁰. Dans celle-ci, le texte écrit et le programme du contrat intelligent reflètent tous deux la volonté des parties. Le code est plus qu'une simple modalité d'exécution des clauses écrites, il indique, avec la même force que le texte écrit, l'intention des parties. Ces dernières ont fait le choix de laisser régir leur relation, au moins en partie, par un programme autant que par des stipulations écrites. Cette forme de contrat intelligent a été mise en avant par un juriste attaché à la philosophie du milieu de la *blockchain* (A), parce qu'elle gommerait les plus grands défauts de la forme dépourvue de texte écrit tout en retenant ses qualités principales (B).

A – Genèse de l'idée

201. Les EDI (échange de données informatisé). Les contrats d'interchange sont souvent cités comme précurseurs de cette forme de contrats⁵⁵¹. Ils consistent en des conventions destinées à encadrer les relations entre des parties faisant recours aux EDI (échange de données informatisé). Ces dernières correspondent à une technologie permettant l'échange instantané de données dans un format électronique convenu entre deux partenaires commerciaux⁵⁵². Imaginons deux entreprises qui

⁵⁵⁰ Norton Rose. « Can smart contracts be legally binding contracts? An R3 and Norton Rose Fulbright White Paper », novembre 2016.

a “split” contract where non-human performance is encoded into computer code, and wider human obligations, remedial and other provisions are written into natural language, the two components operating together as a cohesive contract.

⁵⁵¹ Hinkes, Andrew. « The Limits of Code Deference ». SSRN Scholarly Paper. Rochester, NY, 19 juillet 2021. <https://papers.ssrn.com/abstract=3889630>.

Parties contracting using EDI expressly agreed not to contest the “validity or terms of [agreements] on the basis that they were concluded by EDI, that the original records were in electronic form, or that no signature(s) evidence such [agreements]. (...) Notwithstanding this limitation, EDI agreements are an example of parties who rely upon technology to form enforceable relationships “using a paper contract to create an ‘interpretive regime’ to govern the particular risks of their transaction,” and successfully bind themselves to it.

Szabo, Nick. « Smart contracts: Building Blocks for Digital Markets », 1996. https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html.

The field of Electronic Data Interchange (EDI) (...) can be viewed as a primitive forerunner to smart contracts.

⁵⁵² « Ce Qu'est EDI (Échange de Données Informatisé)? | EDI Pour Tous », 20 septembre 2019. <https://www.edipourtous.fr/ce-qu-est-l-edi/>.

L'Échange de données informatisé (EDI) est un échange ordinateur-à-ordinateur de documents commerciaux dans un format électronique standard entre les partenaires commerciaux.

s'entendent pour que les commandes de leurs marchandises soient effectuées via EDI.

Grâce à ce système, au lieu que chaque commande soit effectuée manuellement ou par l'envoi d'un mail ou un fax, les parties choisissent de s'accorder sur un format de données spécifique pour communiquer toutes les informations relatives à un ordre de transaction. Une partie A pourra prévoir que lorsque son logiciel interne détectera un stock faible, celui-ci enverra automatiquement un ordre d'achat dans le format de donnée convenu afin que le logiciel de la partie B le comprenne et lance l'envoi automatique de la marchandise. Ce procédé de communication s'appelle l'EDI et permet ainsi de rationaliser le processus de commande de biens en ne faisant intervenir aucun humain, ce qui a aussi pour effet de le sécuriser⁵⁵³.

Dans ces relations, un contrat cadre écrit stipule que chaque commande devra être effectuée par le biais de l'EDI. Autrement dit, le document contractuel délègue explicitement aux programmes tout le processus de conclusion des contrats d'applications de vente. Aussi, le code se retrouve être autant un instrument performatif, qui exécute la commande de biens, qu'un élément du corpus contractuel au même niveau que les stipulations écrites : ce n'est que par son biais que les parties peuvent faire commande et se soumettent à son résultat⁵⁵⁴.

202. La déférence au code de Gabriel Shapiro Inspiré par cette forme de contrat, le juriste spécialiste de la *blockchain* Gabriel Shapiro proposa en 2018 une alternative aux formes sans textes écrits et de "primauté à l'écrit" des contrats intelligents. En tant que membre actif de la communauté *blockchain*, et donc très attaché à l'idéologie *cypherpunk*⁵⁵⁵, il regrettait que les juristes de ce milieu ne proposaient que la forme donnant toute primauté au texte écrit comme alternative à la forme qui en est dépourvu. Selon lui, celle-ci reléguait le *smart contract* à un rôle trop minime, sous exploitant ses capacités⁵⁵⁶.

Pour lui, l'opportunité d'utilisation de ces programmes était la plus grande lorsqu'ils étaient utilisés

⁵⁵³ Ibid.

(...) les entreprises bénéficient d'avantages significatifs tels que la réduction des coûts, l'amélioration de la vitesse de traitement, la diminution des erreurs et l'amélioration des relations avec leurs partenaires commerciaux."

⁵⁵⁴ Hinkes, Andrew. « The Limits of Code Deference ». SSRN Scholarly Paper. Rochester, NY, 19 juillet 2021. <https://papers.ssrn.com/abstract=3889630>.

Notwithstanding this limitation, EDI agreements are an example of parties who rely upon technology to form enforceable relationships "using a paper contract to create an 'interpretive regime' to govern the particular risks of their transaction, " and successfully bind themselves to it

⁵⁵⁵ V., *infra*, §177

⁵⁵⁶ Twitter. « _gabrielShapir0 sur Twitter », 12 août 2018. https://twitter.com/lex_node/status/1028727472311873537.

comme outils se substituant au au corps judiciaire garantissant l'exécution de contrats. Or, dans la forme donnant primauté au texte écrit, ils ne sont utilisés qu'en tant qu'outils performant une volonté formalisée par un texte rédigé en langage naturel. Les parties peuvent alors être tentées de vouloir constamment contester le résultat du programme en prétendant qu'il n'implémente pas fidèlement leur volonté écrite ; ce qui nuirait à l'opportunité de recourir aux smart contracts.⁵⁵⁷

Ainsi, selon Gabriel Shapiro, ce modèle ne répondait pas assez aux aspirations autonomistes des personnes attachées à l'éthos *cypherpunk*⁵⁵⁸. Mais en même temps, en tant que juriste, il reconnaissait que la forme de contrat intelligent dépourvu de tout texte écrit en langage naturel était trop insécurisée juridiquement⁵⁵⁹. Il proposa donc une voie intermédiaire : un contrat intelligent avec un document écrit rédigé classiquement, qui toutefois délègue à un *smart contract* le soin de régir des larges aspects de la relation. La philosophie de ce modèle est donc celle de *code is law* où les parties font le choix de se soumettre, dans une grande mesure, au *smart contract* qu'elles auront réalisé, nonobstant le comportement de ce dernier⁵⁶⁰.

Le rôle du texte écrit dans cette forme est de prévoir les cas exceptionnels dans lesquels le *smart contract* ne s'appliquera plus, et donc le moment à partir duquel le texte reprendra la main sur la

If the smart contract can be unilaterally halted, then you are essentially back to relying on traditional methods of deal certainty--a legalese agreement + the ability to enforce it through ADR and/or courts. That system is not a "bad thing" but falls far short of the cypherpunk dream that people thought smart contracts would deliver in the early-ish days of #Ethereum.

⁵⁵⁷ Twitter. « _gabrielShapir0 sur Twitter », 12 août 2018. https://twitter.com/lex_node/status/1028727472311873537.

To me that is the big value-add of deploying smart contracts on a public blockchain-based world computer[draconian result of smart contracts]. It would be a shame if we lost all possibility of parties harnessing that benefit, and we lawyers should try to find ways to preserve it for those who might want it.

⁵⁵⁸ Twitter. « _gabrielShapir0 sur Twitter », 19 juillet 2018. https://twitter.com/lex_node/status/1019819161298456577.

1/Kernel of truth in "code is law": Smart contracts improve finality/efficiency only if we can legally defer to/rely on their results. If we always have to check the code against words and words always win, smart contracts can't REPLACE traditional contracts, only perform them.

⁵⁵⁹ Ibid.

2/ But if we want to defer to smart contracts, since "code is law" is crazy, we need a rule that says the results of running the smart contract are legally binding EXCEPT in the event of x, y or z. And we need to say that in words, in a traditional legal contract.

⁵⁶⁰ _g4brielShapir0. « SCoDA – Simple Code Deference Agreement », 25 avril 2023. <https://github.com/lex-node/SCoDA-Simple-Code-Deference-Agreement->.

This is a (...) Simple Code Deference Agreement. It is designed to provide a model for the type of legal or 'wet' contract that may be entered into by two parties who wish to otherwise agree that 'code is law' with respect to a certain arrangement between them. In this case, the 'code' is a smart contract deployed to Ethereum.

détermination de la relation. En effet, dans le modèle proposé par Shapiro, les parties ne devraient pas se résigner à la philosophie extrême du *code is law* lorsque, par exemple, la *blockchain* sur laquelle le *smart contract* est déployé est complètement compromise⁵⁶¹. En cas d'événements de force majeure, elles doivent pouvoir gérer les conséquences de ces événements sur leur relation en se reposant sur ce qui été stipulé en langage naturel.

B – Un compromis entre praticité et sécurité

203. Avantages de la forme de déférence au code. Comme évoqué, l'avantage principal de l'approche de déférence au code de Gabriel Shapiro est qu'elle opère un compromis relativement minime à la forme sans texte écrit. Elle en retient donc les qualités principales : débridement⁵⁶² et commodité tout en apportant plus de sécurité juridique à la relation. Les partisans de l'idéologie *code is law* disposent, avec cette forme, d'une alternative plus complète et sécurisée que la forme sans texte écrit, et plus autonome que celle donnant "primauté à l'écrit en langage naturel" où le *smart contract* n'a aucune valeur juridique⁵⁶³.

Un autre avantage de cette forme est qu'elle est plus simple à mettre en œuvre que la forme de donnant primauté au texte écrit. Le modèle de Shapiro fait une déférence au code : ce dernier est chargé de régir une grande partie de la relation à la place du texte stipulé. Aussi, il n'y a pas besoin d'écrire toutes les obligations du contrat puis de toutes les traduire en code et ensuite de veiller à une concordance entre ces deux. Dans cette forme *code and law*, les deux opèrent chacun sur le même plan juridique mais régissent différents aspects du contrat⁵⁶⁴.

204. Inconvénients de la forme de déférence au code. De la même manière que cette

⁵⁶¹ Ibid.

Rather, Qualified Code Deference contemplates that the SCoDA is designed to be a legally binding contract whereby the parties agree that, EXCEPT in certain narrow circumstances, they will DEFER to (i.e., refrain from disputing in a legal proceeding) the results of operation of a smart contract. (...) The exceptional circumstances under which the parties are not required to defer to the smart contract are captured under the concept of a "Material Adverse Exception Event." These circumstances are basically the blockchain equivalent of 'force majeure' events—e.g., a 51% attack that causes a double-spend somehow affecting the smart contract. ...

⁵⁶² V., *infra*, §184

⁵⁶³ V., *infra*, §191

⁵⁶⁴ L'ISDA décrit cette approche comme le modèle interne : le contrat intelligent est composé à la fois d'écritures en langage naturel et de codes.

Clack, Christopher D., et Ciaran McGonagle. « Smart Derivatives Contracts: the ISDA Master Agreement and the automation of payments and deliveries ». arXiv, 1 avril 2019. <http://arxiv.org/abs/1904.01461>.

The internal model has the potential to bring significant additional benefits to Smart Derivatives Contracts, for example in improving the fidelity of the smart contract code to the contract.

forme retient les qualités principales de la forme sans texte écrit, elle en garde aussi ses défauts principaux : les parties sont très vulnérables au risque de bogues et piratages du *smart contract*⁵⁶⁵. Si ce dernier défaille, ou dit autrement, qu'il n'exécute pas fidèlement la volonté des parties, les parties ne seront pas en mesure de revenir sur leur document écrit car elles y ont précisément prévu qu'elles se soumettent au résultat du *smart contract*⁵⁶⁶. Seulement dans des hypothèses extrêmement rares, l'écrit pourra prendre le dessus sur le *smart contract*.⁵⁶⁷ Mais comme évoqué, ces défauts n'en sont pas véritablement pour les tenants de l'idéologie *code is law* : ils en ont conscience et les voient comme la juste contrepartie des bénéfices apportés par les smart contracts⁵⁶⁸.

205. Conclusion de la section II et du chapitre I.

Conclusion de la section II. En marge des contrats intelligents dont le seul *instrumentum* est le code informatique, il en est observé d'autres où un texte écrit en langage naturel matérialise l'intention des parties. On distingue ceux qui correspondent au modèle habituel de contrat automatisé dont nous faisons régulièrement l'expérience dans le monde *fiat* : là, le *smart contract* n'est qu'un instrument performatif des stipulations du contrat, une simple modalité d'exécution du rapport décrit en langage naturel. Et ceux où *l'instrumentum* du contrat est autant formé de stipulations écrites que de code informatique ; le but étant de conférer un plus grand rôle aux smart contracts, afin de s'exposer autant que la sécurité juridique le permet, à ses propriétés.

Conclusion du chapitre I. Il est donc distingué entre deux formes de contrats intelligents dans le milieu de la *blockchain*. La forme sans texte écrit en langage naturel, qui est la plus répandue et simple à constituer bien que porteuse d'une grande insécurité juridique. Et celle des contrats intelligents

⁵⁶⁵ V., *infra*, §187

⁵⁶⁶ _g4brielShapir0. « SCoDA – Simple Code Deference Agreement », 25 avril 2023. <https://github.com/lex-node/SCoDA-Simple-Code-Deference-Agreement->.

Rather, QDC contemplates that the SCoDA is designed to be a legally binding contract whereby the parties agree that, EXCEPT in certain narrow circumstances, they will DEFER to (i.e., refrain from disputing in a legal proceeding) the results of operation of a smart contract.

⁵⁶⁷ _g4brielShapir0. « SCoDA – Simple Code Deference Agreement », 25 avril 2023. <https://github.com/lex-node/SCoDA-Simple-Code-Deference-Agreement->.

The exceptional circumstances under which the parties are not required to defer to the smart contract are captured under the concept of a "Material Adverse Exception Event."

⁵⁶⁸ V., *infra*, §178

porteurs d'un écrit en langage naturel, où il est encore possible de sous-catégoriser entre les contrats dans lesquels le texte en langage naturel a toute primauté sur le *smart contract* et ceux où il a un rôle concurrent avec le programme. Les premiers sont une forme plus éprouvée et sécurisante pour les parties, bien que relativement lourde à mettre en œuvre. Les seconds sont des versions moins brutales des contrats à la forme *code is law*, où le *smart contract* régit, dans une large mesure, la relation contractuelle.

206. La forme donnant primauté au texte en langage naturel. Laquelle des différentes formes de contrats intelligents observables doit donc être choisie par des parties ? Nous estimons que la plus adéquate est la forme que nous nommons « ricardienne », où un texte écrit en langage naturel a toute primauté sur les autres éléments du contrat intelligent et lui est incorporé de façon spécifique (Section I). Le corpus qui résultera de cette forme devra être conçu d'une façon particulière compte tenu de la spécificité des contrats intelligents (Section II).

Section I – Le choix de la méthode ricardienne

207. Le contrat ricardien. Le contrat ricardien est une forme de contrat intelligent dans laquelle l'accord rédigé dans un langage naturel a primauté sur les autres éléments du corpus du contrat et est lié de manière originale aux smart contracts l'exécutant (§1) ; il est proposé une manière spécifique de l'implémenter dans notre contexte (§2).

§ I - Définition du contrat ricardien

208. Concept de contrat ricardien. Le concept de contrat ricardien est ancien et a toujours été présenté comme une solution complémentaire aux smart contracts (A) ; aujourd'hui il désigne surtout les contrats intelligents dotés de textes écrits liés par un mécanisme de *hash* aux smart contracts les exécutant (B).

A – Origine du concept

209. Genèse du contrat ricardien. Le contrat ricardien a été inventé en 1996 par le financier-cryptographe Ian Grigg. Il s'agissait, à l'origine, d'un document identifiant un instrument financier échangeable dans des systèmes numériques. Le fonctionnement de ces derniers était le suivant : un logiciel permet à des individus de s'échanger des titres financiers numérisés (comme des produits dérivés, des obligations...) ; chacun d'eux est identifiable par l'emprunte numérique, le *hash*,

du document juridique qui fondent leur existence⁵⁶⁹. Par exemple, chaque produit dérivé numérisé est identifié par une suite de caractères qui correspond au condensat du contrat dérivé écrit en langage naturel dont ils sont issus⁵⁷⁰.

210. Propriétés du contrat ricardien. Ce document est un fichier numérique contenant un texte écrit en langage naturel. Selon Ian Grigg, il est notamment doté des propriétés suivantes⁵⁷¹ :

- il stipule un véritable contrat entre l'émetteur de l'actif et ses détenteurs,
- le document est lisible par un humain ordinaire mais des éléments de son corpus peuvent aussi être interprétés par des programmes,
- il est signé numériquement,
- il contient des informations sur le système d'échange auquel il est lié,
- enfin il est *hashé* .

⁵⁶⁹ Ian Grigg. « The Ricardian Contract », 1996. https://iang.org/papers/ricardian_contract.html.

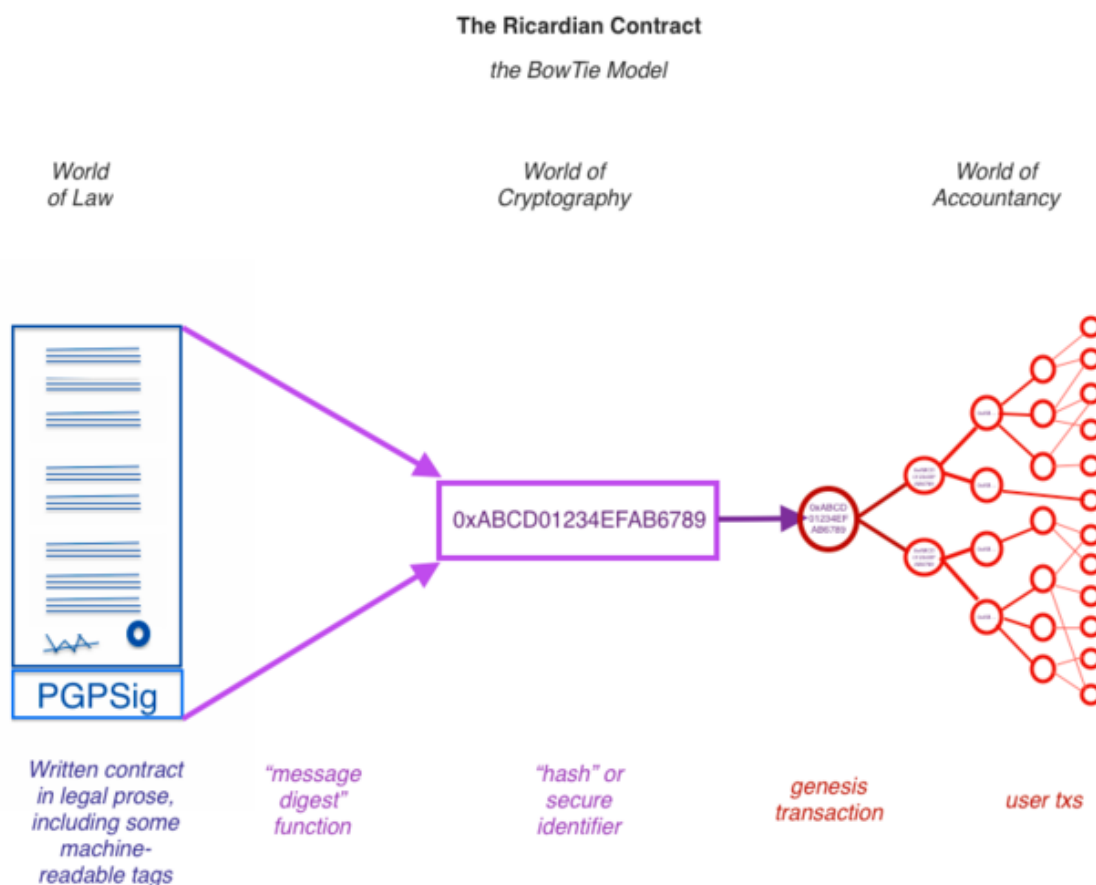
As bonds are, at their essence, contracts between issuers and users, our problem reduces to one of issuing contracts. Whereas other issues have contracts, our issues are contracts. Our innovation is to express an issued instrument as a contract, and to link that contract into every aspect of the payment system. By this process, a document of some broad utility (readable by user and program) is drafted and digitally signed by the issuer of the instrument. This document, the Ricardian Contract, forms the basis for understanding an issue and every transaction within that issue. By extension, all issues of value, such as currencies, shares, derivatives, loyalty systems and vouchers, can benefit from this approach.

⁵⁷⁰ Ibid.

In the simplest possible terms, a Ricardian Contract is a document defining a type of value for issuance over the Internet. It identifies the Issuer; being the signatory; and any terms and clauses the Issuer sees fit to add in to make the document stand as a contract.

⁵⁷¹ Ibid.

A Ricardian Contract can be defined as a single document that is a) a contract offered by an issuer to holders, b) for a valuable right held by holders, and managed by the issuer; c) easily readable by people (like a contract on paper), d) readable by programs (parsable like a database), e) digitally signed, f) carries the keys and server information, and g) allied with a unique and secure identifier.



572

211. Différence avec les smart contracts de Nick Szabo. Les contrats ricardiens sont une invention contemporaine des smart contracts de Nick Szabo. Des interrogations furent donc émises sur les similarités et différences entre ces deux. Dans un article, Ian Grigg lui-même s'adonne à cet exercice de comparaison et souligne la complémentarité entre *smart contract* et contrats ricardiens⁵⁷³. Selon lui, les smart contracts sont des outils proposant une véritable alternative aux moyens classiques d'exécution des contrats⁵⁷⁴. Ils visent à permettre l'émergence d'un système alégal où les transactions entre individus sont garanties d'exécution de manière totalement autonome de notre système juridique

⁵⁷² Oliveira, Ludmila, Felipe Simoyama, Ian Grigg, et Ricardo Luiz Bueno. « Triple entry ledgers with blockchain for auditing ». *International Journal of Auditing Technology* 3 (1 janvier 2017): 163.
<https://doi.org/10.1504/IJAUDIT.2017.10007789>.

⁵⁷³ Ian Grigg. « On the intersection of Ricardian and Smart contracts », février 2015.
https://iang.org/papers/intersection_ricardian_smart.html.

⁵⁷⁴ Ibid.

(...) *the smart contract is really the machine to perform the contract.*

classique. Les smart contracts sont donc non seulement des instruments performatifs, mais en plus, n'ont pas vocation à s'intégrer avec le droit positif⁵⁷⁵.

Tandis que les contrats ricardiens présentent les caractéristiques inverses. Ils visent à être de vrais actes juridiques qui peuvent être traités, dans une certaine mesure, par des programmes mais n'ont pas vocation à servir de moyens d'exécution⁵⁷⁶. Leur caractéristique est de décrire extensivement un rapport juridique. Ils ont précisément pour but de donner une sécurité juridique aux actifs échangés dans le monde de la cryptographie ; ils ne cherchent pas à se passer du monde *fiat* mais à se concilier avec lui. Aussi, Ian Grigg explique que les contrats ricardiens et les smart contracts peuvent combler leurs lacunes respectives. Le contrat ricardien peut servir à décrire précisément le rapport juridique que le *smart contract* ne se contente d'exécuter ; puisque pour rappel, ce dernier est un médiocre instrument pour le décrire⁵⁷⁷. Tandis que les smart contracts serviront à garantir que l'accord écrit du document ricardien sera performé⁵⁷⁸.

⁵⁷⁵ Hinkes, Andrew. « The Limits of Code Deference ». SSRN Scholarly Paper. Rochester, NY, 19 juillet 2021. <https://papers.ssrn.com/abstract=3889630>.

Smart contracts provide a hyper-flexible framework for agreement with a fully integrated, alegal enforcement mechanism. Smart contracts thus provide the tools necessary to implement an independent private system of order.

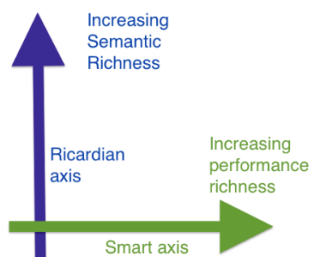
⁵⁷⁶ Ian Grigg. « On the intersection of Ricardian and Smart contracts », février 2015. https://iang.org/papers/intersection_ricardian_smart.html.

In terms of defining differences, the Ricardian Contract works well to describe and differentiate shares, bonds, derivatives, more or less anything that means something to a human. Indeed, a Ricardian Contract is conceptually unlimited in the richness of semantics (...). Likewise, the Ricardian Contract is a clumsy vehicle in which to insert difficult code.

⁵⁷⁷ V., *infra*, §186

⁵⁷⁸ Ibid.

In contrast, it introduces the smart contract which is a design to capture the flow of actions and events (e.g., delivery of payments) within the performance of a contract.(...). We can now see that the real challenge between smart contracts and Ricardian Contracts or legal documents is not to choose, but to incorporate. The Bitcoin world will benefit from adding the semantic richness of legal documentation into its service.



579

B – Le contrat ricardien aujourd’hui

212. Utilisation contemporaine du contrat ricardien. Il ne se trouve pas de contrats ricardiens comportant exactement toutes les propriétés que Ian Grigg avait originellement défini. Cependant, on dénombre plusieurs contrats intelligents qui incluent certaines de ces propriétés. En particulier celle de la liaison par *hash* que nous avons évoquée : le contrat intelligent est composé d’un texte écrit en langage naturel décrivant le rapport contractuel, puis est *hashé* et enregistré dans les smart contracts exécutant cette convention⁵⁸⁰. *Smart contract* et document ricardien sont ainsi liés comme les produits financiers et leurs contrats dans les systèmes d’échanges qui ont vu naître le concept de contrat ricardien⁵⁸¹.

Les contrats intelligents proposés par les outils d’Accord Project et de OpenLaw⁵⁸² ont des propriétés ricardiennes⁵⁸³. Chacun d’eux est en effet composé d’un document électronique écrit en langage

⁵⁷⁹ Ian Griggs, « On the intersection of Ricardian and Smart contracts », février 2015. https://iang.org/papers/intersection_ricardian_smart.html.

⁵⁸⁰ Clause, « REALLY Smart (and Legal!) Contracts », Clause (blog), 28 mars 2018. <https://medium.com/clause-blog/really-smart-and-legal-contracts-a77fcd1d0d10>.

The data for a Smart Clause is represented as JSON and can be hashed. The contents of a Smart Clause Template can also be hashed. These capabilities ensure that content-based hashing can be used to unambiguously link a smart legal contract to executable code or vice-a-versa.

⁵⁸¹ V., *infra*, §209

⁵⁸² V., *infra*, §193, §194

⁵⁸³ Clause, « REALLY Smart (and Legal!) Contracts », Clause (blog), 28 mars 2018. <https://medium.com/clause-blog/really-smart-and-legal-contracts-a77fcd1d0d10>.

Although the Accord Project Template Specification is not directly Ricardian, it does have Ricardian properties.

OpenLaw, « The Smart contract Stack », Medium (blog), 24 septembre 2019. <https://medium.com/@OpenLawOfficial/the-smart-contract-stack-5566ea368a74>.

naturel, dont certains éléments sont encapsulés dans des balises HTML⁵⁸⁴, et de *smart contract* qui contiennent les *hash* de ces stipulations écrites qu'ils exécutent. L'organisation *moleculeDAO*, pour ne citer qu'elle, qui propose la tokenisation d'actifs de propriété intellectuelle en NFT utilise aussi cette méthode pour lier ses actifs aux contrats de cession de droit d'auteur⁵⁸⁵.

§ II – Recours à la méthode ricardienne

213. L'opportunité du recours à la méthode ricardienne. Il est donc recommandé aux parties d'adopter la forme ricardienne pour leur contrat intelligent afin de lui fournir la sécurité juridique nécessaire (A). Il sera avisé d'implémenter cette forme d'une certaine manière (B).

A – Un recours justifié par la sécurité juridique

214. Rappels des inconvénients de l'absence de texte rédigé en langage naturel. Nos précédents développements ont fait ressortir qu'il est difficile, avec la forme dépourvue de texte écrit en langage naturel, de prouver le rapport contractuel⁵⁸⁶. Et malgré la promesse idéologique derrière cette forme, celle-ci ne s'abstrait pas d'une réception juridique⁵⁸⁷. A défaut de stipulations écrites, les parties prendront le risque de la perte de la maîtrise du régime juridique opportun applicable à leur relation⁵⁸⁸. Enfin, une telle forme dépourvoit les parties de la faculté de s'appuyer sur un instrument

OpenLaw brings the vision of ricardian contracts to life, providing the tools to fulfill the initial vision that Ian Grigg set out several decades ago. Using OpenLaw, you can create a human-readable contract where both parties are able to understand the arrangement they are executing and have a signed record of the legal terms.

⁵⁸⁴ Une balise HTML est une commande spéciale utilisée dans la création de pages Web pour donner des instructions au navigateur sur la façon dont le contenu de la page doit être affiché. Elles sont écrites dans du code HTML et sont entourées de chevrons.

⁵⁸⁵ Clemens Ortlepp. « Molecule's Biopharma IP-NFTs - A Technical Description », 6 août 2021. <https://www.molecule.to/blog/molecules-biopharma-ip-nfts-a-technical-description>.

Molecule has set up a license agreement for the "Longevity Molecule" with the Scheibye-Knudsen Lab of the University of Copenhagen. The agreement (as a legal document) is created and will be cryptographically signed via a transaction on the Ethereum Blockchain. A record of the transaction will be added to the sub-license agreement in form of a cryptographic hash. After the signing process is completed, an NFT is being created and the signed document is added to the metadata JSON of the IPNFT.

⁵⁸⁶ V., *infra*, §186

⁵⁸⁷ Leveneur, Claire. « Les smart contracts : étude de droit des contrats à l'aune de la blockchain ». These de doctorat, Université Paris-Panthéon-Assas, 2022. <https://www.theses.fr/2022ASSA0063>

En définitive, il n'est pas raisonnable de penser pouvoir faire échapper la blockchain et ses applications au droit. La blockchain existe dans un monde gouverné par le droit (...). La blockchain est donc, dans ses applications, soumise au droit. Le code informatique, qui structure la blockchain, ne peut remplacer la loi.

⁵⁸⁸ V., *infra*, §185

utile pour maîtriser le risque de bogue de *smart contract*⁵⁸⁹. Un texte écrit répond donc à un besoin basique de sécurité juridique, que la forme ricardienne fournit⁵⁹⁰.

215. La nécessaire primauté du texte rédigé en langage naturel. Comme le risque de bogues dans un *smart contract* est important et perpétuel⁵⁹¹, nous estimons également que les parties doivent donner une importance prépondérante au texte écrit pour maîtriser cette variable. En effet, en le plaçant devant tout autre élément du contrat intelligent, elles instituent ses stipulations comme le réel indicateur de leur volonté en cas de défaillance du programme exécutant⁵⁹². Elles pourront donc s'appuyer sur elles pour maîtriser, autant que possible, les conséquences d'un *bug* dans leur contrat intelligent⁵⁹³. Cela n'est pas à dire que le texte écrit lui-même sera insusceptible de « bogues »⁵⁹⁴ ; mais il y sera nettement moins sujet que des smart contracts⁵⁹⁵. Aussi, nous estimons qu'il est tout simplement plus rationnel pour des parties de donner primauté à leurs stipulations écrites dans l'interprétation de leur contrat intelligent⁵⁹⁶ ; et c'est ce qui justifie, à nouveau, le recours à la forme ricardienne.

⁵⁸⁹ V., *infra*, §187

⁵⁹⁰ Ian Griggs. « On the intersection of Ricardian and Smart contracts », février 2015.
https://iang.org/papers/intersection_ricardian_smart.html.

Smart contracts then can capture unlimited richness in flows of actions and events; computer scientists might prefer to recognise this as a state machine with money. But what is not captured is the semantics: what is the project? What will it do? How do we know that the contributions are going to our project to design the \$100 solar widget to reverse global warming? Or the pension fund for a drug kingpin? How do we even know it is a crowd funding? What do we do when our money doesn't come back or our project deliverables fail?

⁵⁹¹ Le risque de bug de *smart contract* est d'une autre ampleur que s'il s'agissait de programmes non immuables car ces derniers manipulent de la valeur. Mais en tant que programmes, ils seront toujours sujets à des défaillances.

⁵⁹² De Filippi, Primavera, et Aaron Wright. *Blockchain and the Law: The Rule of Code*. Harvard University Press, 2018.
<https://www.jstor.org/stable/j.ctv2867sp>.

If smart contracts are used to model legal agreements, parties can create hybrid arrangements that blend natural-language contracts with smart contracts written in code(...). This approach allows natural-language agreements and smart contracts to work hand-in-hand to memorialize the parties' intent.

⁵⁹³ Mekki, Mustapha. « Le smart contract, objet du droit (Partie 2) », *Dalloz IP/IT*, 2019, 27.

Si le smart contract réduit certains risques, il en crée de nouveaux qu'il convient d'encadrer par des clauses contractuelles qui ne peuvent être algorithmées.

⁵⁹⁴ Par « bugs », nous entendons, dans le contexte d'un contrat, les erreurs de rédaction.

⁵⁹⁵ Chu, Hanting, Pengcheng Zhang, Hai Dong, Yan Xiao, Shunhui Ji, et Wenrui Li. « A Survey on Smart contract Vulnerabilities: Data Sources, Detection and Repair ». *Information and Software Technology* 159 (1 juillet 2023): 107221. <https://doi.org/10.1016/j.infsof.2023.107221>.

⁵⁹⁶ Blycha, Natasha, et Ariane Garside. « Smart Legal Contracts - A Model for the Integration of Machine Capabilities into Contracts ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 7 décembre 2020.
<https://papers.ssrn.com/abstract=3743932>.

216. Atténuation de l'opportunité de recours aux smart contracts. Dans cette situation, le *smart contract* serait inféodé au texte écrit et donc utilisé comme la plupart des programmes ordinaires exécutant des clauses de contrats le sont : en tant que simple outil performatif. Il pourrait être argué qu'il soit regrettable de faire une telle sous-exploitation de cette technologie : en soumettant les smart contracts à un écrit, ils semblent privés de leur proposition de valeur principale⁵⁹⁷. Or selon nous, cette exploitation prétendument insuffisante ne fait pas perdre l'opportunité de son recours. Pour rappel, les smart contracts sont des outils uniques de manipulation d'actifs numériques, fonctionnant dans un environnement, en principe, robuste, disponible, transparent, libre d'accès et désintermédié⁵⁹⁸. Ces atouts font qu'il n'y a pas besoin qu'ils servent, en plus, de substitut total aux mécanismes d'exécution de notre ordre juridique, pour qu'ils soient intéressants d'y recourir. En tant qu'outils automatisant ou exécutant la manipulation d'actifs, ils sont strictement meilleurs que des programmes ordinaires et peuvent donc être, opportunément, utilisés qu'à cette seule fin⁵⁹⁹.

B - Implémentation

217. Inclusion de propriétés ricardiennes. Les propriétés suivantes de la forme ricardienne sont donc proposées d'être incluses dans le contrat intelligent des parties :

- **Document lisible par des humains et des programmes.** Les parties seront donc avisées de rédiger un contrat en langage naturel pour décrire leur relation contractuelle.

We propose that in the short to medium term, a sensible legal approach to initial use of SLCs is to ensure that coded terms are not drafted in isolation from natural language expressions of those clauses. Any coded provision contained in an SLC can be linked to one or a number of natural language counterparts clearly expressing the obligation and the contracting parties' intent that can be understood by all and visa versa any natural language component in an SLC can be linked (or conjoined) to a coded provision or provisions in the SLC.

⁵⁹⁷ Twitter. « _gabrielShapir0 sur Twitter », 12 août 2018. https://twitter.com/lex_node/status/1028727472311873537.

If the smart contract can be unilaterally halted, then you are essentially back to relying on traditional methods of deal certainty--a legalese agreement + the ability to enforce it through ADR and/or courts. That system is not a "bad thing" but falls far short of the cypherpunk dream that people thought smart contracts would deliver in the early-ish days of #Ethereum.

⁵⁹⁸ V., *infra*, § 25, §26 et §27

⁵⁹⁹ Raskin, Max. « The Law and Legality of Smart contracts ». SSRN Scholarly Paper. Rochester, NY, 22 septembre 2016. <https://doi.org/10.2139/ssrn.2842258>.

There are, however, benefits of smart contracts that do not upend the existing social order, but instead decrease transaction costs by cutting out intermediaries. This allows for industrial society to operate more effectively. These benefits extend to financial transactions, corporate governance, financial products, and a host of other potential applications that have been analyzed by economists.

Ce document devra être sous forme électronique afin qu'un *hash* de celui-ci puisse être produit⁶⁰⁰.

- **Signature numérique**⁶⁰¹. Le document électronique devra pouvoir être signé numériquement. Il peut s'agir d'une signature électronique simple⁶⁰², avancée⁶⁰³ ou qualifiée⁶⁰⁴. Dans le cas où l'une des deux dernières serait utilisée, les parties récupérerait le *hash* du fichier signé auprès du tiers de confiance par lequel elles

⁶⁰⁰ Ian Grigg. « The Ricardian Contract », 1996. https://iang.org/papers/ricardian_contract.html.

A Ricardian Contract can be defined as a single document that is (...) c) easily readable by people (like a contract on paper)...

⁶⁰¹ Ian Grigg. « The Ricardian Contract », 1996. https://iang.org/papers/ricardian_contract.html.

A Ricardian Contract can be defined as a single document that is (...) e) digitally signed.

⁶⁰² https://fr.wikipedia.org/wiki/Signature_%C3%A9lectronique_manuscrite

La signature électronique manuscrite est une signature faite à la main sur un appareil électronique qui capte la signature ainsi que ses caractéristiques comme le temps d'exécution et la pression propres à chaque segment de la signature.

⁶⁰³ Règlement (UE) n° 910/2014 du Parlement européen et du Conseil du 23 juillet 2014 sur l'identification électronique et les services de confiance pour les transactions électroniques au sein du marché intérieur et abrogeant la directive 1999/93/CE, 257 OJ L § (2014). <http://data.europa.eu/eli/reg/2014/910/oj/fra>. Article 26.

Une signature électronique avancée satisfait aux exigences suivantes: a) être liée au signataire de manière univoque; b) permettre d'identifier le signataire; c) avoir été créée à l'aide de données de création de signature électronique que le signataire peut, avec un niveau de confiance élevé, utiliser sous son contrôle exclusif; et d) être liée aux données associées à cette signature de telle sorte que toute modification ultérieure des données soit détectable.

⁶⁰⁴ Règlement (UE) n° 910/2014 du Parlement européen et du Conseil du 23 juillet 2014 sur l'identification électronique et les services de confiance pour les transactions électroniques au sein du marché intérieur et abrogeant la directive 1999/93/CE, 257 OJ L § (2014). Article 3. 11

«signature électronique qualifiée», une signature électronique avancée qui est créée à l'aide d'un dispositif de création de signature électronique qualifié, et qui repose sur un certificat qualifié de signature électronique;

seront passées⁶⁰⁵. Si une signature simple est employée⁶⁰⁶, elles devront *hasher* par elles-mêmes le document contractuel, de préférence avec un algorithme SHA-256⁶⁰⁷.

- **Identification/liaison par *hash***⁶⁰⁸. Propriété la plus emblématique du contrat ricardien, le *hash* du document ricardien devra être ensuite relié au programme exécutant l'accord des parties. Elles l'enregistreront par le biais d'une fonction spéciale dans le *smart contract* afin que ce dernier ait toujours trace de la version de l'accord contractuel qu'il implémente.

218. Conclusion Section I. Le contrat ricardien est originellement une forme de contrat inventée par Ian Grigg, qui a la propriété de rendre une convention aussi lisible par un programme qu'un humain ; et de créer la liaison entre ce document écrit en langage naturel et un programme par un jeu de *hash*. Aujourd'hui, l'appellation est utilisée pour désigner les contrats intelligents qui utilisent cette liaison entre le *smart contract* et l'accord écrit. Il est proposé que la forme devant être adoptée par les contrats intelligents des parties soit celle où l'accord écrit a toute primauté sur le *smart contract* l'exécutant et que les deux soient liés par un jeu de *hash*.

⁶⁰⁵ Agence nationale de la sécurité des systèmes d'information. « Guide de sélection du niveau des signatures et des cachets électroniques ».

Lorsque la signature électronique s'appuie sur les mécanismes de cryptographie asymétrique présentés précédemment, la donnée à signer est hachée afin d'obtenir son empreinte numérique. Ainsi dans le cas de document volumineux, il est conseillé de hacher préalablement le document afin d'en signer seulement l'empreinte afin de réduire la puissance de calcul nécessaire. Cette empreinte est alors signée avec la clé privée du signataire. Cette empreinte numérique signée correspond à la signature électronique du signataire. La donnée « signée » est donc la combinaison de la donnée initiale et de cette signature électronique sur l'empreinte.

⁶⁰⁶ Ibid.

La signature électronique « simple » correspond au niveau le plus couramment utilisé en raison de son accessibilité et de sa simplicité. En effet, il peut s'agir du niveau s'approchant le plus de l'idée que l'on peut se faire d'une signature sous format électronique à destination du grand public. Il peut s'agir par exemple d'une signature réalisée sur tablette avec un stylet.

⁶⁰⁷ Délibération SAN-2020-003 du 28 juillet 2020.

La formation restreinte relève que l'algorithme SHA-256 est une fonction de hachage permettant d'assurer l'intégrité des données personnelles traitées par la société. S'il s'agit, à ce jour, d'une fonction qui ne peut être inversée et est donc considérée par l'Agence nationale de sécurité des systèmes d'information (ANSSI) et la CNIL comme garantissant un niveau de sécurité suffisant des données, celle-ci ne permet pas d'anonymiser des données et donc de justifier leur conservation de manière indéfinie par un responsable de traitement.

⁶⁰⁸ Ian Grigg. « The Ricardian Contract », 1996. https://iang.org/papers/ricardian_contract.html.

A Ricardian Contract can be defined as a single document that is (...) g) allied with a unique and secure identifier.

Section II – Le corpus ricardien du contrat intelligent

219. **Composition et rédaction du corpus contractuel.** Dans notre modèle ricardien, *l'instrumentum* du contrat intelligent sera composé de plusieurs éléments (§1) qui pourront être dressés selon des modalités différentes (§2).

§ I – Composition du corpus

220. **Éléments du corpus contractuel.** Les documents aux propriétés ricardiennes qui composeront le corpus du contrat intelligent seront le contrat *fiat* (A) et la documentation technique relative aux smart contracts exécutant le contrat (B).

A – Le contrat *fiat*

Contenu du contrat *fiat*. Nous désignons par contrat *fiat*, le document classique, écrit en langage naturel, destiné à cadrer et sécuriser la relation contractuelle par des stipulations rédigées par un juriste⁶⁰⁹. Par exemple, si les parties créent un contrat de vente intelligent, le contrat *fiat* correspondra au document intitulé « contrat de vente » dans lequel elles décriront quels sont les actifs vendus, à quel prix, quelles sont les conditions de rétractation, etc. Il est nécessaire que ce document figure au plus haut de la hiérarchie du corpus contractuel puisqu'il formalise le *negotium* ; son contenu prime donc sur tous les autres éléments du corpus en cas de contradiction avec l'un d'entre eux, en particulier le code du *smart contract*⁶¹⁰. Ce document contiendra aussi des clauses spécifiques au recours aux smart contracts⁶¹¹. Ce sera ainsi par son biais que des effets juridiques seront donnés aux

⁶⁰⁹ Mekki Mustapha. « Le smart contract, objet du droit (Partie 2) », Dalloz IP/IT, 2019, 27.

Contrairement aux smart contractualistes, nous sommes d'avis qu'il convient de « contractualiser » le smart contract, c'est-à-dire de l'enrichir de stipulations figurant au sein d'un contrat fiat.

⁶¹⁰ Blycha, Natasha, et Ariane Garside. « Smart Legal Contracts - A Model for the Integration of Machine Capabilities into Contracts ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 7 décembre 2020. <https://papers.ssrn.com/abstract=3743932>.

(...) for one particular obligation the contracting parties may agree that the traditional natural language term contains the primary legal obligation. In this case, when the corresponding Incorporated Code fails, it is not to be considered a breach of contract as the contracting parties explicitly agree that in the case that the code fails, the contracting parties should use alternative methods to ensure the legal obligation held in the natural language term is fulfilled.

⁶¹¹ Mekki, Mustapha. « Le smart contract, objet du droit (Partie 2) », Dalloz IP/IT, 2019, p. 27.

Penser la régulation juridique des smart contracts, c'est repenser les règles qui leur sont propres. Cette régulation interne peut consister, d'une part, à contractualiser le smart contract, au moyen de clauses contractuelles adaptées.

opérations du *smart contract*⁶¹². Les parties pourront choisir de diviser ce document entre un texte cadre dans lequel elles décriront seulement le rapport juridique et un texte plus spécifique où elles détailleront comment le contrat sera exécuté à travers la *blockchain*⁶¹³.

B – Les spécifications fonctionnelles et le code source

221. Définition des spécifications fonctionnelles. Lors de la commande d'un logiciel auprès d'un prestataire, il est systématique que le client lui soumette un cahier des charges. Ce document relate en détail toutes les fonctionnalités du logiciel souhaitées par le client : il est l'expression de son besoin⁶¹⁴. Lorsque le prestataire est en possession de ce cahier des charges, il peut formuler sur celui-ci des retours ; le document final qui en résultera est appelé spécifications fonctionnelles⁶¹⁵. Ce document fixe alors la mission du prestataire et pourra lui être opposé en cas de mauvaise exécution alléguée du contrat de développement logiciel⁶¹⁶.

⁶¹² Allen, Jason, et Peter Hunn, éd. *Smart Legal Contracts: Computable Law in Theory and Practice* p. 66; Oxford, New York: Oxford University Press, 2022.

It is essential that there is sound and well-understood legal foundations to be built upon. (...). The most obvious, perhaps, would be a "wrapper" for the smart contract that specified, in conventional terms, what the code-based parts of the smart contract were intended to do (including their legal effect...).

⁶¹³ V., *supra*, §264

⁶¹⁴ Vivant, Michel. *Lamy droit du numérique - Partie 6 Guide -Titre 1 Quelle protection pour les logiciels, matériels informatiques et autres créations ? - Chapitre 1 Le droit d'auteur : mode de protection principal du logiciel - Section 1 Quel est l'objet de la protection ?* – 2696 - Editions Lamy, 2012. <https://hal-sciencespo.archives-ouvertes.fr/hal-03397686>.

Le cahier des charges est le document exprimant les besoins en informatique du client. Il servira de référence technique à l'élaboration d'un contrat.

⁶¹⁵ Vivant, Michel. *Lamy droit du numérique - Partie 3 Numérique et contrats - Division 2 Le régime général des contrats du numérique de droit privé - Chapitre 2 La négociation et la conclusion des contrats du numérique - Section 1 La place du contrat dans le déroulement d'un projet de systèmes d'information* - Editions Lamy, 2012. <https://hal-sciencespo.archives-ouvertes.fr/hal-03397686>.

Client et prestataire collaborent dès le début du projet à un travail de spécifications fonctionnelles afin de définir de manière évolutive le besoin réel du client et d'adapter le projet en fonction des changements de ce besoin.

⁶¹⁶ Ibid - *Partie 3 Numérique et contrats - Division 3 Les principaux contrats du numérique et leurs spécificités - Chapitre 3 Le contrat de développement de logiciel spécifique - Section 2 Principales obligations des parties* § 1. Obligations du prestataire – 1650 D'autres obligations contractuelles possibles - Editions Lamy, 2012. <https://hal-sciencespo.archives-ouvertes.fr/hal-03397686>.

Et toujours dans le même souci d'assurer un certain niveau de pérennité, le prestataire peut s'engager à assurer pendant un certain délai dans le cadre du contrat la correction des anomalies que pourrait rencontrer le client lors de l'utilisation du logiciel. (...) « X, au titre de la conception, de la réalisation et de la mise en fonctionnement du Logiciel s'engage, par une obligation de résultat, à garantir la bonne fourniture et la conformité du Logiciel aux spécifications fonctionnelles et techniques déterminées par les Spécifications, » ...

222. Rôle des spécifications fonctionnelles dans un contrat intelligent. Nonobstant la personne qui réalisera le *smart contract*⁶¹⁷, nous estimons que ces spécifications doivent aussi faire partie du corpus contractuel du contrat intelligent. Leur rôle ne sera alors pas, uniquement, de pouvoir éventuellement engager la responsabilité du concepteur des smart contracts mais surtout de servir de référentiel technique sur le comportement attendu de ces derniers. Ils serviront ainsi à connaître le détail de son fonctionnement, et pourront servir à le suspendre, le faire évoluer ou le détruire s'il n'agit pas conformément à ce qui était prévu de lui dans ce document⁶¹⁸.

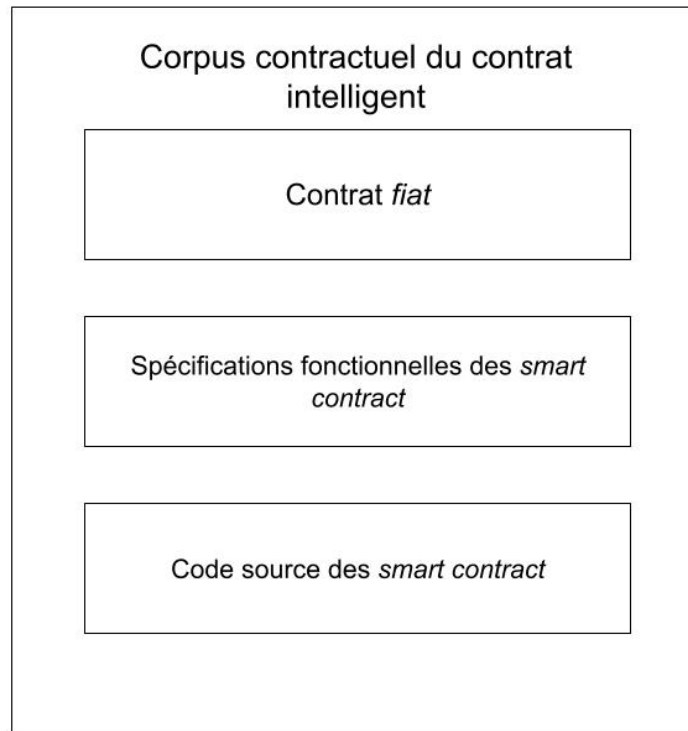
223. Le code source. Tout en bas de la hiérarchie contractuelle de notre corpus se trouvera enfin le code source⁶¹⁹ du *smart contract*. Etant donné qu'il est écrit dans un langage intelligible pour des programmeurs et qu'il peut être accompagné d'annotations et de commentaires, il est un élément pertinent pour renseigner sur le comportement attendu du programme. Mais il sera inférieur aux spécifications fonctionnelles censées décrire dans des termes claires comment le *smart contract* doit agir et au contrat *fiat* qui renseigne sur l'opération juridique souhaitée mise en œuvre par les parties.

⁶¹⁷ V., *supra*, §228

⁶¹⁸ V., *supra*, §445

⁶¹⁹ Arnould Van Eeckhout. « La maîtrise des codes sources par le client utilisateur d'un logiciel ». *Revue des contrats*, n° 4 (1 octobre 2007): 1335.

Les « codes sources » correspondent pour leur part à la version du logiciel écrite dans un langage de programmation compréhensible par l'homme. Ils sont indispensables pour comprendre le fonctionnement du programme et donc pour le modifier.



§ II – Les modalités de production du corpus

224. **Modalités d'élaboration du corpus ricardien.** Le corpus contractuel sera élaboré selon un certain ordre (A) et à la charge de l'une ou de l'ensemble des parties (B).

A – Organisation de la production

225. **La production du contrat *fiat* en premier.** Comme déjà évoqué, les parties auront, préalablement à toute chose, cerné le périmètre de ce qu'elles souhaitent exécuter dans la *blockchain*⁶²⁰. Ce n'est qu'une fois cette étape réalisée qu'elles pourront commencer à produire les éléments du corpus. Le contrat *fiat* prime sur toutes les composantes du corpus contractuel. Il est même possible qu'il fixe la hiérarchie entre elles et organise le contrat intelligent dans son ensemble⁶²¹ ; à ce titre, il doit être le premier élément du corpus à être produit.

⁶²⁰ V., *infra*, §34

⁶²¹ Pierre MOUSSERON, Jacques RAYNARD, et Jean-Baptiste SEUBE. Technique contractuelle. 5^e éd. Francis Lefebvre, 2017

Quelles que soient la qualité, l'ampleur et la précision de la rédaction, un contrat posera, quasi inéluctablement, des

226. Développement du *smart contract*. Une fois rédigé le contrat *fiat*, les parties peuvent concevoir le programme. Cela passe par la rédaction d'un cahier des charges qui aboutira aux spécifications fonctionnelles⁶²². Lorsque le programme sera codé, il sera déployé dans la *blockchain* sans toutefois être démarré ou exécuté. En effet, avant que l'ensemble de la documentation contractuelle écrite soit complète et paraphée, elle doit contenir toutes les informations sur les smart contracts qui les mettent en œuvre, y compris leur adresse de déploiement. Or il est nécessaire que ces smart contracts soient déployés pour que cette information soit récupérée. Cela signifie que les smart contracts doivent être déployés avant que le contrat *fiat* soit signé.

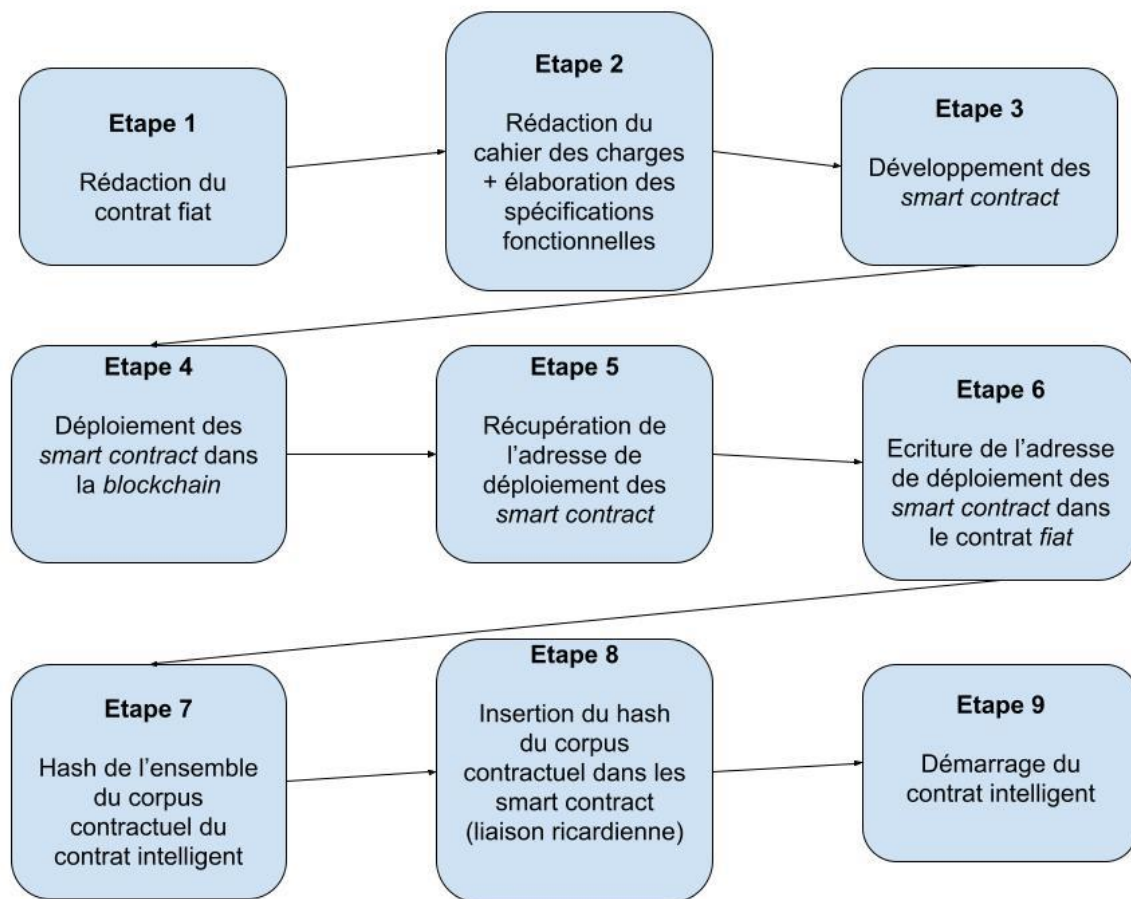
227. Récupération et enregistrement du *hash*. Lorsque l'adresse de déploiement des smart contracts sera obtenue, elle pourra être inscrite dans la documentation écrite, qui sera alors complète et prête pour la signature. Dès ce moment-là, le contrat entier pourra être *hashé* et ce *hash* sera ensuite enregistré dans les smart contracts, conformément aux propriétés du contrat ricardien. Ce n'est qu'à cet instant que le programme pourra être démarré car il contiendra bien la preuve de la documentation écrite qu'il exécute. Chaque fois que cette dernière sera modifiée, par voie d'avenant ou autre, un nouveau *hash* sera produit et devra être renseigné aux smart contracts l'exécutant⁶²³.

problèmes d'interprétation que ses défauts majoreront (...). Il s'agit, ailleurs, d'établir une hiérarchie d'accords entre les différentes pièces d'un montage contractuel et de stipuler, par exemple : - « Les dispositions du présent protocole d'accord s'imposeront en cas de contradiction ou d'ambiguïté des contrats conclus pour son application ». - « Les stipulations de ce contrat de base commanderont l'interprétation des clauses des différents accords satellites qui seront convenus à sa suite ».

⁶²² V., *infra*, §221

⁶²³ Lord Chancellor and Secretary of State for Justice. « Smart legal contracts - Advice to Government », novembre 2021.

(...) the smart (legal) contract is stored off-chain, with only the hash being recorded on-chain. This ensures both immutability and secrecy of data, but could lead to difficulties in recovering the original smart (legal) contract if it is modified.



B – Charge de la production

228. Production conjointe du corpus contractuel. Idéalement, tous les éléments du corpus contractuel auront été produits conjointement par les parties. C'est-à-dire qu'elles auront déterminé ensemble le domaine matériel du contrat intelligent ; puis elles auront négocié, rédigé et conçu de concert leur contrat intelligent, sur un pied d'égalité. De cette sorte, les obligations stipulées dans l'accord juridique seront insusceptibles d'être déséquilibrées⁶²⁴ et la responsabilité du fonctionnement du *smart contract* sera partagée.

229. Production unilatérale du corpus contractuel. Seulement en pratique, il est probable que plusieurs, voir tous les éléments du corpus contractuel soient produits que par une seule

⁶²⁴ Dimitri Houtcieff. « La réactivité en droit contemporain des contrats : des réactions unilatérales au smart contract ». Gazette du Palais, Hors série n°3 (19 juin 2019): 9.

Blockchains et autres smart contracts s'inscrivent généralement dans un contexte d'unilatéralisme. Un seul contractant est souvent détenteur de la technologie dont l'autre n'a pas même la maîtrise intellectuelle (...). L'unilatéralité se greffe à l'unilatéralisme : le smart contract éclôt du contrat d'adhésion. Le déséquilibre caractéristique du contrat d'adhésion est renforcé par les nouvelles technologies, qu'une partie dominante est en position d'utiliser au mieux de ses intérêts.

personne⁶²⁵. Dans le cadre d'une prestation de service *blockchain*, le prestataire aura sans doute rédigé lui-même le contrat *fiat*, les spécifications fonctionnelles et le code source du programme pour le proposer comme tel au client. A ce titre, il devra être vigilant à respecter des obligations spécifiques pouvant peser sur les contrats d'adhésion et/ou de consommation⁶²⁶. Il est pensé notamment aux clauses abusives⁶²⁷ de modification unilatérale du contrat et exclusive de responsabilité : il se pourrait que le prestataire soit tenté de stipuler que le contrat intelligent puisse être modifié/mis à jour unilatéralement⁶²⁸ ou qu'il se décharge de toute responsabilité en cas de dommage causé⁶²⁹ par le *smart contract*. La partie seule rédactrice devra également veiller à bien informer le consommateur du contenu du corpus contractuel et en particulier du rôle du *smart contract* dans l'exécution de la convention⁶³⁰.

230. Conclusion de la section II et du chapitre II.

Conclusion de la section II. Le contrat intelligent des parties sera composé d'un document *fiat* décrivant classiquement leur rapport contractuel. Auquel sera ajouté les spécifications

⁶²⁵ Guerlin, Gaetan. « Considérations sur les smart contracts », Dalloz IP/IT, 2017, 512.

Il est préoccupant de constater que, pour l'heure, ces sanctions (du smart contract) semblent unilatérales : elles sont essentiellement conçues et programmées en faveur d'une partie forte, à l'encontre des parties acceptantes. Le plus souvent, c'est ainsi une banque créancière ou une société de location qui bénéficie de l'usage du smart contract, par la sanction infligée au débiteur défaillant.

⁶²⁶ Garance, Cattalano. « Smart contracts et droit des contrats ». AJ Contrats d'affaires - Concurrence - Distribution, 1 juillet 2019, 321.

Finalement, le domaine rêvé du smart contract est celui des contrats de masse aux prestations simples. C'est pour eux que le smart contract sera le mieux taillé et l'automatisation des prestations la plus utile. Mais (...) à condition que ce recours à des smart contracts ne crée pas de déséquilibre significatif à leur détriment puisque le contrat sera probablement un contrat d'adhésion, voire de consommation.

⁶²⁷ Dimitri Houtcieff. « La réactivité en droit contemporain des contrats : des réactions unilatérales au smart contract ». Gazette du Palais, Hors série n°3 (19 juin 2019): 9.

D'autres observeront que les smart contracts portent en eux le risque d'une forme renouvelée de l'adhésion, où la puissance technologique d'une partie lui permet, non plus seulement d'imposer des clauses abusives, mais aussi de se ménager des processus d'exécution automatisés et se dispensant du juge.

⁶²⁸ Article R212-1 du code de la consommation : Dans les contrats conclus entre des professionnels et des consommateurs, sont de manière irréfragable présumées abusives, au sens des dispositions des premier et quatrième alinéas de l'article L. 212-1 et dès lors interdites, les clauses ayant pour objet ou pour effet de : (...) 3° Réserver au professionnel le droit de modifier unilatéralement les clauses du contrat relatives à sa durée, aux caractéristiques ou au prix du bien à livrer ou du service à rendre ;

⁶²⁹ Ibid : (...) 6° Supprimer ou réduire le droit à réparation du préjudice subi par le non-professionnel ou le consommateur en cas de manquement par le professionnel à l'une quelconque de ses obligations ;

⁶³⁰ Article L211-1 du code de la consommation : Les clauses des contrats proposés par les professionnels aux consommateurs doivent être présentées et rédigées de façon claire et compréhensible.

fonctionnelles des smart contracts, qui décriront précisément le comportement voulu des programmes qui exécuteront les stipulations écrites. Et enfin figurera le code source des smart contracts, qui peuvent également renseigner sur le fonctionnement attendu de ces derniers. Le document fiat sera celui placé au sommet de la hiérarchie d'interprétation du corpus, avant les spécifications fonctionnelles et le code source du *smart contract*.

Conclusion du chapitre II. Les parties seront donc avisées de conférer la forme ricardienne à leur contrat : c'est-à-dire produire un texte écrit en langage naturel dans lequel elles décriront leur rapports et le comportement attendu du *smart contract* ; ce dernier étant soumis qu'à un rôle de simple outil performatif de la convention. Cette forme a l'avantage d'être la plus sécurisante des formes de contrats intelligents et s'implémente de manière élégante avec le *smart contract* : dans le contrat ricardien, le *hash* de la documentation écrite en langage naturel est en permanence enregistré dans le programme qu'il exécute. Le corpus contractuel de la forme ricardienne sera alors formé d'un contrat *fiat* et la documentation technique relative au *smart contract*. Cet ensemble sera produit, par l'une ou l'ensemble des parties en commençant par le document *fiat* contenant les stipulations écrites en langage naturel du contrat.

231. Conclusion de la partie. Ainsi s'achève la phase de délimitation du contrat intelligent des parties. Au cours de cette dernière, les parties auront cerné exactement la mesure dans laquelle elles souhaitent exécuter leur contrat dans la *blockchain* : c'est-à-dire identifier les différents processus qui sont les plus opportuns à exécuter par *smart contract*, qui se retrouvent dans un nombre assez élevé de contrats. Une fois cette étape réalisée, elles auront déterminé la forme de leur contrat et son architecture générale dans laquelle elles auront donné à la documentation écrite en langage naturel toute primauté. Les parties seront alors prêtes pour élaborer effectivement leur contrat intelligent.

PARTIE II – ÉLABORATION DU CONTRAT INTELLIGENT

232. Elaboration du contrat intelligent. Après avoir délimité les domaines de leur contrat intelligent, les parties peuvent commencer à le confectionner ; c'est-à-dire s'atteler concrètement à sa réalisation. A cette étape, elles ont une idée précise de la mesure dans laquelle elles souhaitent exécuter leur contrat dans la *blockchain*, ainsi que de l'architecture qu'elles veulent lui donner. Le travail à effectuer peut alors être vaste en fonction de leur projet, même si il est possible de l'essentialiser en deux grandes tâches. Conformément à notre préconisation de forme de contrat intelligent, nous estimons que la première tâche principale sera celle de la rédaction des clauses du contrat *fiat* (Titre I). Nous avons vu, en effet, qu'un contrat intelligent sécurisé ne peut faire l'économie d'une formalisation écrite en langage naturel des intentions des parties. Dans celle-ci, les parties y feront figurer un certain nombre de stipulations qui intéresseront spécialement le recours à la *blockchain* et aux smart contracts pour l'exécution de leur contrat.

Ensuite, les parties pourront passer à l'étape du développement technique du contrat intelligent (Titre II), et en particulier la réalisation des smart contracts chargés d'exécuter les stipulations de l'accord écrit. Cette phase d'implémentation peut abriter beaucoup d'autres tâches⁶³¹, mais nous avons fait le choix de nous concentrer uniquement sur celles relatives à la *blockchain* car notre méthodologie est centrée sur cet aspect. Cela signifie que nous ne discuterons pas d'autres aspects techniques que ceux intéressant la *blockchain* et les smart contracts. Alors qu'un contrat intelligent peut fonctionner à l'aide de nombreux éléments technologiques : des objets connectés⁶³², des applications web ou mobiles⁶³³, de l'intelligence artificielle⁶³⁴ ; nous nous focaliserons que sur le choix des protocoles où les parties pourront faire fonctionner leur smart contracts, ainsi que du codage de ces derniers.

⁶³¹ Dans un article Elise GUILHAUDIS donne l'exemple d'un contrat de covoiturage exécuté dans la *blockchain*. Outre les aspects purement *smart contract*, l'application fait intervenir bien d'autres éléments techniques.

Elise GUILHAUDIS. « Comprendre la blockchain à travers l'étude d'un cas pratique : Le covoiturage "BlockCar" », Lamy Droit de l'immatériel, n° 143 (1 décembre 2017).

⁶³² Kumar, S., A. Murugan, B. Muruganantham, et B. Sriman. « IoT-smart contracts in data trusted exchange supplied chain based on block chain ». International Journal of Electrical and Computer Engineering (IJECE) 10 (1 février 2020): 438. <https://doi.org/10.11591/ijece.v10i1.pp438-446>.

⁶³³ Dai, Patrick, Neil Mahi, Jordan Earls, et Alex Norta. « Smart-Contract Value-Transfer Protocols on a Distributed Mobile Application Platform », 3 mars 2017. <https://doi.org/10.13140/RG.2.2.35140.63365>.

⁶³⁴ Reshi, Iraq, Muneeb Khan, Sadaf Shafi, Sahil Sholla, Assif Assad, et Huzaib Shafi. « AI-Powered Smart contracts: The Dawn of Web 4. O », 6 mars 2023. <https://doi.org/10.36227/techrxiv.22189438.v1>.

233. **Le postulat de la proposition de rédaction de clauses.** La première étape de la confection d'un contrat intelligent est donc la rédaction de l'accord juridique en langage naturel, qui est au cœur de notre proposition de méthodologie. Nous partirons du postulat que les parties seront placées sur un relatif pied d'égalité. Nos recommandations d'écriture de clauses seront donc au bénéfice de tous les co-contractants et viseront dans la mesure du possible à instituer un équilibre contractuel dans la relation plutôt qu'à avantager l'une ou l'autre des parties⁶³⁵.

Le contenu de l'accord écrit dépendra évidemment de la relation encadrée. Toutefois il demeure un bon nombre de clauses qui se retrouveront dans le corps de toute convention exécutée dans une *blockchain*⁶³⁶ : ce sont sur celles-ci que nous suggérons aux parties. Par souci de simplification, nous les aborderons dans l'ordre selon lequel elles sont le plus susceptibles d'apparaître au sein d'un contrat. Ainsi, nous distinguerons les clauses se situant typiquement en tête de contrat (Chapitre I), celles se situant en son cœur (Chapitre II) et celles figurant à la fin (Chapitre III).

⁶³⁵ Nous souhaitons éviter la situation où une partie forte confectionnerait qu'à son avantage le contrat.

Mekki, Mustapha. « Le contrat, objet des smart contracts (Partie 1) ». Dalloz IP/IT, 2018, 409.

La morale des algorithmes est la morale des forts.

⁶³⁶ Mekki, Mustapha. « Les mystères de la blockchain ». Recueil Dalloz, n° 37 (2 novembre 2017): 2160.

(...) un contrat « fiat » va devoir accompagner la mise en oeuvre des smart contracts. Il faut revenir au contrat comme instrument de gestion des risques.

Chapitre I - Stipulations initiales du contrat

234. **Clauses situées au « Haut du contrat ».** Il n'existe pas de normes prévoyant ce qui doit figurer dès les premiers paragraphes d'un contrat, mais il est très courant que ceux-ci soient consacrés à l'introduction des protagonistes, des termes (Section I) et de l'opération encadrée par la convention (Section II)⁶³⁷. Ces clauses appellent un traitement particulier dans notre contexte car elles introduiront la *blockchain* et les smart contracts aux lecteurs du contrat.

Section I - Définitions et identités

235. **Présentations des parties et des concepts.** En sus de leurs identités réelles, les parties devront préciser comment elles seront identifiées dans le protocole abritant leur smart contracts (§1) ; tout comme elles seront contraintes d'établir plusieurs éléments de la terminologie de la *blockchain* dans la clause dédiée à cet effet (§2).

§ I – Clause relative à l'identité des parties

236. **Identification des parties.** Les parties devront être clairement identifiées dans un contrat intelligent afin d'assurer sa légalité et sa sécurité (A) ; et la clause relative à leur identité devra aussi contenir leur adresse cryptographique qui les identifie dans la *blockchain* (B).

⁶³⁷ Dissaux Nicolas, Chantepie Gaël, Auque Françoise, Deroussin David, Guerlin Gaëtan juriste, Houtcieff Dimitri, Joyeux Arthur, et al. La stylistique contractuelle. Thèmes et commentaires Études. Paris: Dalloz, 2022. p. 127

Tous les actes obéissent à des standards de présentation, comme c'est le cas, notamment, de ce que Bertrand Fages nomme « le haut du contrat » (...). En haut, tout en haut, figure d'abord l'intitulé (...). Puis vient la désignation des parties avec leur identité et leurs éventuels pouvoirs (...) Cette présentation (...) se poursuit par un sous-ensemble emprunté à l'élaboration de la loi, le préambule ou exposé des motifs...

A – L'identité réelle des parties

237. Intérêts de la déclamation de l'identité. Après l'intitulé du contrat, il est systématique de trouver dans les stipulations qui suivent des informations sur l'identité des parties. Cet exercice a de nombreuses utilités⁶³⁸ :

- en exposant des informations sur son identité, une partie renseigne sur sa capacité juridique à contracter, ce qui est une condition de la formation d'un contrat⁶³⁹,
- elle renseigne aussi son co-contractant sur la personne avec qui elle conclue, ce qui éclaire son consentement et qui est autre condition de la formation d'un contrat⁶⁴⁰,
- enfin plus pratiquement, la description de l'identité donne aussi des renseignements sur le domicile et les appellations de la partie⁶⁴¹ ; cela fournit au co-contractant des informations à l'aide desquelles il pourra éventuellement porter des réclamations en cas de litige⁶⁴².

238. Contracter sous pseudonyme. Ce dernier élément est fondamental dans notre contexte d'exécution sur *blockchain*. En effet, nous avons déjà expliqué en quoi, dans notre proposition de méthodologie, les parties comptent autant sur la force légale que sur les smart contracts

⁶³⁸ Pierre MOUSSERON, Jacques RAYNARD, et Jean-Baptiste SEUBE. Technique contractuelle - Chapitre premier Les partenaires - Section 3 Les parties - I L'identification des parties - §226 . 5^e éd. Francis Lefebvre, 2017

Cette identification est importante, car c'est au niveau des parties ainsi identifiées que devront être appréciées un certain nombre de conditions du contrat (consentement, capacité...) et que se produiront ses effets.

⁶³⁹ Article 1128 du code civil : Sont nécessaires à la validité d'un contrat : (...) 2° Leur capacité de contracter ...

⁶⁴⁰ Article 1128 du code civil : Sont nécessaires à la validité d'un contrat : 1° Le consentement des parties ...

⁶⁴¹ Pierre Mousseron, Jacques Raynard, et Jean-Baptiste Seube. Technique contractuelle - Chapitre premier Les partenaires - Section 3 Les parties - I L'identification des parties - §226 . 5^e éd. Francis Lefebvre, 2017

L'identification des parties est assurée par quelques informations d'état civil : nom du partenaire personne physique, raison ou dénomination sociale du partenaire personne morale, domicile ou siège social ...

⁶⁴² Pour une assignation ou une requête, l'article 54 du code de procédure civile dispose que : *La demande initiale est formée par assignation ou par requête remise ou adressée au greffe de la juridiction (...) La demande initiale mentionne : (...) 3° a) Pour les personnes physiques, les nom, prénoms, profession, domicile, nationalité, date et lieu de naissance de chacun des demandeurs ; b) Pour les personnes morales, leur forme, leur dénomination, leur siège social et l'organe qui les représente légalement ;*

pour garantir l'exécution de leurs contrats⁶⁴³. Or, si une partie ne renseigne pas son identité, et se contente, par exemple, que de donner son adresse public pour s'identifier, alors *de facto*, son co-contractant ne peut compter que sur le bon fonctionnement du *smart contract* pour être certain que les engagements formulés à son égard seront bien exécutés⁶⁴⁴. Puisqu'en effet ce dernier ne pourra pas, sans le nom réel et l'adresse physique de son co-contractant, l'astreindre devant le juge pour le forcer à exécuter les obligations du contrat⁶⁴⁵. Ainsi, nonobstant la faisabilité de contracter dans la *blockchain* pseudonymement⁶⁴⁶, notre proposition de forme de contrat intelligent est incompatible avec cette option. Il sera nécessaire que les parties déclament exhaustivement leur identité afin de garantir légalement le respect des engagements.

B – L'adresse cryptographique

239. Intérêt de renseigner l'adresse cryptographique. Les parties devront également renseigner l'adresse cryptographique, c'est-à-dire celle de leur *wallet*, qui constitue leur identité dans la *blockchain*⁶⁴⁷. Il n'y a, en principe, aucun moyen de savoir quelle est l'identité réelle de la personne derrière une adresse cryptographique⁶⁴⁸. Il est donc nécessaire que cette liaison, *wallet* – personne,

⁶⁴³ V., *infra*, §215

⁶⁴⁴ Raskin Max. « The Law and Legality of Smart contracts ». SSRN Scholarly Paper. Rochester, NY, 22 septembre 2016. <https://doi.org/10.2139/ssrn.2842258>.

Among the most radical visions for smart contracts is that the technology will subject the provision of justice to market forces and break the state's monopoly over the court system.

⁶⁴⁵ Lord Chancellor and Secretary of State for Justice. « Smart legal contract, Advice To Government », §3. 21, novembre 2021.

However, an agreement reached between parties unknown to one another may give rise to difficulties in practice. Herbert Smith Freehills made the point that, practically speaking, it may be very difficult for a party to seek and enforce a remedy against a counterparty whose identity is unknown.

⁶⁴⁶ Eva Théochardi. « La conclusion des smart contracts : révolution ou simple adaptation ? », Revue Lamy Droit civil, n° 161 (1 juillet 2018).

D'un point de vue juridique, l'identification des parties ne suscite pas nécessairement des difficultés majeures pour leur consentement (...) notamment lorsque le contrat, le plus souvent d'exécution instantanée, est conclu sans considération des qualités propres de la personne du cocontractant. Ainsi, dans un contrat de vente de bitcoins, l'absence d'identification des parties importe peu pour la conclusion du contrat.

⁶⁴⁷ Il est entendu ici l'adresse de leur *wallet* : le moyen à l'aide duquel elles peuvent initier toute interaction avec la *blockchain*.

⁶⁴⁸ Eva Théochardi. « La conclusion des smart contracts : révolution ou simple adaptation ? », Revue Lamy Droit civil, n° 161 (1 juillet 2018).

Il résulte de ce processus que l'identification des parties au contrat s'établit à travers leur clé publique, qui constitue un pseudonyme. Il devient alors très difficile d'établir un lien entre le pseudonyme et la vraie identité de la partie contractante. En effet, lorsque les relations contractuelles sont créées sur des blockchainss publiques, il n'existe aucune

soit faite dans le document *fiat*. De cette sorte, les parties pourront, par exemple, prouver que l'envoi de cryptoactifs à une personne a été régulièrement accompli, ou généralement démontrer l'accomplissement ou non d'une obligation par une personne en observant son adresse cryptographique.

240. Localisation de la définition de l'adresse. Une adresse cryptographique ressemble à cela : *0x76703A497ea6c61285B43eCD89Ed97C87eD3bce1*. Elle pourra donc être renseignée dans le paragraphe sur la description de l'identité réelle des parties comme suit : *Mme A, demeurant..... et identifiée par l'adresse publique cryptographique : 0x....dans la Blockchain*.

Les parties pourront également définir le terme d'adresse cryptographique, *Adresse Blockchain*, dans la clause de définition des termes.

Adresse Blockchain : adresse cryptographique identifiant un individu ou un Smart contract dans la Blockchain. Exemple d'utilisation : Mme A demeurant.... et dont l'Adresse Blockchain est 0x...

§ II – Clause de définition des termes

241. Terminologie du contrat. Il est usuel, dans un contrat, de prévoir une clause consacrée à la définition des notions qui y seront employées⁶⁴⁹. La démarche est opportune à plusieurs égards. D'une part, car elle rend commode l'écriture du contrat : en définissant précisément et exhaustivement un terme, il peut être réutilisé dans les clauses qui suivront sans nécessiter de répéter à nouveau tout le sens qu'il recouvre. D'autre part, la définition claire d'une notion dans le contrat permet de contractualiser le sens que les parties leur donnent⁶⁵⁰. L'acceptation d'une définition donnée d'un terme peut ainsi avoir pour effet de régler, en amont, d'éventuels points de contention

gouvernance centralisée qui serait éventuellement en mesure de fournir des informations sur l'identité d'un membre du réseau. Par conséquent, les parties ignorent la véritable identité de leur cocontractant.

⁶⁴⁹ Étienne Vergès. « Contrats sur la recherche et l'innovation - Chapitre 212 - Clauses relatives aux définitions des termes du contrat - § 1 - Utilité des définitions dans le contrat », Dalloz, 2019 2018.

Une clause fréquente dans les contrats. La pratique d'origine anglo-américaine consistant à insérer des clauses de définitions dans les contrats s'est largement développée dans la pratique rédactionnelle contractuelle au niveau international.

⁶⁵⁰ Lettre des réseaux. « Clause de définition ». Consulté le 15 mai 2023. <https://www.lettredesreseaux.com/P-783-678-PI-clause-de-definition.html>.

Les clauses de définition, souvent placées en préambule de l'acte (ou dans un glossaire annexé à l'accord), permettent de contractualiser la définition que les parties entendent donner à des termes utilisés par la suite dans le contrat.

ou servir d'arguments utiles en cas de litige⁶⁵¹. Dans notre contexte spécifique, les parties devront définir des termes relatifs à la *blockchain* (A) et aux smart contracts (B).

A – Définitions relatives à la blockchain

242. Fork et blockchain. Le premier terme, particulier à notre contexte, que les parties devront définir est évidemment celui de *blockchain* : l'infrastructure où se trouvera leurs actifs, les smart contracts qu'elles auront développés ou fait développer pour exécuter leurs contrats et d'autres smart contracts avec lesquels elles pourront interagir. Les parties seront néanmoins confrontées à un obstacle de taille dans l'écriture de sa définition à cause du phénomène de *fork*.

Un *fork* correspond concrètement à une mise à jour du logiciel de la *blockchain*⁶⁵². Comme celle-ci est en principe décentralisée, l'acceptation de la mise à jour du logiciel de la *blockchain* ne se fait pas forcément sans difficultés. Il est nécessaire que les individus détenant le logiciel de la *blockchain*, les « nœuds », acceptent cette mise à jour, car il n'existe pas d'entité centralisée pouvant contraindre la désignation de la version officielle⁶⁵³. Lorsque cette acceptation s'opère sans contestation, le *fork* est dit non contentieux : les nœuds ont intégré sans discussions la version suivante du logiciel, qui a été proposée par la communauté.

Au contraire, lorsqu'une partie significative des nœuds décide de ne pas adopter une nouvelle version

⁶⁵¹ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires - Chapitre 28 Clause de définition - § I - Objet et utilité - 446 Notion. Les Intégrales. LGDJ, 2018.

Les clauses de définition ont pour objet de définir de manière circonstanciée les principaux termes du contrat, afin de limiter les risques d'ambiguïté ou d'incohérence et les discussions qui pourraient en découler. Les clauses de définition ont une fonction de prévention des difficultés d'interprétation évidente, en écartant les interprétations divergentes qui pourraient naître de la terminologie employée. Les clauses de définition sont ainsi une forme d'interprétation du contrat par les parties elles-mêmes, qui imposent une interprétation commune et univoque des termes du contrat.

⁶⁵² Artzt, Matthias, et Thomas Richter. *Handbook of Blockchain Law: A Guide to Understanding and Resolving the Legal Challenges of Blockchain Technology* - §1.09 BLOCKCHAIN FORKS. Kluwer Law International B.V., 2020.

The software which runs on the nodes of a blockchain network is updated from time to time to add new features or to make changes in how the blockchain functions. These code changes are considered adopted when a majority of nodes on the network install and start using an updated version of the blockchain software.

⁶⁵³ Laure De La RAUDIÈRE et Jean-Michel MIS. « Mission d'information commune sur les chaînes de blocs (blockchains) ». Assemblée Nationale, 12 décembre 2018.

Comment cela se déroule-t-il concrètement ? Une blockchain dont le code est modifié, par exemple pour une mise à jour bénigne donne, en réalité, naissance à une nouvelle blockchain. Celle-ci ne fonctionne que si une majorité suffisante de nœuds du réseau l'accepte et l'intègre dans son processus de validation. À la différence des logiciels classiques, dont l'éditeur propose à l'utilisateur de façon unilatérale une mise à jour, la blockchain fonctionne, sur un modèle décentralisé. Lorsque la mise à jour ne pose pas de problème, la nouvelle blockchain est acceptée à l'unanimité.

du logiciel pour diverses raisons, le *fork* est dit contentieux⁶⁵⁴. Dans ce cas-là, la *blockchain* fait l'objet d'une scission : il existe une version qui intègre la mise à jour et une autre qui ne l'intègre pas. Ainsi, les parties se retrouvent dans une situation où leurs smart contracts et leurs actifs seront dupliqués sur deux infrastructures dont la légitimité sera concurrente. Il peut alors exister une période d'incertitude, plus ou moins longue, pour déterminer quelle *blockchain* doit être considérée comme « officielle »⁶⁵⁵.

La solution la plus sécurisée pour neutraliser ce problème est de prévoir une procédure d'amendement du contrat en cas de *fork* contentieux. À l'occurrence d'un tel événement, la clause permettra à des parties de modifier et préciser dans le contrat laquelle des branches de la *blockchain* récemment scindée devra être considérée comme canonique⁶⁵⁶. Néanmoins, la modification d'un contrat étant une démarche lourde et pouvant être abusée, les parties pourront être avisées de donner une définition de la *blockchain* qui puisse gérer, au moins pour un temps, l'éventualité d'un tel *fork*.

Une manière de s'y prendre pourrait être de s'appuyer sur le choix de la majorité des nœuds. Les parties choisiraient un logiciel de *blockchain* comme référence⁶⁵⁷. En cas de *fork* contentieux, la version de ce logiciel à considérer comme officielle sera alors celle qu'une majorité de nœuds aura choisie⁶⁵⁸. Cette méthode, simple, a toutefois plusieurs failles et ne peut pas constituer un moyen

⁶⁵⁴ Heal, Jordan. « Hard forks: Contentious or not? » Coin Rivet, 16 janvier 2019. <https://coinrivet.com/hard-forks-contentious-or-not/>.

There are two types of forks: contentious and non-contentious. A contentious hard fork will typically occur when there is a disagreement within a community. The two disagreeing factions will fork the chain and implement the changes they desire on their respective chains. A non-contentious hard fork will occur when a fundamental change in the code is required to upgrade the blockchain.

⁶⁵⁵ Laure De La RAUDIÈRE et Jean-Michel MIS. « Mission d'information commune sur les chaînes de blocs (blockchains) ». Assemblée Nationale, 12 décembre 2018.

Cependant, lorsque les modifications radicales du code ne sont pas intégrées par l'ensemble du réseau, par exemple lorsqu'un ensemble de nœuds « puristes » refusent une modification du code originel malgré la présence d'une faille, deux blockchains commencent à coexister: En août 2017, les blockchains Bitcoin Cash et Bitcoin Gold sont ainsi nées de bifurcations de Bitcoin d'origine (core) : les trois cryptomonnaies se font désormais concurrence. En cela, un tel mouvement de bifurcation n'est guère différent d'un contentieux classique entre associés, sur l'avenir de leur entreprise par exemple, et qui conduit au départ de certains d'entre eux pour créer une entreprise concurrente.

⁶⁵⁶ Il s'agit plus ou moins de la méthode que retient Gabriel Shapiro, principal promoteur de la forme « code and law » Dans le contrat qu'il propose, les parties désignent la *blockchain* qu'ils considèrent comme canonique mais une clause prévoit que le contrat pourra être amendé/modifié

⁶⁵⁷ [_g4brielShapir0. « SCoDA – Simple Code Deference Agreement », 25 avril 2023. https://github.com/lex-node/SCoDA-Simple-Code-Deference-Agreement-](https://github.com/lex-node/SCoDA-Simple-Code-Deference-Agreement-)

⁶⁵⁸ Pour Ethereum, le logiciel ou client « go-ethereum » est le plus ancien et le mieux considéré comme le logiciel officiel.

⁶⁵⁸ Dans le cas du fork d'Ethereum en 2016, un nombre bien plus élevé de nœuds avait choisi la version 1.4.10 du client *go-ethereum* (implémentant le correctif annulant les effets du hack de the DAO) plutôt que celle 1.4.9 (qui laissait le client

absolument fiable de détermination de la version officielle d'une *blockchain* :

- le logiciel que la majorité de nœuds a choisi n'est pas nécessairement celui considéré comme officiel par la communauté⁶⁵⁹;
- les outils qui existent pour mesurer le nombre de nœuds sont faillibles⁶⁶⁰ ;
- enfin, il est difficile à l'occasion d'un *fork* d'établir à partir de quand la communauté se sera stabilisée pour se mettre d'accord sur la *blockchain* à considérer comme canonique. Il s'agit d'une période dont le délai est impossible à anticiper.

Pour toutes ces raisons, il est donc indispensable que les parties gardent la possibilité de modifier leur contrat *a posteriori* afin de désigner la version de la *blockchain* qu'elles considèrent comme officielle. Pour neutraliser les potentiels abus, elles pourront prévoir l'application d'une clause pénale⁶⁶¹ lorsque la détermination de la version canonique d'une *blockchain* ne posait pas sérieusement question et qu'une procédure de modification a tout de même été déclenchée. Les parties prendront donc le soin de définir un logiciel de la *blockchain*, un *fork* contentieux et ce qu'elles estiment être la version canonique du client.

Exemple de rédaction :

Client : désigne le logiciel « go-ethereum »⁶⁶² (provenant de la fondation Ethereum) utilisé pour

inchangé après le hack).

⁶⁵⁹ Comme ça a été le cas avec la blockchain Steemit dont Justin Sun avait acheté tous les nœuds mais que la communauté a tout de même décidé d'abandonner au profit du *fork* Hive.

B, Hugh. « Hive vs Steem - Le fork rebelle dépasse de maître vendu ». CryptoActu (blog), 4 juin 2020. <https://cryptoactu.com/hive-vs-steem-fork-rebelle-depasse-maitre-vendu/>.

⁶⁶⁰ Happy Sharer. « Exploring How Many Ethereum Nodes Are There: An Analysis of the Ethereum Network - The Enlightened Mindset », 18 janvier 2023. <https://www.lihpao.com/how-many-ethereum-nodes-are-there/>.

When it comes to understanding the size and scope of the Ethereum network, one of the first questions people have is "how many Ethereum nodes are there?" Unfortunately, there is no single answer to this question as the exact number of Ethereum nodes can vary depending on the data source used. To get an accurate picture of the number of Ethereum nodes, it is necessary to analyze multiple data sources.

⁶⁶¹ Article 1231-5 du code civil : Lorsque le contrat stipule que celui qui manquera de l'exécuter paiera une certaine somme à titre de dommages et intérêts, il ne peut être alloué à l'autre partie une somme plus forte ni moindre.

⁶⁶² Il s'agit du premier client/logiciel de la *blockchain* Ethereum, développé par la fondation Ethereum. Il est une bonne pratique de s'appuyer sur celui-ci car on peut estimer qu'il est ce qui se rapproche le plus de la version « officielle » et donc qui sera la plus pérenne.

interagir avec la Blockchain.

Fork contentieux : désigne un point de contention sérieux dans la communauté sur la version du Client à considérer comme canonique.

*Version canonique : désigne la version du Client que le plus grand nombre de nœuds (consultable notamment sur le site *ethernodes.com*) a choisi pendant une durée ininterrompue de 3 semaines à compter d'un Fork contentieux.*

L'instrument servant à mesurer peut ne pas être mentionné, surtout s'il ne paraît pas pérenne⁶⁶³. Le délai est important : il faut laisser une période à la communauté pour s'accorder sur la version à considérer comme officielle qui se reflétera dans le choix de la majorité des nœuds.

Procédure d'amendement en cas de Fork contentieux : Si les parties sont incapables de déterminer la Version Canonique d'après les instructions données à l'article, alors elles conviennent de suspendre le contrat (conformément à l'article Suspension) et l'amender pour désigner conjointement quelle version du Client elles choisissent comme canonique. La procédure peut être unilatéralement déclenchée par Celles-ci s'engagent à trouver un accord sur la Version Canonique dans un délai de ... Dans le cas où elles n'y seraient pas parvenues à l'issue de ce délai, elles soumettront la désignation à [nom arbitre].... Si la détermination de la Version Canonique de la Blockchain ne posait pas sérieusement question, la partie qui a déclenché abusivement la présente procédure pourra être contrainte au paiement d'une pénalité de euros.

243. Définition de la blockchain dans le contrat. La question du fork réglée, les parties pourront ensuite simplement décrire l'infrastructure d'exécution de leurs smart contracts. Il faudra mentionner son appellation courante et préciser qu'il s'agit de sa version en production, car il existe des versions tests⁶⁶⁴.

La Blockchain : le dispositif d'enregistrement électronique partagé⁶⁶⁵ communément appelé Ethereum, supporté par la Version Canonique du Client.

⁶⁶³ Il n'est pas improbable que des sites internet comme *ethernodes* n'existent plus plusieurs années après la constitution du contrat.

⁶⁶⁴ A l'heure où sont écrites ces lignes, la blockchain *Ethereum* dispose de plusieurs versions test : *Rinkeby* et *Ropsten* notamment.

⁶⁶⁵ La définition reprend le terme de dispositif d'enregistrement électronique partagé choisie par le législateur français.

Article R211-9-7 code monétaire et financier : *Le dispositif d'enregistrement électronique partagé mentionné à l'article L. 211-3 est conçu et mis en œuvre de façon à garantir l'enregistrement et l'intégrité des inscriptions et à permettre, directement ou indirectement, d'identifier les propriétaires des titres, la nature et le nombre de titres détenus.*

B – Définitions relatives aux smart contracts

244. Définition d'un *smart contract* dans le contrat. Nous avons déjà expliqué en quoi l'expression de *smart contract* pouvait être malvenue et confusante⁶⁶⁶. Malheureusement, son usage s'est durablement ancré dans le milieu de la *blockchain* et il ne nous semble pas avoir émergé un terme plus naturel et approprié que celui-ci pour désigner les programmes exécutés dans une *blockchain*⁶⁶⁷. De plus, comme déjà évoqué, la bonne définition de la notion rend inconséquente le terme par la suite utilisé⁶⁶⁸.

Les parties pourront donc être amenées à mobiliser le terme de *smart contract* pour désigner les programmes, de façon générale, fonctionnant dans la *blockchain* mais également pour désigner spécifiquement certains de ces programmes en particulier. Pour les premiers, elles pourront les définir en mettant simplement en avant leurs caractéristiques fondamentales : celles d'être des programmes informatiques qui sont déployés et fonctionnent dans une *blockchain*⁶⁶⁹.

Smart contract : désignent des programmes déployés et fonctionnant dans La Blockchain.

Exemple d'utilisation : *M. A décline toute responsabilité quant aux dommages résultant de dysfonctionnements de Smart contract autres que ceux qu'il a développés ;*

Pour les seconds, ceux que les parties auront conçus ou fait concevoir, ou encore ceux avec lesquels elles souhaiteront interagir, elles prendront le soin de préciser leurs adresses de déploiement dans la

⁶⁶⁶ V., *infra*, §3

⁶⁶⁷ « Automate exécuteur de clauses » proposée dans le vocabulaire officielle des actifs numériques nous paraît trop réducteur malgré la longueur de l'expression.

Vocabulaire des actifs numériques (liste de termes, expressions et définitions adoptés) (s. d.).

(...) 2. Un automate exécuteur de clauses peut, par exemple, déclencher l'indemnisation automatique d'un assuré dont l'avion aurait pris du retard. 3. Un automate exécuteur de clauses utilise généralement un dispositif d'enregistrement électronique partagé.

⁶⁶⁸ Lettre des réseaux. « Clause de définition ». Consulté le 15 mai 2023. <https://www.lettredesreseaux.com/P-783-678-P1-clause-de-definition.html>.

(...) la clause peut donner à un mot employé dans le contrat un sens différent de son sens habituel : l'interprète du contrat sera alors lié par ce sens dérogatoire ;

⁶⁶⁹ De Filippi, Primavera, et Aaron Wright. Blockchain and the Law: The Rule of Code. p : 538. Harvard University Press, 2018. <https://www.jstor.org/stable/j.ctv2867sp>.

Blockchains are equipped to store or reference other forms of information, including what are essentially small computer programmes -- which technologists often refer to as smart contracts.

blockchain, leurs codes sources et le cas échéant leurs spécifications fonctionnelles⁶⁷⁰. Elles pourront aussi utilement décrire le rôle dévolu à ces smart contracts.

Les smart contracts de séquestre: désignent les smart contracts destinés à exécuter le mécanisme de séquestre du présent contrat. Le code source de ces smart contracts est accessible à Il est déployé dans la Blockchain à l'Adresse Blockchain suivante : 0x..... Les parties peuvent consulter ses spécifications fonctionnelles en annexe du présent contrat.

245. Définition des jetons dans le contrat. En accord avec notre propos sur le domaine matériel du contrat intelligent, les smart contracts des parties seront presque systématiquement amenés à manipuler toutes sortes d'actifs représentés par des jetons⁶⁷¹. Les parties devront donc définir ces actifs en explicitant leur rôle. Les jetons étant issus de *smart contract*⁶⁷², elles préciseront également leur adresse de déploiement afin de référencer précisément l'actif qu'elles souhaitent manipuler.

Euro Numérique : désigne la représentation numérique, dans la Blockchain, d'un euro. Elle provient du Smart contract déployé à l'Adresse Blockchain suivante 0x...

Action Numérique : désigne la représentation numérique, dans la Blockchain, d'une action de la société A. Elle provient du Smart contract déployé à l'Adresse Blockchain suivante 0x...

246. Interaction confirmée. Beaucoup de types d'interaction sont susceptibles d'être effectués dans une *blockchain* : des transferts d'actifs, du pointage⁶⁷³, de la signature, du stockage d'informations... Quelles qu'elles soient, les parties devront être vigilantes à décrire selon quelles

⁶⁷⁰ V., *infra*, §221

⁶⁷¹ Allen, J. G. « Wrapped and Stacked: 'Smart contracts' and the Interaction of Natural and Formal Language ». *European Review of Contract Law* 14, n° 4 (19 décembre 2018): 307-43. <https://doi.org/10.1515/ercl-2018-1023>.

The connection between smart contracts and cryptocurrency is not accidental, because smart contracts lend themselves particularly well to manipulating assets such as digital tokens that take the form of immaterial object. the core use case of smart contracts would seem to be where the subject matter of the contract is an immaterial object which can be manipulated directly by the smart contract algorithm.

⁶⁷² Nicolas BARBAROUX, Richard BARON, et Amélie FAVREAU. « Blockchain et finance – approche pluridisciplinaire – Les actifs numériques - Section 2 - Les jetons - Art. 1 - Classification informatique et logicielle du jeton », *Dalloz Répertoire IP/IT et Communication*, juin 2020.

L'invention des smart contracts autorise la circulation d'autres actifs, qui ne sont plus nécessairement monétaires : les jetons.

⁶⁷³ Par pointage, il est entendu l'action de simplement signifier une présence dans un programme déployé dans la *blockchain*. Il est pensé au cas d'usage d'un salarié qui pointe après sa journée de travail.

modalités ces dernières devront être considérées comme « ayant eu lieu » ou étant « finales »⁶⁷⁴. En effet, une *blockchain* étant un réseau décentralisé, il n'existe pas d'entité unique confirmant qu'une interaction avec elle a bel et bien eu lieu : il est nécessaire que les nœuds, formant le réseau, parviennent à un consensus sur la validité ou non de cette interaction⁶⁷⁵.

Or ce consensus, qui varie d'une *blockchain* à une autre, peut ne pas avoir été trouvé. L'interaction est alors réversible : tandis qu'on a l'impression d'avoir correctement interagi avec la *blockchain*, cette dernière considère que l'interaction en question n'a jamais eu lieu et ne réalise pas les opérations qui lui ont été prescrites. Par exemple, une partie pourrait initier un paiement dans la *blockchain*, payer les frais de gaz⁶⁷⁶, et avoir l'impression d'avoir dûment rempli son obligation contractuelle alors que son transfert n'a finalement jamais eu lieu en raison d'une défaillance du mécanisme de consensus de la *blockchain*⁶⁷⁷.

Les parties disposent de différentes techniques pour établir la finalité dans une *blockchain*. Dans celles fonctionnant avec le mécanisme de consensus *proof of work*, il est d'usage de considérer une interaction comme finale lorsqu'un nombre relativement élevé de blocs ont été inscrits à la suite de celui dans lequel était située l'interaction⁶⁷⁸. Les parties pourraient alors être tentées de définir une

⁶⁷⁴ « What Is Transactional Finality? | Avalanche Support ». Consulté le 16 mai 2023.
<https://support.avax.network/en/articles/5325234-what-is-transactional-finality>.

Transactional finality is the guarantee that cryptocurrency transactions can't be altered after they've been completed. It is used to measure the amount of time a user has to wait to receive confirmation that a transaction made on the blockchain won't be changed, or canceled.

⁶⁷⁵ Laure De La RAUDIÈRE et Jean-Michel MIS. « Mission d'information commune sur les chaînes de blocs (blockchains) ». Assemblée Nationale, 12 décembre 2018.

Un bloc ne peut être validé et donc s'ajouter à la chaîne que si un consensus des nœuds le permet : les centaines, les milliers voire les dizaines de milliers de copies du registre sont alors mises à jour simultanément et régulièrement, à mesure que les blocs sont minés puis validés.

⁶⁷⁶ Dominique LEGEAIS. « Fascicule 179 : Blockchain », JurisClasseur Droit bancaire et financier, 1 janvier 2020.

Cette monnaie est utilisée pour payer des ressources sur le réseau via un mécanisme appelé « gas ». Les frais de transaction sont ainsi payés en Ether.

⁶⁷⁷ Cela arrive souvent avec les *phénomènes de reOrg* où un bug intervient dans le consensus entre les nœuds ; qui invalident une chaîne de blocs relativement longue.

Toon, Mark. « Polygon Hit by 157-Block 'Reorg' despite Hard-Fork to Reduce Reorgs ». Protos (blog), 24 février 2023.
<https://protos.com/polygon-hit-by-157-block-reorg-despite-hard-fork-to-reduce-reorgs/>.

Reorgs occur when network nodes fall out of sync with each other, and two distinct chains of blocks are produced concurrently. This may be due to a bug, network latency, or even malicious activity. When nodes sync once again, one canonical version of the chain is kept, and the blocks included in the invalid 'fork' are ignored.

⁶⁷⁸ Nabilou, Hossein. « Probabilistic Settlement Finality in Proof-of-Work Blockchains: Legal Considerations ». SSRN Scholarly Paper. Rochester, NY, 31 janvier 2022. <https://doi.org/10.2139/ssrn.4022676>.

interaction comme finale lorsqu'elle figure dans un bloc après lequel x blocs ont été minés ; mais cette définition est spécifique aux blockchains en preuve de travail qui sont de plus en plus rares⁶⁷⁹.

Pour les blockchains fonctionnant avec le *proof of stake*, la confirmation des transactions est différente. Ainsi, sur Ethereum, une transaction est considérée comme finale lorsque deux tiers des validateurs ont voté correctement sur le *checkpoint* d'un bloc de transactions⁶⁸⁰. Sur *Avalanche*, le mécanisme de consensus permettrait d'avoir une finalité approchant une seconde⁶⁸¹. Aussi, les parties auront du mal à abstraire une définition de la finalité qui sera satisfaisante pour toutes les blockchains.

Nous estimons alors qu'un moyen efficace d'établir la finalité d'une interaction serait de s'appuyer sur les services de tiers spécialistes : les explorateurs officiels de *blockchain*⁶⁸². Ceux-ci sont des outils dédiés à l'analyse de *blockchain*, qui disposent de fonctionnalités permettant de connaître avec un haut degré d'assurance si une interaction est finale. Pour toute interaction avec une *blockchain*, il sera fourni un *hash*⁶⁸³, que des parties pourront renseigner à un explorateur afin qu'il renseigne si

As more and more blocks are built on the Bitcoin blockchain, the lower the probability of undoing the embedded transactions, and as the transaction gets deeper and deeper in the blockchain, the probability becomes infinitesimal. At a certain point, this probability becomes so small that it seems that it has persuaded some authors to suggest that the PoW algorithm of the Bitcoin protocol ensures that the extrinsic investment in expended energy would act as a 'thermodynamic guarantee of immutability'.

⁶⁷⁹ Benjamin CURRY et E. Napoletano. « What Is Proof of Stake? How Does It Work? – Forbes Advisor ». Forbes, 16 février 2023. <https://www.forbes.com/advisor/investing/cryptocurrency/proof-of-stake/>.

Proof of stake is becoming more prevalent as a consensus mechanism in the cryptocurrency world. There are currently about 80 different cryptocurrencies that use PoS as the consensus mechanism.

⁶⁸⁰ Prysmatic Labs. « What Happens After Finality in ETH2? » HackMD. Consulté le 16 mai 2023. <https://hackmd.io/@prysmaticlabs/finality>.

Ethereum proof of stake, however, does not function on the concept of probabilistic finality. Instead, it enshrines finality into the protocol by saying "If > 2/3s of validators have voted correctly on the chain head for a long period of time, we can consider everything before a specific checkpoint as finalized".

⁶⁸¹ « What Is Transactional Finality? | Avalanche Support ». Consulté le 16 mai 2023. <https://support.avax.network/en/articles/5325234-what-is-transactional-finality>.

On Avalanche®, transaction finality is usually around one second.

⁶⁸² Le plus connu est etherscan (etherscan.com) pour Ethereum. Nous aurons l'occasion de revenir plus en détail sur eux dans les prochains développements.

⁶⁸³ Coinbase Help. « What Is a Transaction Hash/Hash ID? » Consulté le 18 mai 2023. <https://help.coinbase.com/en/coinbase/getting-started/crypto-education/what-is-a-transaction-hash-hash-id>.

A transaction hash/ID (often abbreviated as tx hash or txn hash) is a unique identifier, similar to a receipt, that serves as proof that a transaction was validated and added to the blockchain. In many cases, a transaction hash is needed in order to locate funds. A transaction hash can be used to: Provide confirmation that the transaction was made (similar to receiving a receipt of purchase when you buy something at a store)

l'interaction est finale. Nous proposons alors la définition suivante :

Confirmation : caractère d'une interaction avec la Blockchain incluse dans un bloc de transactions indiqué comme finalisé et insusceptible de réversibilité par l'explorateur Etherscan/Arbiscan/...

Les parties pourront donc utiliser cette définition chaque fois qu'elles décriront une interaction dans la *Blockchain*, afin de bien préciser que celles-ci doivent être indiquées comme finales par l'explorateur pour être considérées comme telles.

Exemples :

Définition Paiement : désigne le transfert, dans la Blockchain, d'une certaine quantité d'Euros numériques, ayant fait l'objet d'une Confirmation.

Définition d'une Cession des actions : désigne le transfert, dans la Blockchain, d'une certaine quantité d'Actions numériques, ayant fait l'objet d'une Confirmation.

Définition d'un Vote : désigne l'expression du choix par une Partie, dans le Smart contract de Vote, ayant fait l'objet d'une Confirmation.

Définition Actionnement : désigne l'interaction réussie avec une fonction d'un Smart contract dans la Blockchain, ayant fait l'objet d'une Confirmation.

247.

248. Définitions des bogues et piratages dans le contrat. Enfin, il sera indispensable pour les parties de définir les situations dans lesquelles les smart contracts ne fonctionneront pas comme prévu, afin de régler dans le contrat les conséquences juridiques de tels événements⁶⁸⁴. Un *smart contract* peut dysfonctionner à cause d'un bogue ou d'un piratage. La différence entre ces deux est que le second découle d'un acte intentionnel⁶⁸⁵ tandis que le premier a une cause fortuite⁶⁸⁶. Les

⁶⁸⁴ Mekki, Mustapha. « Le smart contract, objet du droit (Partie 2) », Dalloz IP/IT, 2019, 27.

Si le smart contract réduit certains risques, il en crée de nouveaux qu'il convient d'encadrer par des clauses contractuelles qui ne peuvent être algorithmées.

⁶⁸⁵ Larousse. « Définitions : piratage - Dictionnaire de français Larousse ».

Accéder illégalement à un système informatique depuis un ordinateur distant afin d'en consulter les données, de les modifier, voire de les subtiliser.

⁶⁸⁶ Larousse. « Définitions : bogue, bug - Dictionnaire de français Larousse ».

parties pourront opportunément se servir des spécifications fonctionnelles des smart contracts pour leurs définitions : un dysfonctionnement est un comportement déviant de ce que les parties avaient prévu du programme dans la documentation contractuelle.

Piratage : désigne tout acte intentionnel visant à compromettre, altérer ou détourner de quelque manière que ce soit le fonctionnement normal du Smart contract ...tel que prévu dans ses spécifications fonctionnelles.

Bug : désigne toute erreur ou défaillance affectant le fonctionnement normal du Smart contract tel que prévu dans ses spécifications fonctionnelles.

249. Conclusion Section I. Les parties introduiront donc de cette manière leur usage de la *blockchain* dans leur contrat. Elles déclameront leur identité complète en spécifiant leurs adresses cryptographiques. Et elles établiront plusieurs éléments de la terminologie *blockchain* en définissant les termes comme *blockchain*, *smart contract* et jetons. Cette entrée en matière effectuée, elles pourront alors rédiger le préambule et l'objet du contrat en mettant en avant quelques éléments particuliers de la *blockchain*.

Section II - L'opération d'ensemble du contrat

250. Présentation de l'objet du contrat. Les clauses de préambule (§1) et d'objet (§2) auront toutes deux pour but de renseigner les parties et tiers au contrat sur l'opération mise en œuvre par le contrat, dans le contexte particulier qu'est celui d'une exécution par *smart contract*.

§ I – Clause de préambule

251. Un préambule est un texte figurant en tout début de contrat qui vise à le présenter de manière générale. Son contenu consiste le plus souvent à relater le contexte de conclusion de

Défaut de conception ou de réalisation d'un programme informatique, qui se manifeste par des anomalies de fonctionnement de l'ordinateur.

l'accord⁶⁸⁷(A) ainsi que l'opération projetée par les parties (B) ; ce faisant, les lecteurs bénéficient d'une représentation claire de celui-ci, en comparaison avec celle morcelée qu'offre l'examen de chacune des clauses⁶⁸⁸.

A – Contexte de conclusion du contrat

252. Relations antérieures. Avant de rédiger le document *fiat*, il est possible que les parties à un contrat intelligent se soient entendues sur une prestation de développement spécifique de leur smart contracts. Comme évoqué, elles l'auront peut-être développé individuellement, conjointement ou fait développer par un prestataire⁶⁸⁹. Dans tous les cas, à cette occasion, elles auront, au moins, cerné ensemble la mesure de l'exécution de leur convention dans la *blockchain* et constitué de la documentation additionnelle à leur contrat.

Ainsi, elles pourront relater dans le préambule cette étape de la conception de leur contrat intelligent afin de la contractualiser. Cet exercice aura en effet pour but qu'elles se prémunissent de l'apparition de vices de consentement. En spécifiant connaître le fonctionnement du *smart contract*, les parties pourront alors difficilement exciper de l'erreur, du dol, ou d'un manquement à l'obligation d'information sur le fonctionnement du programme⁶⁹⁰.

Les Parties ont développé conjointement des smart contracts conformément à un contrat de

⁶⁸⁷ Marcus Mandel, Isabelle, Tamara Bootherstone, et Pierre Massot. « 7. Les contrats ». In Guide pratique du droit du design, 2e éditio:161-98. Hors collection. Paris: Dunod, 2015. <https://www.cairn.info/guide-pratique-du-droit-du-design-9782100726530-p-161.htm>.

Le préambule est laissé à la libre rédaction des parties. Il sert le plus souvent à éclairer le contexte dans lequel le contrat intervient et pourra servir par exemple pour interpréter certaines clauses.

⁶⁸⁸ Pierre MOUSSERON, Jacques RAYNARD, et Jean-Baptiste SEUBE. Technique contractuelle - Titre I - Dispositions Communes initiales - Chapitre II Le préambule - Section 2 Les fonctions du préambule - §164. 5^e éd. Francis Lefebvre, 2017.

Le préambule sert, surtout, à donner des informations sur le futur, notamment en exposant le projet des parties, l'opération économique qu'elles veulent réaliser, et dont les clauses de l'accord risquent de donner une image brisée, en puzzle.

⁶⁸⁹ V., *infra*, §228

⁶⁹⁰ Marcus Mandel, Isabelle, Tamara Bootherstone, et Pierre Massot. « 7. Les contrats ». In Guide pratique du droit du design, 2e éditio:161-98. Hors collection. Paris: Dunod, 2015. <https://www.cairn.info/guide-pratique-du-droit-du-design-9782100726530-p-161.htm>.

Il peut aussi servir à justifier par écrit que l'une des parties a bien informé l'autre de tel ou tel élément essentiel, ou d'une particularité de tel ou tel élément. Le préambule a la même valeur que le contrat : il oblige les parties. Par exemple, la partie qui a été informée par écrit de tel ou tel élément ne pourra pas par la suite prétendre qu'elle n'en avait pas connaissance. Si rien n'est écrit, il y aura lieu à interprétation, source de remise en cause du contrat. Le préambule est donc assez utile.

développement spécifique figurant en annexe. Elles attestent ainsi connaître précisément le fonctionnement de ces derniers et la manière dont ceux-ci serviront à l'exécution des obligations du présent contrat.

B – Exposition du projet

253. Usage des smart contracts. Dans notre contexte particulier, le préambule servira principalement à décrire la spécificité du contrat : le fait qu'il soit exécuté en totalité ou partie dans la *blockchain* à l'aide de *smart contract*. Malgré que les spécifications fonctionnelles s'en chargeront plus en détail, les parties pourront, dès le préambule, décrire succinctement le rôle que joueront ces smart contracts dans la convention afin d'éclairer les lecteurs du contrat sur l'opération mise en œuvre⁶⁹¹.

Les parties se sont entendues pour faire exécuter une partie de leur contrat de prêt à l'aide de Smart contract déployés dans la Blockchain. Ces derniers seront chargés de permettre le paiement, le remboursement ainsi que la mise en œuvre du mécanisme de garantie du présent contrat.

254. Hiérarchie contractuelle. Le préambule pourra aussi intégrer une clause de hiérarchie contractuelle⁶⁹² afin de renseigner sur la composition du corpus contractuel. Dans notre contexte, les parties préciseront ainsi que l'accord écrit en langage naturel prime sur tout le reste de la documentation contractuelle et qu'il est le document à privilégier en cas de conflit d'interprétation.

Le présent contrat comprend l'accord ci-présent, une annexe technique et les spécifications fonctionnelles des smart contracts. En cas de contradiction entre l'un de ces documents, ce sont les

⁶⁹¹ Blycha, Natasha, et Ariane Garside. « Smart Legal Contracts - A Model for the Integration of Machine Capabilities into Contracts ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 7 décembre 2020. <https://papers.ssrn.com/abstract=3743932>.

The recitals should include a statement to the effect that the contracting parties are entering into the contract as an SLC (...) and that the SLC will operate on the specified digital platform of the contracting parties' choice in order to automate certain parts of the contractual relationship.

⁶⁹² Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires. Chapitre 92 - Clause de priorité. Les Intégrales. LGDJ, 2018

Les contrats sont susceptibles de comporter des informations contradictoires, surtout lorsqu'ils sont longs, complexes, ou que de nombreux documents contractuels ont été échangés par les parties, circonstances qui amplifient les risques de contradictions et donc les sources potentielles de conflits. Les ambiguïtés susceptibles d'en résulter exposent alors les parties aux aléas de l'interprétation judiciaire ou arbitrale de leur contrat. Conscientes de ce risque, les parties peuvent stipuler des clauses de priorité, également appelées clauses de hiérarchie, de classement ou encore de préséance, qui offrent une issue convenue en prévoyant les clauses ou documents qui devront prévaloir en cas de contradiction.

stipulations du présent contrat qui l'emporteront.

§ II – Clause d'objet du contrat

255. **L'opération juridique mise en œuvre.** Après avoir introduit l'économie générale du contrat dans le préambule, les parties pourront décrire plus précisément l'opération juridique de leur contrat intelligent dans la clause "objet du contrat"⁶⁹³. À ce titre, elle leur permettra de spécifier clairement quelle est la qualification juridique qu'elles souhaitent donner à leur contrat (A) ou celle qu'elles souhaitent spécialement écarter (B), compte tenu de leur recours aux smart contracts.

A – Choix d'une qualification juridique précise

256. **Maîtrise du régime juridique.** Donner une qualification déterminée à un contrat revient à choisir le régime juridique applicable à sa convention, ce qui est gage de sécurité juridique⁶⁹⁴. Cette dernière est bienvenue dans notre contexte particulier, puisque le recours à la *blockchain* et aux smart contracts peut être porteur de beaucoup d'originalités pouvant décontenancer un juge ou un arbitre. Il est alors crucial que les parties ne prennent pas le risque de les laisser déterminer par eux-mêmes l'habit juridique le plus approprié à l'opération mise en œuvre dans le contrat.⁶⁹⁵

Elles préciseront en sus des qualifications qu'elles voudront donner à leur contrat, les textes juridiques qui fondent celles-ci, toujours dans le but de ne laisser aucune ambiguïté sur le régime voulu⁶⁹⁶.

Le présent contrat est un contrat de vente au sens des articles ... du code civil dans lequel le Vendeur

⁶⁹³ Eva Mouial Bassilana et Jean-Baptiste Racine. « V° Contrats et obligations - Fasc. 9-1 : CONTRAT. – Contenu du contrat : objet du contrat ». In JurisClasseur Notarial Répertoire, 8 mars 2018.

Plus récemment, l'objet du contrat a pu être décrit comme « l'opération juridique concrète voulue par les parties ». L'intérêt de la notion « est qu'elle implique une vision globale du contrat seule apte à traduire l'opération juridique, et non seulement économique, poursuivie par les parties ».

⁶⁹⁴ V., *infra*, §185

⁶⁹⁵ Eva Mouial Bassilana et Jean-Baptiste Racine. « V° Contrats et obligations - Fasc. 9-1 : CONTRAT. – Contenu du contrat : objet du contrat ». In JurisClasseur Notarial Répertoire, 8 mars 2018.

Quant à l'objet du contrat, il est un outil de qualification pertinent. Il permet une classification des divers contrats suivant leur « fonction économique ».

⁶⁹⁶ A.T. « De la qualification d'une clause pénale », Dalloz Etudiant, 13 décembre 2011. <https://actu.dalloz-etudiant.fr/a-la-une/article/de-la-qualification-dune-clause-penale/h/fd9fb62eeff44a330c3680651dc086cd.html>.

accepte de vendre à l'Acheteur, qui en accepte de faire l'acquisition le bien suivant ...Les parties entendent ainsi donner cette qualification à la présente, dans le respect de l'article 12 alinéa 2 du code de procédure civile⁶⁹⁷.

La présente clause est une clause pénale au sens de l'article .. du code civil à laquelle les parties entendent être liées, conformément à l'article 12 alinéa 2 du code de procédure civile.

B – Exclusion d'une qualification juridique

257. Ecart de régimes inopportuns. Les parties peuvent également se servir de la clause objet du contrat pour expressément écarter l'application de régimes juridiques inopportuns à l'opération mise en œuvre dans leur contrat intelligent. En effet, certaines relations idoines exécutées via des smart contracts peuvent tenter un juge ou un arbitre de les catégoriser dans des régimes définis que les parties souhaitent au contraire éviter⁶⁹⁸.

258. Illustration d'un régime souhaité écarté. Imaginons une entreprise mettant à disposition d'un individu un accès simplifié et ergonomique vers un protocole de prêt décentralisé⁶⁹⁹.

Ainsi, après avoir relevé les termes du mandat et la référence faite aux articles 1142 et 1152 du Code civil, la Haute cour conclut, au visa de l'article 1152 du Code civil, que la clause « avait pour objectif de contraindre le débiteur à exécuter ses engagements, évaluait forfaitairement l'indemnisation en cas d'inéquation et constituait donc une clause pénale ».

⁶⁹⁷ Article 12 du code de procédure civile alinéa 2 : [Le juge] doit donner ou restituer leur exacte qualification aux faits et actes litigieux sans s'arrêter à la dénomination que les parties en auraient proposée.

⁶⁹⁸ Ce comportement se retrouve chez un bon nombre de plateformes crypto qui écartent expressément dans leurs conditions générales d'utilisation l'application de régimes inopportuns ; par exemple les CGU de Equisafe et Aragon :

« CGU Equisafe ». Consulté le 18 mai 2023. <https://www.equisafe.io/cgu-equisafe>.

(...)les Services proposés par EQUISAFE sont des prestations de nature technique et opérationnelle rendues exclusivement au bénéfice et dans le seul intérêt d'un Émetteur d'Actifs (...) L'Utilisateur, l'Investisseur sont informés et reconnaît que dans le cadre des présentes CGU, EQUISAFE agit exclusivement au nom et pour le compte d'un Émetteur d'Actifs et ne fournit aucun service d'investissement au sens de la directive 2014/65/UE (la « MIF 2 ») ni aucun service sur actifs numériques au sens de la loi n°2019-486 du 22 mai 2019 (la « loi Pacte ») et, plus généralement, ne fournit aucune activité réglementée de nature bancaire et financière.

« Terms and Conditions ». Consulté le 18 mai 2023. <https://aragon.org/terms-and-conditions>.

The purpose of the Aragon's tokens ("Tokens") is to be used in a blockchain platform developed by us for the rendering the Services (...)The Tokens are not and are not intended to be a digital currency, security, commodity or any kind of financial instrument.

⁶⁹⁹ Pierre Bordais. « Finance décentralisée et NFT (non fungible token) : deux nouvelles innovations de la blockchain », Revue de droit bancaire et financier, n° 6 (1 novembre 2021).

Au-delà d'une logique de décentralisation propre à l'écosystème blockchain (qui n'est pas toujours atteinte), [la finance

Dans ce dernier, il est possible d'emprunter des cryptoactifs à un ensemble de *smart contract*, mais le protocole, bien qu'ayant été développé par l'entreprise mettant à disposition cet accès, n'est plus du tout sous son contrôle⁷⁰⁰. Le client n'emprunte donc pas à l'entreprise mettant à disposition cet accès facilité. Dans leur conditions générales d'utilisation, l'entreprise veillera alors à bien disqualifier le contrat de contrat de prêt, sans pour autant choisir un régime spécifique pour draper la relation idoine dans laquelle elle se trouve.

Cette situation est celle à laquelle sont confrontées de nombreuses entreprises de protocoles de finance décentralisée qui sont parvenus à être décentralisés : après avoir déployé leur smart contracts et les avoir rendu autonomes, ils proposent des services destinés à interagir avec eux, qui ne consistent toutefois pas dans le cœur du produit en lui-même. De crainte que la confusion soit faite par des juges ou des arbitres, elles précisent systématiquement dans leur conditions générales d'utilisation que le service fourni n'est pas celui de prêt, d'échange, de mandat... Ainsi, dans les toutes premières lignes des conditions générales *UniSwap*, sont écrites ces lignes :

Ces Conditions d'utilisation (l'"Accord") expliquent les termes et conditions en vertu desquels vous pouvez accéder et utiliser les Produits fournis par Uniswap Labs (...). Les Produits comprennent, sans nécessairement s'y limiter (...) une interface utilisateur hébergée sur un site web (l'"Interface" ou "Application"). L'Interface est distincte du Protocole et constitue un moyen, mais non exclusif, d'accéder au Protocole. (...) Uniswap Labs ne contrôle ni n'exploite aucune version du Protocole sur aucun réseau blockchain. En utilisant l'Interface, vous comprenez que vous n'achetez ni ne vendez des actifs numériques auprès de nous, et que nous n'exploitons aucune piscine de liquidité sur le

décentralisée] s'agit avant tout d'automatiser et de faciliter l'accès aux outils financiers traditionnels. S'agissant du prêt, quelques clics suffisent pour prêter ou emprunter des sommes plus ou moins grandes. Loin de la lourdeur d'un emprunt bancaire traditionnel, les emprunts en cryptomonnaie permettent un gain substantiel de célérité pour le financement d'opérations diverses.

⁷⁰⁰ Blockchain Partner. « Comprendre La DeFi (Decentralized Finance) : Définition, Usages, Enjeux, Perspectives – Blockchain Partner ». Consulté le 20 janvier 2022. <https://blockchainpartner.fr/comprendre-open-finance-definition-usages-enjeux-perspectives/>.

La différence ontologique [avec la finance centralisée] est que dans l'Open Finance il n'existe pas d'intermédiaire bureaucratique ou technologique entre le constructeur et son idée. Ceux qui construisent des applications en Open Finance savent que dans cet univers, les intermédiaires peuvent seulement limiter l'accès au niveau de l'interface, mais ne peuvent pas empêcher de développer et de déployer une logique fondamentale en premier lieu.

259. Conclusion de la section II et du chapitre I. Les premières clauses de l'accord écrit en langage naturel d'un contrat intelligent seront ainsi consacrées à la présentation des parties, des termes et de l'opération mise en œuvre. S'agissant de la présentation des parties, ces dernières devront être vigilantes à déclamer leur identité complète ainsi que leurs adresses cryptographiques. Pour la définition des termes, elles rédigeront la terminologie spécifique de la *blockchain* ; cette dernière sera définie à l'aune du phénomène de *fork*. Enfin dans les clauses de préambule et d'objet du contrat, les parties présenteront le contexte et l'opération mise en œuvre à travers l'exécution par *smart contract*, et préciseront les qualifications juridiques auxquelles elles souhaitent ou non être soumises.

⁷⁰¹ Uniswap Protocol. « Terms of Service | Uniswap Labs ». Consulté le 18 mai 2023. <https://uniswap.org/terms-of-service>.

Chapitre II - Stipulations relatives à l'exécution du contrat

260. Corps du contrat. Une fois les présentations générales faites, les parties pourront être en mesure de rédiger les clauses relatives à l'exécution du contrat : c'est-à-dire celles relatives à la vie du contrat intelligent (Section I). Certains tiers pourront se trouver mêlés à l'exécution de la convention ; cette dernière devra donc également organiser leurs rapports avec les parties (Section II).

Section I – Les clauses intéressant les parties au contrat

261. La vie du contrat. Les stipulations relatives à l'exécution du contrat aborderont nécessairement les obligations des parties⁷⁰²(§1). Afin de parer aux événements imprévisibles des smart contracts et de la *blockchain*, les parties seront également avisées d'apporter un soin particulier à la rédaction de la clause d'imprévision (§2) et de la clause de responsabilité (§3).

§ I - Clauses relatives aux obligations

262. Plan. Nonobstant l'usage qu'elles feront des smart contracts dans leur contrat, les parties devront toujours prendre le soin de décrire précisément comment leurs obligations seront éteintes à travers le recours à ces derniers (A). L'obligation de paiement d'argent, qui constitue une des obligations les plus récurrentes⁷⁰³des contrats à titre onéreux⁷⁰⁴et donc adéquates à être

⁷⁰² Pierre MOUSSERON, Jacques RAYNARD, et Jean-Baptiste SEUBE. Technique contractuelle – Titre 3 Dispositions relatives à l'existence des relations contractuelles - Chapitre premier Dispositions relatives aux effets du contrat – Section I Les effets personnels du contrat - § 502. 5^e éd. Francis Lefebvre, 2017.

Si certains contrats seulement développent des effets créateurs de personnes morales ou relatifs aux droits réels, tous, en revanche, font naître des obligations. C'est là l'effet ordinaire et même permanent du contrat. Les différentes catégories de contrats se caractérisent par les systèmes originaux d'obligations qu'elles produisent.

⁷⁰³ Qu'est-ce qu'un marché public ? « Qu'est-ce qu'un marché public ? », s. d. Consulté le 20/05/2023 <https://www.demarches.interieur.gouv.fr/professionnels/qu-est-ce-qu-un-marche-public>.

Contrat onéreux | Ce type de contrat se définit par opposition au contrat à titre gratuit. Dans un contrat à titre onéreux, chaque contractant reçoit une contrepartie (généralement le paiement d'une somme en argent) en échange de la réalisation d'une prestation.

⁷⁰⁴ Article 1107 du code civil : *Le contrat est à titre onéreux lorsque chacune des parties reçoit de l'autre un avantage en contrepartie de celui qu'elle procure.*

matérialisée *on-chain*⁷⁰⁵, méritera une attention particulière dans la rédaction de l'accord écrit en langage naturel d'un contrat intelligent (B).

A – Exécution des obligations par *smart contract*

263. Modalités d'usage d'un *smart contract* dans l'exécution des obligations. Les smart contracts peuvent, globalement, être utilisés de deux manières dans l'exécution des obligations d'un contrat :

- ils peuvent être chargés d'accomplir matériellement, à la place des parties, les tâches qui leur incombent au titre des obligations stipulées dans le contrat⁷⁰⁶. Par exemple, un *smart contract* peut être configuré afin de verser automatiquement l'indemnité qu'une partie doit au titre d'une clause pénale⁷⁰⁷;
- mais nous avons déjà évoqué que l'exécution par *smart contract* ne signifie pas nécessairement une automatisation des tâches⁷⁰⁸. Ils peuvent aussi être utilisés comme des *medium* à travers lesquels les parties exécutent leurs obligations. Par exemple, dans un contrat de vente à distance, un *smart contract*, constitué en tant que séquestre, pourra être mobilisé afin de réceptionner le versement du prix d'achat qu'aura effectué l'acheteur, que le vendeur pourra ensuite retirer après qu'un oracle ait indiqué que le bien ait été correctement reçu. Dans cette hypothèse, les parties versent et retirent elles-mêmes les montants, et le *smart contract* sert uniquement à sécuriser la transaction⁷⁰⁹.

⁷⁰⁵ V., *infra*, §51

⁷⁰⁶ Roda Jean-Christophe. « Smart contracts, dumb contracts ? » Dalloz IP/IT, n° 07-08 (4 juillet 2018): 397.

(...) les smart contracts sont présentés comme des programmes informatiques permettant d'exécuter automatiquement les termes du contrat. Il est également question de contrats qui s'auto-exécutent.

⁷⁰⁷ V., *infra*, §128

⁷⁰⁸ V., *infra*, §6

⁷⁰⁹ Cet usage est bien illustré dans le *smart contract* d'achat à distance donné en exemple sur le site officiel de la documentation Solidity (le langage de programmation de *smart contract* sur *Ethereum*).

« Solidity by Example — Solidity 0.8.13 documentation ». Consulté le 19 mai 2023.
<https://docs.soliditylang.org/en/v0.8.13/solidity-by-example.html#safe-remote-purchase>.

Purchasing goods remotely currently requires multiple parties that need to trust each other (...). In the following example, both parties have to put twice the value of the item into the contract as escrow. As soon as this happened, the money will stay locked inside the contract until the buyer confirms that they received the item. After that, the buyer is returned the value (half of their deposit) and the seller gets three times the value (their deposit plus the value). The idea behind this is that both parties have an incentive to resolve the situation or otherwise their money is locked forever.

Quel que soit le cas de figure, les parties devront donc décrire comment leurs obligations seront éteintes à travers leur interaction avec leurs smart contracts.

264. Description des modalités d’accomplissement dans la *blockchain*. Conformément à notre proposition d’architecture de corpus contractuel⁷¹⁰, elles peuvent choisir entre décrire ces modalités d’accomplissement de ces obligations dans un seul document ou plutôt les séparer dans deux documents différents : l’accord écrit principal et l’annexe technique⁷¹¹. Imaginons un contrat de prêt entre Monsieur B et Madame A exécuté par le biais d’un *smart contract*. Les parties peuvent décider de décrire les obligations du contrat en les découpant ainsi :

- dans l’accord écrit principal, la clause relative aux obligations contiendra classiquement les tâches que doivent accomplir les parties avec la mention que les modalités techniques d’accomplissement de ces tâches, c’est-à-dire par *smart contract*, seront précisées dans l’annexe technique :

M. B devra rembourser à Mme. A, avant le 01/01/2022, la somme de 2000 euros avec des intérêts de 10 % . À défaut de remboursement avant cette échéance, la créance de M. B affectée en garantie du prêt sera saisie par Mme A. Les modalités techniques de remboursement et de mise en œuvre de la garantie sont décrites dans l’annexe X du présent contrat.

- dans l’annexe technique figurera la manière dont les parties pourront s’y prendre pour

⁷¹⁰ V., *infra*, §220

⁷¹¹ Ce procédé ressemblerait quelque peu à celui des contrats informatiques dans les nuages (contrats cloud). Les obligations principales figurent dans la convention principale et les engagements de niveaux de services (qui sont les obligations spécifiques mais essentielles du contrat) figurent dans une annexe.

Vivant Michel. « Lamy droit du numérique » - Partie 3 Numérique et contrats - Division 3 Les principaux contrats du numérique et leurs spécificités - Chapitre 6 Les contrats d’informatique dématérialisée (cloud computing) - Section 5 Les clauses essentielles des contrats de cloud - § 2. La clause de qualité du service attendu. Editions Lamy, 2012.

Afin que le client puisse vérifier que le service répond à ses besoins, la mise en place d'un Service Level Agreement (dit « SLA ») s'est généralisée. Il s'agit d'un document dans lequel le prestataire formalise la qualité du service et précise notamment les modalités, la performance du service (temps de réponse, temps de transmission des données ...), la disponibilité des applications (horaires d'ouverture et de fermeture, périodes d'indisponibilités...). (...) Cette clause fait partie intégrante du contrat de cloud et fait le plus souvent l'objet d'une annexe.

accomplir leurs droits et obligations dans la *blockchain* :

Pour rembourser Mme A, M. B devra Actionner la fonction « rembourser() » du Smart contract de Prêt avant la date d'échéance, qui Transférera 2200 Euros Numériques de son Adresse Blockchain vers celle du Smart contract de Prêt. Mme A pourra récupérer ces Euros numériques en Actionnant la fonction « récupérerSomme () » du Smart contract de Prêt.

Si M. B n'effectue pas ce remboursement avant la date d'échéance, Mme A pourra Actionner la fonction « recupererGarantie() » du Smart contract de Prêt afin de Transférer à son Adresse Blockchain la Créance Numérique affectée en garantie.

Dans les contrats qu'elles souhaiteront moins verbeux, les parties pourront préférer indiquer dans un seul document comment les obligations devront être exécutées à travers les smart contracts. Pour ne laisser aucune place à l'ambiguïté sur la qualification de l'obligation qu'elles entendront éteindre, elles seront avisées de la déclamer clairement avant de préciser les modalités techniques de son extinction⁷¹². Exemple :

Le Client s'engage à payer la somme de X euros pour chaque sollicitation de l'API du Prestataire.

Ce paiement sera déclenché après chaque Actionnement, par l'Oracle, de la fonction « pull () » du Smart contract de l'API, qui Transférera, à l'Adresse Blockchain du Prestataire, X Euros Numériques.

265. Description de l'interaction avec le *smart contract*. En sus de la description des obligations *stricto sensu*, les parties seront également avisées de formuler des consignes générales sur l'interaction avec les smart contracts. Nous avons déjà évoqué en quoi il sera nécessaire qu'elles précisent ce qu'est une action confirmée dans la *blockchain*⁷¹³ afin qu'elles soient certaines que l'accomplissement de leurs obligations par ce biais aient bien été pris en compte. Elles pourront préciser, en outre, que chacune d'entre elles s'assurera d'interagir avec les smart contracts conformément à leurs fonctionnements décrits dans les spécifications fonctionnelles. Par exemple, si

⁷¹² Pierre MOUSSERON, Jacques RAYNARD, et Jean-Baptiste SEUBE. Technique contractuelle - §320. 5^e éd. Francis Lefebvre, 2017.

Il convient, donc, de rédiger ces clauses (de détermination de l'ouvrage) de façon précise en décrivant sinon exactement du moins au mieux la prestation due et en précisant toutes les obligations de faire et de ne pas faire les ayant pour objet.

⁷¹³ V., *infra*, §246

un contrat de fourniture d'électricité est exécuté dans la *blockchain*⁷¹⁴, le paiement de la facture ne pourra être considéré comme réalisé que si un certain jeton est envoyé à un *smart contract* de la société, provenant d'une adresse spécifique et à une certaine date.

Les Parties s'assurent d'interagir avec les smart contracts X conformément à leurs fonctionnements, détaillés dans les spécifications fonctionnelles figurants en annexe de ce contrat.

266. Obligation de bonne foi. Les parties pourront également prévoir une obligation générale de bonne foi⁷¹⁵ spécifique à l'utilisation des smart contracts. En effet, alors même qu'un individu peut techniquement interagir, en apparence, de manière conforme avec le fonctionnement d'un *smart contract*, il peut toutefois le faire de manière malhonnête et contraire à son esprit de conception.

Autrement dit, une personne peut interagir avec un *smart contract* sans l'avoir techniquement piraté, mais pourtant de mauvaise foi. Imaginons une clause statutaire prévoyant qu'un individu puisse devenir actionnaire *on-chain* : un apport à travers le *smart contract* de la société lui permet d'obtenir des actions représentées par des jetons. La société émettrice des jetons peut coder un programme prévoyant qu'à chaque fois qu'un individu tentera de racheter une trop grosse quantité d'actions, elle acquerra avant ce dernier la quantité d'actions qu'il souhaitait obtenir, faisant systématiquement faillir sa tentative d'investissement⁷¹⁶. Dans cette hypothèse, la société n'a ni piraté, ni mal agi

⁷¹⁴ Cohn, A., T. A. P. West, et Chelsea Parker. « Smart after all : Blockchain, Smart contracts, Parametric insurance and Smart energy grids », 2017. <https://www.semanticscholar.org/paper/SMART-AFTER-ALL%3A-BLOCKCHAIN%2C-SMART-CONTRACTS%2C-AND-Cohn-West/829c0b86c8ef2eac18b3e6cbaf4c2cf93eb9409a>.

In the energy industry, blockchain-based smart contracts can enable smart grids, microgrids, and other types of innovative grid management technologies by both providing the mechanism for automating value transmission and the means to streamline transaction administration.

⁷¹⁵ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires - Chapitre 11 Clause de bonne foi - § I - Objet et utilité - 162 Notion. Les Intégrales. LGDJ, 2018.

La clause de bonne foi a pour objet premier d'attirer l'attention des contractants sur l'exigence de bonne foi à laquelle ceux-ci doivent se conformer dans l'exécution de leurs obligations. Autrement dit, elle peut, de prime abord, apparaître comme un simple rappel, hier, de l'article 1134, alinéa 3, du Code civil aux termes duquel les conventions devaient être exécutées de bonne foi et, désormais, du nouvel article 1104, prévoyant de manière plus large encore, et au titre d'un principe directeur, que « les contrats doivent être négociés, formés et exécutés de bonne foi. Cette disposition est d'ordre public ».

⁷¹⁶ Il s'agit d'un procédé malveillant récurrent dans le monde crypto appelé MEV pour Maximum Extractable Value.

Filippi, Primavera de, Bruno Deffains, et Philémon Poux. « « Maximal Extractable Value » ou la Tragédie des blockchainsss en tant que Communs ». Terminal. Technologie de l'information, culture & société, n° 136 (4 avril 2023). <https://doi.org/10.4000/terminal.9006>.

Cette forme de MEV repose sur la possibilité, pour certains acteurs, d'insérer certaines transactions avant les autres. Le frontrunning permet à ceux qui le pratiquent de « voler » des transactions rentables, en les reproduisant (à leur bénéfice) pour inclure ces transactions modifiées plus tôt dans le bloc. En particulier, les ordres d'arbitrage ou de liquidation sont

conformément aux spécifications fonctionnelles de son *smart contract*, mais l'usage qu'elle en a fait est manifestement de mauvaise foi. Pour ne pas être redondant avec l'article 1104 du code civil⁷¹⁷, les parties pourront spécifier en fonction de leur contexte particulier ce qu'elles estiment être une interaction de mauvaise foi.

Les Parties s'engagent à interagir avec les smart contracts en toute bonne foi, en conformité avec leur esprit de conception. Une interaction de type valeur extractible maximale telle que définie à l'article ... sera à considérer comme de mauvaise foi.

267. Convention de preuve. Enfin, les parties pourront prévoir une convention de preuve afin de conférer une certitude juridique à leurs obligations exécutées dans la *blockchain*⁷¹⁸. L'article 1356 du code civil rappelle, en effet, que la preuve n'est pas, en principe, d'ordre public et qu'elle peut faire l'objet, dans une certaine mesure, d'un aménagement contractuel⁷¹⁹. Une clause de convention de preuve pourra alors modifier les règles normales de la preuve entre les parties, dès lors que ces dernières respecteront l'ordre public⁷²⁰ et ne l'imposeront pas à des consommateurs⁷²¹.

des cibles courantes du frontrunning.

⁷¹⁷ Article 1104 du code civil : *Les contrats doivent être négociés, formés et exécutés de bonne foi.*

⁷¹⁸ Mekki, Mustapha. « Blockchain : l'exemple des smart contracts ». Consulté le 20 mai 2023.
<https://mustaphamekki.openum.ca/publications/1314/>.

Enfin, en raison des incertitudes sur les causes d'un dommage, en raison de la multiplicité des intervenants et/ou de la complexité de la technologie utilisée, il est préférable de déterminer par des clauses l'objet, la charge et les modes de preuve.

⁷¹⁹ Article 1356 du code civil : *Les contrats sur la preuve sont valables lorsqu'ils portent sur des droits dont les parties ont la libre disposition. Néanmoins, ils ne peuvent contredire les présomptions irréfragables établies par la loi, ni modifier la foi attachée à l'aveu ou au serment. Ils ne peuvent davantage établir au profit de l'une des parties une présomption irréfragable. Les contrats sur la preuve sont valables lorsqu'ils portent sur des droits dont les parties ont la libre disposition. Néanmoins, ils ne peuvent contredire les présomptions irréfragables établies par la loi, ni modifier la foi attachée à l'aveu ou au serment. Ils ne peuvent davantage établir au profit de l'une des parties une présomption irréfragable.*

⁷²⁰ Collectif. « Chapitre 2 - Les conventions de preuve - §67 ». In Dictionnaire permanent - Droit des affaires - Archivage. ELNET, s. d.

Les conventions de preuve doivent notamment respecter les principes suivants : ne pas déroger aux règles d'ordre public, comme notamment la force probante des actes authentiques ; se conformer aux règles substantielles de l'administration judiciaire de la preuve, telles que le respect du contradictoire ou la possibilité de discuter la preuve versée aux débats ; garantir systématiquement la possibilité d'apporter la preuve contraire ; en effet, interdire à une partie d'apporter la preuve contraire ne constitue pas un aménagement d'une règle de preuve mais un aménagement d'une règle de fond touchant l'ordre public ; respecter l'équilibre des intérêts en présence, la convention ne devant pas avoir pour objet ou pour effet d'avantager l'une des parties.

⁷²¹ Article R212-1 du code de la consommation : *Dans les contrats conclus entre des professionnels et des consommateurs, sont de manière irréfragable présumées abusives, au sens des dispositions des premier et quatrième alinéas de l'article L. 212-1 et dès lors interdites, les clauses ayant pour objet ou pour effet de : (...) 12° Imposer au consommateur la charge de la preuve, qui, en application du droit applicable, devrait incomber normalement à l'autre partie au contrat.*

Dans notre contexte, et dans la mesure de ce qui est légalement possible, une convention de preuve pourra prévoir que certaines opérations ayant eu lieu dans la *blockchain* vaudront preuve de l'accomplissement des obligations stipulées⁷²². De sorte qu'en cas de contentieux, les parties pourront valablement se servir des activités réalisées dans la *blockchain* comme moyens de preuve à présenter devant le juge et rendre opposables entre elles⁷²³.

Les parties procéderont en commençant par identifier précisément ces obligations juridiques qu'elles souhaitent pouvoir prouver par des opérations dans la *blockchain*. Puis elles définiront tout aussi précisément ces dernières et stipuleront qu'elles valent preuve de la réalisation des obligations juridiques susmentionnées. Nous verrons plus tard qu'il est possible de configurer des smart contracts afin qu'ils envoient des signaux spéciaux lorsqu'une certaine action a eu lieu dans la *blockchain*⁷²⁴. Les parties pourront se servir de cette fonctionnalité comme « opération » dans la *blockchain* prouvant l'exécution d'une obligation. Par exemple, dans un contrat de prêt, elles pourront prévoir que :

Les Parties entendent utiliser les données figurant dans la Blockchain afin de prouver l'accomplissement de certaines opérations des présentes. Ainsi, elles reconnaissent que la mise à disposition de la somme empruntée, son remboursement et la mise en œuvre de la garantie seront valablement prouvés par l'Emission des Evènements respectifs suivants : « sommeDéposée() », « pretRemboursé() », « garantieRéalisée() » dans la Blockchain. Ces données issues de ces évènements pourront être produites en justice en cas de litiges, y compris dans les litiges opposant les Parties.

⁷²² Mathieu MARTIN. « Contrat de l'informatique - Pratique contractuelle. Contrats de l'informatique. Les clauses de convention de preuve », Communication commerce électronique, n° 3 (1 mars 2021).

L'organisation d'une convention de preuve peut passer par 2 procédés : Un procédé technologique auquel les parties reconnaissent toute légitimité et leur accordent la confiance : l'exemple le plus communément utilisé et celui d'une carte magnétique et la composition concomitante d'un code confidentiel, comme validé par la Cour de cassation dans son arrêt « Crédicas » (Cass. 1re civ., 8 nov. 1989, n° 86-16.197). Notons que cette même qualification peut s'appliquer dans le cadre de la blockchain et plus particulièrement des smart contracts, où un algorithme s'applique automatiquement si certaines conditions prédéfinies sont remplies.

⁷²³ Barbry, Éric. « Smart contracts... Aspects juridiques ! » Annales des Mines - Réalités industrielles Août 2017, n° 3 (2017): 77-80. <https://doi.org/10.3917/rindu1.173.0077>.

La troisième question porte sur l'opposabilité des smart contracts aux juges (notamment de leurs éléments issus de la blockchain). Ici, le droit est déjà prêt, en France tout du moins, grâce au droit des « conventions de preuve ». Articles 1353 et 1368 du Code civil, qui permettent de définir entre parties contractantes les règles d'opposabilité en termes de preuve. Cependant, ces conventions de preuve devront assurément être conclues avant les smart contracts et ne pourront, quant à elles, être codifiées.

⁷²⁴ V., *supra*, §429

B – Obligation de paiement monétaire

268. Nécessité d'organiser l'usage de *stablecoin* dans le contrat. Les parties seront très certainement amenées à stipuler des obligations de versement d'argent dans leur contrat intelligent⁷²⁵. Conformément à nos recommandations sur l'hybridation⁷²⁶, elles seront avisées, alors, de ne chercher à utiliser que des *stablecoin* pour éteindre ces types d'obligations. Or, en l'absence de monnaie numérique de banque centrale fonctionnant sur une *blockchain*, elles seront toujours confrontées à des difficultés pour réaliser ces obligations à l'aide d'actifs tokenisés⁷²⁷. En effet, les *stablecoin* utilisables aujourd'hui n'ont pas le même pouvoir libératoire universel que la monnaie légale. Cela signifie que les parties devront conventionnellement organiser leur recours afin de leur conférer le pouvoir de libérer un débiteur d'une obligation de versement d'une somme d'argent⁷²⁸. Autrement dit, elles devront prévoir une clause évoquant spécialement le recours aux *stablecoin*. Nous distinguons deux types de ces actifs que les parties peuvent actuellement mobiliser pour régler les obligations monétaires ; qui feront ainsi varier l'écriture de la clause dédiée aux modalités de paiement de sommes d'argent dans la *blockchain*.

269. Monnaie électronique. Les parties peuvent utiliser des *stablecoin* ayant la qualification de monnaie électronique⁷²⁹, ou dit autrement, de la monnaie électronique tokenisée. Celle-ci est considérée comme étant à même d'éteindre les obligations de paiement de sommes

⁷²⁵ V., *infra*, §39

⁷²⁶ V., *infra*, §48

⁷²⁷ Dans l'hypothèse où la banque centrale européenne décide de lancer une MNBC, elle envisagerait de lui donner cours légal. European Central Bank. « Report on a digital euro », octobre 2020.
https://www.ecb.europa.eu/pub/pdf/other/Report_on_a_digital_euro~4d7268b458.en.pdf.

Legal tender status would be a desirable feature of the digital euro.

⁷²⁸ Malika Douaoui-Chamseddine. « Cryptomonnaie - Paiement en crypto-monnaie ou jetons de monnaie électronique et article L. 622-7, I du Code de commerce . - Le paiement en crypto-monnaie ou en jetons de monnaie électronique des créances de somme d'argent antérieures ou postérieures non privilégiées à l'épreuve de l'article L. 622-7, I du Code de commerce », *Revue de droit bancaire et financier*, n° 2 (1 mars 2023).

La crypto-monnaie se distingue aussi de la monnaie numérique de banque centrale, véritable monnaie électronique. Elle n'est pas comme précédemment indiqué de la monnaie légale, car elle n'a pas cours légal, par opposition à cette dernière, faute d'avoir un pouvoir libératoire universel. La remise de crypto-monnaie comme le BTC par exemple, ne libère pas le débiteur de plein droit de sa dette. L'euro est la seule monnaie ayant cours légal en France (C. mon. fin., art. L. 111-1. – Et C. civ., art. 1343-3). Les crypto-monnaies ne peuvent être imposées en paiement, sauf à violer l'article R. 162-2 du Code monétaire et financier. C'est pourquoi, le débiteur ne sera libéré de sa dette que si son créancier accepte ce mode de paiement.

⁷²⁹ Article 3.§1.(4) du règlement sur les marchés de cryptoactifs : *jeton se référant à un ou des actifs : un type de crypto-actif qui vise à conserver une valeur stable en se référant à la valeur de plusieurs monnaies fiat qui ont cours légal, à une ou plusieurs matières premières ou à un ou plusieurs crypto-actifs, ou à une combinaison de tels actifs;*

d'argent visées à l'article 1343-3 du code civil⁷³⁰ bien qu'elle n'ait pas cours légal⁷³¹. En tant que telle, cela signifie qu'une personne ne pourra donc pas être contrainte d'accepter d'être payée en monnaie électronique : les parties devront donc prévoir dans leur clause qu'elles consentent à n'utiliser que cette forme de monnaie pour régler leurs obligations monétaires⁷³².

Les parties conviennent que tous les paiements de sommes d'argent stipulés dans ce contrat seront effectués en monnaie électronique au sens des articles L315-1 et suivants du code monétaire et financier. Les unités de cette monnaie électronique sont matérialisées dans la Blockchain par les Euros Numériques.

270. Seuil de la monnaie électronique. Actuellement, l'usage de ces actifs pourra poser des grandes difficultés aux parties car les articles L112-6 et D112-3 du Code monétaire et financier imposent des plafonds maximums aux montants pouvant être payés par le biais de monnaies électroniques. Ceux-ci sont de 3000, 10 000 ou 15 000 euros⁷³³ en fonction du *lieu du domicile fiscal du débiteur, la finalité professionnelle ou non de l'opération et de la personne au profit de laquelle le paiement est effectué*⁷³⁴. Aussi, sauf à espérer que le législateur national relève ces plafonds, le recours à la monnaie électronique tokenisée ne pourra convenir que pour les contrats mettant en jeu

⁷³⁰ Article 1343-3 du code civil : *Le paiement, en France, d'une obligation de somme d'argent s'effectue en euros.*

⁷³¹ Lansky S. « La nature juridique de la monnaie électronique », Bulletin de la Banque de France, n° 70 (octobre 1999).

Compte tenu du fait que la monnaie électronique ne dispose pas du régime du cours légal ou forcé, le porteur de cette monnaie doit toujours avoir le droit de demander à l'émetteur la conversion des unités électroniques contenues dans le PME, en monnaie fiduciaire ou scripturale.

⁷³² Article L315-7 du code monétaire et financier : *Le contrat liant l'émetteur et le détenteur de monnaie électronique établit clairement les conditions et le délai de remboursement des unités de monnaie électronique.*

⁷³³ Article D112-3 du code monétaire et financier : *I. – Le montant prévu au I de l'article L. 112-6 est fixé : 1° Lorsque le débiteur a son domicile fiscal sur le territoire de la République française ou agit pour les besoins d'une activité professionnelle, à 1 000 euros pour les paiements effectués en espèces et à 3 000 euros pour les paiements effectués au moyen de monnaie électronique ; 2° Lorsque le débiteur justifie qu'il n'a pas son domicile fiscal sur le territoire de la République française, n'agit pas pour les besoins d'une activité professionnelle et paie une dette au profit d'une personne qui n'est pas mentionnée à l'article L. 561-2, à 10 000 euros pour les paiements effectués en espèces ou au moyen de monnaie électronique ; 3° Lorsque le débiteur justifie qu'il n'a pas son domicile fiscal sur le territoire de la République française, n'agit pas pour les besoins d'une activité professionnelle et paie une dette au profit d'une personne mentionnée à l'article L. 561-2, à 15 000 euros pour les paiements effectués en espèces ou au moyen de monnaie électronique.*

⁷³⁴ Article L112-6 du code monétaire et financier : *I. – Ne peut être effectué en espèces ou au moyen de monnaie électronique le paiement d'une dette supérieure à un montant fixé par décret, tenant compte du lieu du domicile fiscal du débiteur, de la finalité professionnelle ou non de l'opération et de la personne au profit de laquelle le paiement est effectué...*

des sommes modestes⁷³⁵.

271. Monnaie conventionnelle. Alternativement, les parties peuvent décider d'utiliser d'autres *stablecoin*, qui n'ont pas le statut de monnaie électronique ou légale, mais qui peuvent avoir celui de « monnaie contractuelle »⁷³⁶. Afin que leur recours ne bouleverse pas le régime juridique du contrat, elles pourront recourir à la méthode de l'obligation facultative⁷³⁷. Elles prévoiront dans leur clause que les paiements en sommes d'argent se feront en euros, mais qu'elles conviennent, au titre d'une obligation facultative, que celles-ci pourront être éteintes par le transfert de *stablecoin* dans la *blockchain*⁷³⁸. L'objet principal de l'obligation est le paiement en euros de la somme d'argent, mais le débiteur a la faculté, pour se libérer, de payer en *stablecoin*.

Les avantages de cette approche sont multiples : d'abord elle permet, théoriquement, de contourner le plafond de la monnaie électronique. En effet, contrairement à cette dernière, lorsque les *stablecoin* sont utilisés en tant que mode facultatif d'extinction d'une obligation, ils ne sont pas soumis à aucune limite de plafond. L'autre avantage est que cet artifice juridique ne bouleverse pas le régime des obligations de paiement en euros : l'obligation de paiement reste en euros, c'est seulement à titre subsidiaire qu'elle est réglée en *stablecoin*. Cette approche est donc la seule permettant d'utiliser légalement les *stablecoin* comme moyen de paiement tout en conservant le libellé des obligations

⁷³⁵ Matthieu Lucchesi et Bastien Raisse. « Règlement Régime Pilote - Le règlement européen sur le régime pilote : l'innovation réglementaire pour les infrastructures de marché en blockchain face au défi de sa mise en œuvre », Revue de Droit bancaire et financier, n° 5 (octobre 2022).

(...) en droit français le recours à la monnaie électronique obéit à certaines contraintes. Le montant des transactions est notamment limité à certains seuils définis précisément (C. mon. fin., art. L. 112-6 et D. 112-3). Lorsque MiCA aura été finalisé, et si ces seuils sont maintenus pour la monnaie électronique (y compris sous forme tokenisée), le recours aux jetons de monnaie électronique dans le cadre du règlement Régime Pilote serait très compliqué compte tenu desdites limitations de montant.

⁷³⁶ Jérôme Huet. « Le bitcoin, dont la légalité paraît admise, est une sorte de monnaie contractuelle », Revue des contrats, n° 1 (1 mars 2017): 54.

Le bitcoin, qualifié au niveau communautaire de « monnaie virtuelle », dont la conversion avec une monnaie traditionnelle échappe à la TVA et qui ne tombe pas sous les incriminations protégeant la monnaie nationale ou la monnaie émise par une union monétaire, apparaît comme une monnaie privée, comme une monnaie contractuelle.

⁷³⁷ Article 1308 du code civil : *L'obligation est facultative lorsqu'elle a pour objet une certaine prestation mais que le débiteur a la faculté, pour se libérer, d'en fournir une autre...*

⁷³⁸ Autorité des marchés financiers. « Analyse sur la qualification juridique des produits dérivés sur crypto-monnaies », 22 octobre 2018.

En pratique, cette différence de cours est réglée en euros (ou dans une autre monnaie ayant cours légal) d'où l'appellation de « règlement en espèces » mais cette différence pourrait très bien être réglée dans un autre actif (y compris une crypto-monnaie) sans que cela n'influe sur la qualification du contrat. Juridiquement, les parties à un contrat ont en effet la possibilité de prévoir que le paiement du différentiel s'effectuera en autre chose que la somme d'argent convenue, soit une dation en paiement au titre de laquelle la contrepartie débitrice se libérera de sa dette par un bien différent de celui initialement prévu.

d'argent en euros⁷³⁹ ; ce qui préserve la qualification et le régime de certains contrats⁷⁴⁰.

L'inconvénient toutefois de cette technique est que l'obligation facultative telle que définie à l'article 1308 du Code civil autorise le débiteur à choisir librement son mode de paiement. Or le but de cette technique et l'objet de nos précédentes recommandations, sont, précisément, de ne permettre que le paiement *on-chain*, donc en *stablecoin*. A défaut de pouvoir interdire ce paiement en *fiat*, les parties seront donc avisées d'user de la liberté contractuelle pour l'organiser, voire le désinciter.

Ainsi, dans leurs clauses, elles devront prévoir la procédure particulière que devra respecter le débiteur qui souhaite régler sa dette en monnaie *fiat*. Cette procédure pourra lui être rendue contraignante, c'est-à-dire moins simple que de passer par la *blockchain*. Nous estimons même que les parties pourront prévoir que la somme en *fiat* que devra acquitter le débiteur, s'il le souhaite, sera plus importante que celle qu'il devra en *stablecoin*. En effet, dans l'obligation facultative, le paiement en euros et le transfert de *stablecoin* sont deux obligations juridiques distinctes qui peuvent ne pas avoir un rapport d'équivalence. Le transfert d'« euros-*stablecoin* », qui seront considérés comme des biens, pourra donc être rendu moins élevé que le paiement d'euros ordinaires, qui sera considérée comme de l'argent, afin que le débiteur soit incité à ne transférer que des *stablecoin*.⁷⁴¹

Paiement.

M.A/le débiteur dispose du choix de payer la somme de 2000 euros par virement vers le compte

⁷³⁹ Marin, Gaetan. « Le bitcoin à l'épreuve de la monnaie », AJ Contrat, décembre 2017, 522.

Les parties prévoient alors une obligation facultative par laquelle le créancier promet d'accepter de recevoir en paiement une prestation différente de celle due. L'objet de l'obligation est déterminé ab initio, mais le créancier accepte que le débiteur se libère en exécutant une prestation facultative. Le créancier peut uniquement exiger l'exécution de la prestation in obligatione quand le débiteur a, pour sa part, le choix d'exécuter la prestation in obligatione ou bien celle in facultate solutionis. Les commerçants qui acceptent en paiement des bitcoins libellent le prix de leurs produits ou services en euro, mais acceptent que leur créance soit éteinte par le versement de bitcoins.

⁷⁴⁰ Par exemple le code monétaire et financier prévoit que les contrats dérivés doivent être réglés en espèces. Il peut en être déduit que sans cette possibilité, les contrats mettant en œuvre ces opérations ne peuvent plus être considérés comme des contrats dérivés selon l'article L211-1 du code monétaire et financier.

Article D211-1 A du code monétaire et financier : (...) 2. *Les contrats d'option, contrats à terme fermes, contrats d'échange, accords de taux futurs et tous autres contrats à terme relatifs à des matières premières qui doivent être réglés en espèces ou peuvent être réglés en espèces à la demande d'une des parties pour des raisons autres qu'une défaillance ou d'autre incident conduisant à la résiliation ;*

⁷⁴¹ Marjault, Yvan. « Les obligations disjonctives : étude des obligations alternatives et facultatives » §377. These de doctorat, Le Mans, 2016. <https://www.theses.fr/2016LEMA2001>.

La stipulation d'une obligation disjonctive dans l'engagement des parties implique que les parties ont entendu conférer un choix à l'une d'elles dans la détermination des prestations à exécuter par le débiteur. Les prestations objets de l'obligation disjonctive seront, d'une part, considérées comme équivalentes par les parties, et d'autre part, devraient être en adéquation avec la prestation qui devra être fournie par le créancier. Dès lors que cette modalité de l'obligation est prévue par les parties, elle doit être considérée comme normale et proportionnée.

bancaire de Mme B ou de Transférer 1500 Euros Numériques vers l'Adresse de Mme B. Il s'agit d'une obligation facultative, au sens des articles 1308 et suivants du code civil.

Il demeure plusieurs inconvénients à employer cette technique d'obligation facultative, surtout lorsque le prix *fiat* est majoré. D'abord, si le contrat est rédigé à l'initiative d'une seule des parties et/ou avec pour autre partie consommatrice⁷⁴², la clause pourrait courir le risque d'être qualifiée de clause abusive⁷⁴³ et/ou créant un déséquilibre significatif^{744 745}. Ensuite, il existe une incertitude sur la validité de ce schéma vis-à-vis de l'article L112-6 du code monétaire et financier : il permettrait de contourner totalement les limites de plafond de l'espace et de la monnaie électronique, alors que les parties utiliseraient le même actif.⁷⁴⁶

§ II - Clause d'imprévision

272. L'utilité de la clause de *hardship*. Une clause d'imprévision est une mesure contractuelle donnant les moyens aux parties de gérer les conséquences d'un événement imprévu bouleversant l'équilibre de leur contrat. Le but de ce mécanisme est de les *prémunir contre des modifications inattendues de leurs prévisions initiales, rendant l'exécution du contrat préjudiciable*

⁷⁴² Article lumineux du code de la consommation : 1° Consommateur : toute personne physique qui agit à des fins qui n'entrent pas dans le cadre de son activité commerciale, industrielle, artisanale, libérale ou agricole ; ...

⁷⁴³ Article L212-1 du code de la consommation : Dans les contrats conclus entre professionnels et consommateurs, sont abusives les clauses qui ont pour objet ou pour effet de créer, au détriment du consommateur, un déséquilibre significatif entre les droits et obligations des parties au contrat.

⁷⁴⁴ Article 1171 du code civil : Dans un contrat d'adhésion, toute clause non négociable, déterminée à l'avance par l'une des parties, qui crée un déséquilibre significatif entre les droits et obligations des parties au contrat est réputée non écrite. L'appréciation du déséquilibre significatif ne porte ni sur l'objet principal du contrat ni sur l'adéquation du prix à la prestation.

⁷⁴⁵ Marjault, Yvan. « Les obligations disjonctives : étude des obligations alternatives et facultatives ». §369 ; p.259 These de doctorat, Le Mans, 2016. <https://www.theses.fr/2016LEMA2001>.

Le choix en tant que droit potestatif pourrait créer un déséquilibre significatif entre ou dans les droits et obligations des parties au contrat.

⁷⁴⁶ Dominique Legeais. « Fasc. 535 : Actifs numériques et prestataires sur actifs numériques », JurisClasseur Commercial, 14 octobre 2019.

Pour que le paiement ait valeur libératoire, il faut alors que le créancier l'accepte. À supposer même que le crypto-actif ne soit pas considéré comme une monnaie, il y a possibilité de qualifier l'opération de dation en paiement au sens de l'article 1342-4 du Code civil ou d'échange de biens. Des obstacles ont cependant été soulevés à cette analyse. L'usage d'une monnaie alternative ne devrait pas permettre de contourner les règles d'ordre public encadrant la réalisation des paiements et notamment les interdictions de paiement en espèces de certaines créances posées par l'article L. 112-6 du Code monétaire et financier.

ou à tout le moins, d'un intérêt économique moindre⁷⁴⁷.

Depuis la réforme du droit des obligations⁷⁴⁸, c'est l'article 1195 du Code civil qui donne un cadre à ce type de mécanisme. Il prévoit qu'à défaut de stipulations contraires, la partie qui fera face à un changement de circonstances imprévisibles rendant l'exécution du contrat excessivement onéreuse pour elle pourra demander une renégociation du contrat, qui pourra aboutir à son éventuelle révision par le juge en cas d'échec des négociations, voire à une résolution⁷⁴⁹.

Les parties peuvent toutefois instituer leur propre clause de *hardship*, dérogeant à cette procédure supplétive de volonté⁷⁵⁰. Il leur est proposé d'en créer une spécialement dédiée aux événements concernant leur smart contracts et à la *blockchain* (A), qui leur permettra de modifier leur accord écrit afin de faire primer leur volonté écrite sur les possibles imprévus de la technologie qu'elles mobiliseront (B).

A – L'évènement imprévisible déclencheur

273. Spécificité de la clause d'imprévision dans le contexte d'une exécution par smart contract. Une clause d'imprévision spécialement dédiée aux smart contracts et à la *blockchain* permettra de mettre en œuvre le mécanisme cœur de notre approche donnant primauté à l'accord écrit en langage naturel⁷⁵¹. En effet, en cas d'incidents relatifs à l'ensemble technologique dédié à l'exécution du contrat, les parties disposeront dans la clause de *hardship* d'un moyen de régler les conséquences néfastes de ceux-ci en faisant primer leur volonté écrite sur les événements déroulés

⁷⁴⁷ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires – Chapitre 57 Clause d'imprévision - §906. Les Intégrales. LGDJ, 2018.

⁷⁴⁸ Ordonnance n° 2016-131 du 10 février 2016 portant réforme du droit des contrats, du régime général et de la preuve des obligations.

⁷⁴⁹ Article 1195 du code civil : *Si un changement de circonstances imprévisible lors de la conclusion du contrat rend l'exécution excessivement onéreuse pour une partie qui n'avait pas accepté d'en assumer le risque, celle-ci peut demander une renégociation du contrat à son cocontractant(...) En cas de refus ou d'échec de la renégociation, les parties peuvent convenir de la résolution du contrat.*

⁷⁵⁰ Dimitri Houtcieff. « Les dispositions de l'article 1195 sont-elles supplétives ? », Gazette du Palais, n° 16 (10 mai 2022): 7.

Afin de ne pas renoncer à la consécration d'un dispositif passant pour résolument moderne, le gouvernement communiqua volontiers sur le caractère supplétif de ce mécanisme nouveau. Ainsi la garde des Sceaux tenta-t-elle de calmer les inquiétudes exprimées en faisant valoir que « la révision judiciaire pour imprévision ne s'applique que si les parties n'en ont pas convenu autrement » (N. Belloubet, Compte rendu analytique de la séance du 1er février 2018, Sénat).

⁷⁵¹ V., *infra*, §215

dans la *blockchain*⁷⁵².

Imaginons un *smart contract* automatisant une clause de préférence de titres financiers, lesquels sont matérialisés par des jetons. Le programme est piraté par un tiers, qui arrive à transférer à son adresse les titres tokenisés. Dans ce contrat intelligent de préférence, le promettant et le bénéficiaire de la clause auront prévu qu'un tel évènement causera un déclenchement de la clause d'imprévision : les parties pourront alors procéder à la procédure de « réparation » de ces incidents ; c'est-à-dire la suppression des anciens titres-jetons, la création de nouveaux smart contracts pour émettre de nouveaux titres-jetons, etc.

La différence donc entre la clause d'imprévision proposée et une autre plus classique réside essentiellement dans l'évènement déclencheur de la procédure de renégociation du contrat. Nous ne visons pas exclusivement des évènements rendant excessivement onéreuse l'exécution d'une obligation pour les parties, mais toute défaillance ou incident qui fera dévier l'exécution du contrat de ce que les parties avaient initialement voulu. En ce sens, la clause ressemble à celle de *material adverse change* qui figure dans la plupart des contrats de fusion-acquisition⁷⁵³. Il s'agit de définir les évènements qui permettront aux parties de gérer, *a posteriori*, le risque de défaillance lié à la *blockchain* et/ou des smart contracts.⁷⁵⁴

274. Liste des évènements déclencheurs. Les parties devront donc dresser une liste, dans leur clause d'imprévision, des évènements susceptibles de déclencher la procédure de neutralisation

⁷⁵² Mekki, Mustapha. « Les mystères de la blockchain ». Recueil Dalloz, n° 37 (2 novembre 2017): 2160.

La mise en oeuvre quasi-automatique d'un smart contract peut également causer quelques dommages qui peuvent rendre nécessaire l'introduction d'une clause de hardship (...) pour régler les conséquences excessives de cette automaticité.

⁷⁵³ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires - Chapitre 65 Clause MAC (material adverse change). Les Intégrales. LGDJ, 2018.

La clause dite « MAC » tire sa dénomination de l'expression anglaise « material adverse change » (on parle aussi de « material adverse effect »), qui signifie « changement/effet significatif défavorable » (...). Cette clause a vocation à protéger l'un des contractants des conséquences défavorables d'une modification importante des circonstances avant la conclusion définitive du contrat envisagé (pendant la période de réalisation des conditions suspensives) ou en cours d'exécution.

⁷⁵⁴ Dans la forme *code and law* proposée par Gabriel Shapiro, ce dernier fait usage de ce terme pour définir les situations où les parties ne seront plus liées par le *smart contract* :

g4brielShapir0. « SCoDA – Simple Code Deference Agreement », 25 avril 2023. <https://github.com/lex-node/SCoDA-Simple-Code-Deference-Agreement->.

The exceptional circumstances under which the parties are not required to defer to the smart contract are captured under the concept of a “Material Adverse Exception Event.” These circumstances are basically the blockchain equivalent of ‘force majeure’ events—e.g., a 51% attack that causes a double-spend somehow affecting the smart contract.

des conséquences. Ce faisant, elles devront trouver un équilibre entre :

- la préservation de l'opportunité de recourir à un *smart contract* ; elles doivent se garder de prévoir trop d'évènements pouvant interrompre le fonctionnement du contrat intelligent par l'effet de cette clause ; sinon l'opportunité de recourir aux smart contracts s'en trouvera diminué⁷⁵⁵;
- l'effectivité du principe de primauté du document *fiat* de la forme ricardienne ; qui doit permettre aux parties de modifier le contrat lorsque la technologie utilisée pour l'exécuter défaille et contrevient à leur volonté matérialisée dans l'accord écrit en langage naturel⁷⁵⁶.

Il sera donc recommandé aux parties de commencer par donner une définition générale d'un évènement pouvant activer la procédure d'amendement du contrat.

Les parties conviennent de modifier le contrat, tel que prévu dans la procédure de révision de cette clause, en cas de survenance d'un événement imprévisible, relatif à la Blockchain et aux smart contractss, rendant excessivement difficile ou impossible l'exécution normale et initialement voulue du présent contrat.

Puis, nous recommandons aux parties de constituer une liste précise mais non exhaustive de tels évènements⁷⁵⁷ afin de déclencher quasi-automatiquement la procédure de l'imprévision lorsqu'ils se présenteront. Nous estimons que ceux-là peuvent être :

- une défaillance des smart contracts qu'auront développés ou fait développer des parties due à un bogue ou un piratage. Comme évoqué, les parties peuvent faire le choix de définir négativement ces notions en stipulant que ce sont des comportements (provoquées en cas de piratage, ou non, en cas de bogue), inconformes des smart contracts aux spécifications fonctionnelles, comme il est courant de le faire en informatique rédigé à l'avantage du client

⁷⁵⁵ Il s'agit notamment du reproche que Gabriel Shapiro faisait à la forme de contrat donnant toute primauté à l'accord écrit en langage naturel. V., *infra*, §202

⁷⁵⁶ V., *infra*, §215

⁷⁵⁷ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires - Chapitre 57 Clause d'imprévision. Les Intégrales. LGDJ, 2018.

(...) La troisième forme hybride [de rédaction d'une clause d'imprévision] consiste à énoncer une définition générique de l'évènement [imprévisible] susceptible de faire jouer la clause, puis d'énumérer des cas précis. Elle réunit les avantages des deux premières formes, en incluant de manière indiscutable ou presque certains évènements et en laissant la porte ouverte à l'inclusion d'autres évènements particulièrement imprévisibles.

⁷⁵⁸ ;

- une défaillance des smart contracts avec lesquels interagissent des parties. Les parties peuvent être amenées à interagir avec des smart contracts qui ne sont pas les leurs : des *stablecoin*, des oracles, des protocoles de finance décentralisée. En cas de défaillance de ceux-ci, elles peuvent vouloir modifier leur contrat ;
- une hausse prolongée et importante des frais de d'interaction avec une *blockchain*⁷⁵⁹. Dans cette hypothèse, l'interaction avec les smart contracts devient bien plus onéreuse que prévue et peut pousser les parties à vouloir changer de *blockchain* et/ou modifier leurs obligations⁷⁶⁰ ;
- la compromission du *wallet*. Le *wallet* est le dispositif contenant l'identité d'un individu dans la *blockchain* et lui permettant d'interagir avec elle⁷⁶¹. Les parties peuvent prévoir que la compromission de celui d'une des parties les obligera à modifier le contrat ne serait-ce que pour refléter la nouvelle l'adresse du nouveau *wallet* ;
- une défaillance de la *blockchain*. L'infrastructure sur laquelle réside les smart contracts peut elle aussi défaillir pour différentes raisons : une attaque 51%⁷⁶², une congestion prolongée, un *fork* contentieux... Cela peut pousser les parties à vouloir en changer et/ou réadapter le *smart*

⁷⁵⁸ Pierre SIRINELLI et Michel VIVANT. Formulaires ProActa Droit de l'immatériel - II.330-35 Article 1. Définitions. LAMY, 2020.

Anomalie. (...) Même en convenant mutuellement de s'appuyer sur une documentation objective, le prestataire souhaitera souvent prendre comme référence une documentation minimaliste (la documentation officielle associée au logiciel par exemple) alors que le client préférera faire référence aux Spécifications, pouvant inclure d'autres éléments tels que des spécifications générales et détaillées de la solution logicielle.

⁷⁵⁹ V., *supra*, §335

⁷⁶⁰ Cette hypothèse est celle qui se rapproche le plus du cas d'usage originel du mécanisme de révision pour imprévision : un événement qui rend excessivement chère la poursuite de la réalisation des obligations pour une partie.

Arrêt Canal de Craponne, Cass civ, 6 mars 1876, D. 1876, 1, p. 193

Article 1195 du code civil : *Si un changement de circonstances imprévisible lors de la conclusion du contrat rend l'exécution excessivement onéreuse pour une partie qui n'avait pas accepté d'en assumer le risque, celle-ci peut demander une renégociation du contrat à son cocontractant. Elle continue à exécuter ses obligations durant la renégociation (...).*

⁷⁶¹ V., *supra*, §407

⁷⁶² Dans sa proposition de modèle, Gabriel Shapiro parle d'attaque consensus qu'il définit comme tel :

_g4brielShapiro. « SCoDA – Simple Code Deference Agreement », 25 avril 2023. <https://github.com/lex-node/SCoDA-Simple-Code-Deference-Agreement>.

(...) (d) "Consensus Attack" means an attack that: (i) is undertaken by or on behalf of a block producer who controls, or group of cooperating block producers who collectively control, a preponderance of the means of block production on the Designated Blockchain Network; and (ii) has the actual or intended effect of: (A) reversing any transaction made to or by the Designated Smart contract after Confirmation of such transaction, including any "double spend" attack having or intended to have such effect; or (B) preventing inclusion in blocks or Confirmation of any transaction made to or by the Designated Smart contract, including any "censorship attack," "transaction withholding attack" or "block withholding attack" having or intended to have such effect.

contract.

B – La révision du contrat

275. Procédure de renégociation. À la survenance d'un événement imprévisible, les parties pourront démarrer la procédure de révision du contrat, qui pourra emprunter aux procédures classiques d'imprévision⁷⁶³ :

- notification par lettre recommandée, dans un certain délai, par l'une des parties, afin d'informer l'autre de la survenance d'un événement imprévisible,
- négociation de bonne foi du contrat dans un autre délai imparti, celui-ci devra être assez long pour que les parties puissent avoir le temps d'amender le contrat et de modifier ou créer un nouveau *smart contract* éventuel. Le contrat ainsi que le *smart contract* le cas échéant seront suspendus,
- intervention d'un tiers, arbitre ou médiateur, dans le cas où il y a une dispute sur la survenance de l'événement imprévisible et/ou si les négociations n'ont pas abouti au terme du délai imparti⁷⁶⁴.

276. Terme de la renégociation. A l'issue du processus de renégociation et/ou révision, trois options s'offriront aux parties :

- elles pourront choisir de résilier leur contrat et ainsi détruire les *smart contracts* les exécutant⁷⁶⁵,
- elles pourront choisir, au contraire, de poursuivre leur contrat mais sans recourir à un *smart*

⁷⁶³ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires – Clause d'imprévision 815 - Procédure d'activation de la clause. Les Intégrales. LGDJ, 2018.

Il est souhaitable que les contractants envisagent les modalités de déclenchement de la clause. Principalement, les parties doivent prévoir les modalités de la notification de la survenance des circonstances mettant la clause en mouvement. Doivent figurer dans la clause un délai, une forme (un écrit par exemple), et un contenu (description de l'événement survenu, voire une première proposition de réadaptation) (...).

⁷⁶⁴ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires – Clause d'imprévision 815 - Procédure d'activation de la clause. Les Intégrales. LGDJ, 2018.

Enfin et surtout, il est éminemment souhaitable que les parties prévoient les conséquences liées à une contestation par le cocontractant de la réunion des conditions d'application de la clause. (...) Toutefois le risque d'abus de partie à qui est dévolu ce pouvoir unilatéral [d'appréciation de la réalisation de l'hypothèse visée par la clause] conduit à préférer des clauses organisant une procédure bilatérale.

⁷⁶⁵ V., *supra*, §449

contract. Par exemple, si la *blockchain* sur laquelle est exécuté un contrat intelligent de location d'un bien immobilier est défaillante, les parties peuvent décider de poursuivre leur contrat de location de façon ordinaire, sans automatisation par des smart contracts,

- elles pourront enfin choisir de poursuivre leur contrat tout en apportant des modifications aux logiciels qu'elles auront utilisés : la *blockchain* et le *smart contract* notamment. Si l'on poursuit avec le dernier exemple, les parties pourront décider de déployer de nouveaux smart contracts sur une *blockchain* opérationnelle.

Si une Partie estime reconnaître un Evènement imprévisible tel que défini à l'article ..., elle pourra avertir l'autre par lettre recommandée avec accusé de réception pour lui proposer de modifier le contrat, dans un délai maximal de Dans le cas où les parties ne s'accordent pas sur la qualification d'Evènement imprévisible, elles s'engagent à soumettre l'appréciation de cette qualification à un tiers désigné| à l'arbitre désigné dans la clause compromissoire.

Les Parties devront ensuite renégocier de bonne foi le contrat afin de le modifier, éventuellement avec les smart contracts, afin de remédier aux conséquences de l'Evènement imprévisible sur leur relation. Cette étape ne devra pas dépasser ...jours. Pendant toute la durée de la révision, les effets du contrat seront suspendues et le Smart contract sera pausé dans les conditions prévues aux articles...

En cas d'échec de la révision, les Parties conviennent qu'il ne sera pas possible de saisir le juge d'une demande de révision du contrat. Les Parties devront saisir leur Arbitre conformément à la procédure décrite de clause compromissoire du présent contrat.

§ III - Clause relative à la responsabilité

277. Limitation de responsabilité. Les clauses relatives à la responsabilité dans les contrats visent le plus souvent à limiter ou exclure, dans le cadre de ce qui est légalement possible, le montant des dommages et intérêts dus par une des parties en cas d'inexécution contractuelle. Dans des cas plus rares, elles ont vocation à écarter des hypothèses d'engagement de la responsabilité du débiteur⁷⁶⁶. Dans notre contexte, les parties peuvent trouver pertinent d'insérer dans l'accord écrit en

⁷⁶⁶ Nathalie BLANC, Romain BOFFA, et Denis MAZEAUD. Dictionnaire du contrat §43 – p.184 . LGDJ, 2018.

Les clauses de responsabilité peuvent, en premier lieu, porter sur les conditions de la mise en jeu de la responsabilité de l'auteur d'un dommage. (...) Ensuite, et dans la même perspective, ils peuvent exclure l'existence de toute responsabilité alors que toutes les conditions de celle-ci sont réunies (clause évasive de responsabilité).

langage naturel de leur contrat intelligent une clause visant spécifiquement à organiser leur responsabilité en cas de dommages provoqués par leurs smart contracts⁷⁶⁷.

Celle-ci peut toutefois déjà figurer dans le contrat de développement de *smart contract* qu'auront éventuellement conclu les parties, préalablement à leur contrat intelligent. En effet, si l'une d'entre elle a conçu, au titre d'une prestation de service, les smart contracts qui seront utilisés pour exécuter le contrat intelligent, la question de la réparation des dommages due à sa mauvaise conception⁷⁶⁸ a pu déjà être réglée dans ce contrat de développement. Les parties pourront alors faire un renvoi vers cette clause dans leur contrat intelligent.

Responsabilité : En cas de dommages causés par les smart contracts, le régime de responsabilité est celui stipulé à l'article ... du contrat de développement figurant en annexe du présent contrat.

Dans les cas où il n'existe pas de telles précédentes relations, les parties seront contraintes d'organiser cette responsabilité dans le contrat exécuté par le *smart contract*. Elles définiront les faits générateurs de responsabilité (A) puis le régime (B).

A - Faits générateurs de responsabilités

278. Faute de conception du *smart contract*. Un *smart contract* peut provoquer un dommage s'il est bogué ou piraté : par exemple, des sommes tenues en séquestre par le programme sont bloquées ou volées par un pirate en raison de la mauvaise écriture d'une ligne de code ; le créancier de ces sommes subit le préjudice financier de leur perte⁷⁶⁹. Afin que ces deux événements

⁷⁶⁷ Jeffrey D. Neuburger, Wai L. Choy, et Kevin P. Milewski. « Smart contracts: Best Practices », Thomson Reuters - Practical Law, 2019, 11-19.

In addition to typical indemnity clauses, the parties should include in the governing traditional contract provisions to address smart contracts specifically.

⁷⁶⁸ A l'instar des renvois que l'on trouve dans les conditions générales de contrat d'interchange. Par exemple, cette clause de responsabilité dans un contrat EDI qui fait un renvoi à un contrat commercial en cas de dommage lié aux opérations de l'ensemble logiciel

https://www.hella.com/hella-com/assets/media_global/EDI_Agreement_Hella_Signature.pdf

Article 11: Liability

Responsibility for damages related to the operation of EDI, caused by a party itself or any third party acting on behalf of the party, shall be specified in a commercial agreement and is not subject to this agreement otherwise the legal liability shall apply.

⁷⁶⁹ Jeffrey D. Neuburger, Wai L. Choy, et Kevin P. Milewski. « Smart contracts: Best Practices », Thomson Reuters - Practical Law, 2019, p. 11-19.

In addition to typical indemnity clauses, the parties should include in the governing traditional contract provisions to address smart contracts specifically. For example, indemnities protecting the parties that did not code the smart contract with respect to intellectual property infringement by the smart contract and damages resulting from improper operation

engendrent la responsabilité contractuelle d'une partie, il faut qu'ils découlent d'une faute de conception du *smart contract*. Elle-même ne peut être caractérisée que si une des parties s'est préalablement engagée à une obligation de bonne conception du *smart contract*, qui n'aura donc pas été respectée⁷⁷⁰. Il sera difficilement envisageable cependant d'obtenir l'engagement de la responsabilité du concepteur d'un *smart contract* pour le piratage de celui-ci, puisqu'il s'agit d'un fait sur lequel il n'a que très peu de contrôle. Les parties pourront tout de même estimer qu'une faute sera caractérisée si le piratage a été causé par un défaut de conception du *smart contract* particulièrement grave.

Un Bogue ou un Piratage découlant d'un défaut de conception particulièrement grave constitue une faute de X engageant sa responsabilité.

En revanche dès lors que les parties utiliseront un *smart contract* qu'elles n'auront ni développé, ni fait développer, alors sa défaillance ne sera pas forcément génératrice de responsabilité ; à moins que l'une d'entre elles se soit portée garante de son bon fonctionnement et sa sécurité⁷⁷¹. Par exemple, une partie A peut utiliser un *smart contract open source* pour exécuter un contrat d'option avec sa co-contractante. À cette occasion, A peut s'engager sur le bon fonctionnement et la sécurité de ce *smart contract*. Nous pensons toutefois qu'il est peu probable, en pratique, qu'une partie obtienne de l'autre un tel engagement sur un programme qu'elle n'a pas conçu. Il sera plus plausible qu'elle formule des garanties de moindre intensité, comme celles de ne pas avoir laissé passer certains bogues connus dans le programme.

A s'engage à ce qu'il n'y ait aucun Bogue de réentrance dans le Smart contract tel que défini à

of the smart contract or from other errors in the smart contract.

⁷⁷⁰ Le concepteur d'un logiciel spécifique est soumis selon une jurisprudence constante à une obligation de résultat de délivrance conforme.

Legalis | L'actualité du droit des nouvelles technologies. « Logiciel spécifique : manquement à l'obligation de résultat mais pas de résolution du contrat », 2 février 2021. <https://www.legalis.net/actualite/logiciel-specifique-manquement-a-lobligation-de-resultat-mais-pas-de-resolution-du-contrat/>.

Le tribunal a rappelé qu'en matière de logiciels spécifiques développés pour les besoins d'un utilisateur, le prestataire est tenu de délivrer un produit conforme aux spécifications détaillées dans le cahier des charges et que, par conséquent, il est soumis à une obligation de résultat à l'égard de son client.

⁷⁷¹ Ainsi si les parties récupèrent un programme en licence libre sur internet, il est probable que cette licence prévoit une clause particulièrement évasive de responsabilité.

Open Source Initiative. « The MIT License », 31 octobre 2006. <https://opensource.org/license/mit/>.

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

l'article « Définitions ». En cas de survenance de celui-ci, il engage sa responsabilité conformément à l'article....

279. Garantie d'éviction. Les dommages liés à un *smart contract* peuvent également être issus de l'irrespect, par la partie conceptrice, des droits des tiers⁷⁷². Imaginons une partie A mettant à disposition de B un *smart contract* utilisé pour exécuter le contrat intelligent conclu entre ces deux. Le programme fourni par A est un plagiat d'un autre *smart contract* créé par C, qui l'a protégé par une licence sur ses droits d'auteur. C peut demander l'arrêt de l'utilisation de ce *smart contract* à B ; ce qui lui portera préjudice et pourra le pousser à vouloir engager la responsabilité de B. Comme dans tout contrat de développement logiciel⁷⁷³, les parties seront donc avisées de prévoir une garantie d'éviction pesant sur la partie fournisseuse, dont la violation entraînera l'engagement de sa responsabilité.

M. A garantit que le Smart contract n'enfreint aucun droit de tiers et respecte en particulier leurs droits de propriété intellectuelle. En cas de violation de ces droits, M. A engage sa responsabilité conformément à l'article... de ce contrat.

280. Écarts d'hypothèses de responsabilité. Les parties seront en revanche avisées d'expressément écarter certaines hypothèses d'engagement de responsabilité. Ainsi, elles veilleront à ce que des événements sur lesquelles elles n'ont aucun contrôle ou qui relèvent de leur responsabilité strictement individuelle ne puissent pas constituer une faute. Ce pourra être le cas de la compromission du *wallet* d'une des parties, la défaillance générale de la *blockchain*, un piratage très sophistiqué de leurs *smart contracts*, un dysfonctionnement d'un *smart contract* qui n'a pas été

⁷⁷² Il s'agit d'une attaque pirate très connu des *smart contracts*.

Kamil Polak. « Hack Solidity: Reentrancy Attack | HackerNoon », 17 janvier 2022. <https://hackernoon.com/hack-solidity-reentrancy-attack>.

The Reentrancy attack is one of the most destructive attacks in the Solidity smart contract. A reentrancy attack occurs when a function makes an external call to another untrusted contract. Then the untrusted contract makes a recursive call back to the original function in an attempt to drain funds.

⁷⁷³ Vivant Michel. « Lamy droit du numérique » - Partie 3 Numérique et contrats - Division 3 Les principaux contrats du numérique et leurs spécificités - Chapitre 3 Le contrat de développement de logiciel spécifique - Section 3 Quelques clauses caractéristiques. Editions Lamy, 2012.

Bien entendu, parmi les garanties dues au client par le prestataire, se trouve également la garantie d'éviction, dont la portée est d'autant plus importante que l'est l'étendue de la cession des droits intellectuels (...). Cette clause d'éviction permettant au cessionnaire d'appeler en garantie le cédant dans le cas où la propriété intellectuelle du logiciel cédé lui serait contestée par un tiers prend encore plus d'importance dans le contexte contemporain du développement des logiciels dits « libres » régis par des licences « open source ».

développé par l'une des parties, un piratage du site web faisant l'interface avec le *smart contract*⁷⁷⁴, constitueront autant de faits devant être spécifiés comme insusceptibles d'engager la responsabilité de l'une quelconque des parties.

B – Indemnisation

281. Réparation des dommages. Après avoir établi les faits générateurs de responsabilités, les parties pourront prévoir dans leur clause la mesure des dommages et intérêts qu'elles seront susceptibles de verser à l'une ou l'autre d'entre elles. Le but de la clause limitative de responsabilité est de réduire le plus possible cette assiette de réparation des dommages⁷⁷⁵. Les parties seront donc avisées de commencer par rappeler que la loi limite la réparation aux dommages personnels, certains, directs et prévisibles⁷⁷⁶. Ensuite elles pourront plafonner le maximum de l'indemnité de réparation.

282. Plafonnement des dommages. Il est d'usage de la cantonner à la valeur totale du contrat, soit le maximum des sommes en jeu dans la relation⁷⁷⁷. Or, comme l'architecture des smart contracts fonctionne en séquestre, les sommes objets des transferts seront en principe entièrement provisionnées dans le programme⁷⁷⁸ ; ce qui facilitera l'identification du montant total en jeu que le plafond d'indemnisation ne pourra dépasser.

Le plafond maximal d'indemnisation ne pourra pas dépasser le total des Euros Numériques déposés

⁷⁷⁴ V., *supra*, §403

⁷⁷⁵ Jean-Luc JUHAN. « Pratique du droit du contrat : les clauses limitatives de responsabilité », Revue Le Lamy Droit civil, n° 99 (1 décembre 2012).

Le premier commandement en vue d'aboutir à une rédaction qui soit efficace concerne l'attention portée aux exclusions de dommages. Sur ce point, on a tendance à se focaliser sur le montant du plafond inséré dans ces contrats. Il convient naturellement d'observer également – en général c'est inséré dans la même clause, et de toute façon c'est le même sujet – quels sont les dommages qui sont exclus de la responsabilité de l'une ou l'autre des parties.

⁷⁷⁶ Article 1231-3 du code civil : *Le débiteur n'est tenu que des dommages et intérêts qui ont été prévus ou qui pouvaient être prévus lors de la conclusion du contrat, sauf lorsque l'inexécution est due à une faute lourde ou dolosive.*

⁷⁷⁷ Jean-Luc JUHAN. « Pratique du droit du contrat : les clauses limitatives de responsabilité », Revue Le Lamy Droit civil, n° 99 (1 décembre 2012).

Il s'agit de choisir un montant de plafond de responsabilité qui soit cohérent et justifiable lors de la négociation, mais également justifiable devant une juridiction si nécessaire. La référence au prix du contrat est très utilisée – c'était d'ailleurs le cas dans l'affaire Oracle-Faurecia, où le montant de la clause était équivalent à celui du contrat. C'est une bonne référence dans la mesure où il s'agit d'une référence économique, au prix, et non une référence au dommage potentiel, qui s'avère plus hasardeuse.

⁷⁷⁸ V., *infra*, §283

dans le Smart contract.

283. Conformité des clauses limitatives de responsabilité à l'ordre public. Bien évidemment, les parties devront prendre garde à préserver la validité de leur clause en veillant à ce qu'elle ne vide pas le contrat de son obligation essentielle⁷⁷⁹. Pour le concepteur du *smart contract*, il s'agira en plus de ne pas créer pas un déséquilibre significatif entre les droits et les obligations des parties⁷⁸⁰. De plus, ces clauses ne sauraient couvrir le dol ou la faute lourde⁷⁸¹. En toutes circonstances, et en sus de toutes les mesures de sécurité prises contre le risque de défaillances d'un *smart contract*, les parties pourront également stipuler une clause indiquant la souscription d'une assurance qui pourra couvrir le montant des dommages.⁷⁸²

Enfin, les parties ne pourront pas écarter l'application de régimes de responsabilités légaux. Et pour préserver la validité de leur clause limitative de responsabilité, elles seront avisées de préciser dans leur clauses que leurs stipulations ne s'appliquent pas pour certains dommages sur lesquels la loi institue un régime de responsabilité impératif. Ce sera notamment le cas des violations de données personnelles, qui peuvent figurer parmi les dommages causés par un *smart contract*. Or l'article 82 du RGPD n'est pas supplétif de volonté⁷⁸³.

284. Conclusion de la section I. Les parties seront donc avisées dans la rédaction de leurs clauses relatives à leurs obligations de distinguer les obligations juridiques qui les incombent et leur mise en œuvre à travers les smart contracts. Pour parer aux comportements inattendus de la *blockchain* et des smart contracts, elles pourront rédiger une clause de *hardship* qui leur permettra de modifier leur contrat intelligent en cas de survenance d'un événement contraire à leur volonté écrite. Enfin, dans le cas où ces événements seraient causés par la faute d'une des parties, une clause pourra

⁷⁷⁹ Article 1170 du code civil : Toute clause qui prive de sa substance l'obligation essentielle du débiteur est réputée non écrite.

⁷⁸⁰ Article 1171 du code civil : Dans un contrat d'adhésion, toute clause non négociable, déterminée à l'avance par l'une des parties, qui crée un déséquilibre significatif entre les droits et obligations des parties au contrat est réputée non écrite.

⁷⁸¹ Cour de Cassation, Chambre commerciale, du 3 avril 2001, 98-21.233, Publié au bulletin.

⁷⁸² Jeffrey D. Neuburger, Wai L. Choy, et Kevin P. Milewski. « Smart contracts: Best Practices », Thomson Reuters - Practical Law, 2019, 11-19.

Although it remains to be seen how insurers will cover liabilities resulting specifically from smart contract failures, such as faulty oracle performance or coding mistakes, insurance will likely play a role in assessing the use of smart contracts.

⁷⁸³ Article 82 du RGPD : 1. Toute personne ayant subi un dommage matériel ou moral du fait d'une violation du présent règlement a le droit d'obtenir du responsable du traitement ou du sous-traitant réparation du préjudice subi (...).

être conçue afin d'organiser leur responsabilité.

Section II – Les clauses intéressant les tiers au contrat

285. Tierce-parties. Notre attention portera sur deux types de tiers qui peuvent se retrouver mêlés à l'exécution d'un contrat intelligent, sans remettre en question l'effet relatif du contrat. Ceux qui seront chargés d'approvisionner en information le *smart contract* exécutant l'accord, qu'on appelle les oracles (§1), et ceux dont les données personnelles pourront être traitées par les smart contracts (§2). L'interaction de ces personnes avec les parties du contrat intelligent nécessitera la rédaction de clauses spécifiques.

§ I – Clause relative aux oracles

286. Définition d'un oracle. Un contrat intelligent nécessite presque systématiquement le recours à un oracle : soit un individu et/ou une solution approvisionnant un *smart contract* d'une information provenant du monde *off-chain*⁷⁸⁴. Lorsque l'oracle est une personne ou sous la responsabilité d'une personne identifiée, cette dernière peut intervenir au contrat en souscrivant des engagements vis-à-vis des parties. Dans un document, annexe ou inclus dans l'accord écrit en langage naturel, sera donc formalisé les obligations de la personne-oracle ou responsable de l'oracle, et les conséquences de leur irrespect⁷⁸⁵ ; cela confèrera une sécurité additionnelle aux mesures techniques mises en place pour assurer son bon fonctionnement⁷⁸⁶. Le régime de responsabilité qui y sera stipulé (B) dépendra du type d'oracle que les parties utiliseront (A).

A – Typologie d'oracles

⁷⁸⁴ « What Is an Oracle in Blockchain? » Explained | Chainlink ». Consulté le 29 mai 2023.
<https://chain.link/education/blockchain-oracles>.

Solving the oracle problem is of the utmost importance because the vast majority of smart contract use cases like DeFi require knowledge of real-world data and events happening off-chain. Thus, oracles expand the types of digital agreements that blockchains can support by offering a universal gateway to off-chain resources while still upholding the valuable security properties of blockchains.

⁷⁸⁵ Mustapha Mekki. « Blockchain : l'exemple des smart contracts Entre innovation et précaution » p.14 , 2018.
<https://lesconferences.openum.ca/files/sites/97/2018/05/Smart-contracts.pdf>.

Le talon d'Achille de la blockchain semble être cet Oracle. Que se passe-t-il si l'Oracle ne fournit pas l'information empêchant le smart contract de se déclencher ? Que fait-on si l'information communiquée est erronée ou piratée ? Quelle responsabilité pour cet Oracle, lui applique-t-on le droit commun ou doit-on penser à la création d'un droit spécial ? En attendant, son intervention doit être minutieusement encadrée par un ensemble de stipulations contractuelles ?

⁷⁸⁶ V., *supra*, §454

287. Humain ou machine. L'oracle fournissant l'information à un *smart contract* peut-être de deux natures⁷⁸⁷ : il peut s'agir d'un programme (a) ou d'un humain (b).

a) Oracle-programme

288. Oracle et source de données. Lorsqu'il s'agit d'un programme, un oracle est le logiciel acheminant une donnée provenant de l'extérieur de la *blockchain* vers un *smart contract*. Il se distingue ainsi du programme constituant la source d'information, qui se contente de mettre à disposition une donnée quérable. La caractéristique fondamentale d'un oracle est de récupérer cette donnée, et non pas de la créer, pour l'approvisionner ensuite à un *smart contract*⁷⁸⁸. Ce n'est qu'accessoirement qu'il peut aussi en être la source.

Par exemple, dans une clause intelligente d'assurance paramétrique, l'information déclenchant l'indemnisation peut être une donnée météorologique⁷⁸⁹. Cette dernière peut provenir de l'API d'une station météorologique, que différents programmes peuvent librement solliciter. Bien que cette API soit la source de l'information, elle n'est pas considérée comme un oracle ; c'est le programme en charge de récupérer la donnée météorologique et la délivrer au *smart contract* qui le sera. Toutefois dans d'autres cas, le programme fournissant l'information peut aussi être celui approvisionnant le *smart contract*. Par exemple, dans une lettre de crédit exécutée *on-chain*, le transfert des fonds peut être déclenché au moment de l'envoi de la marchandise par bateau. L'entreprise de fret ou un objet connecté peut alors être chargé de diffuser et fournir cette information directement au *smart*

⁷⁸⁷ Egberts, Alexander. « The Oracle Problem - An Analysis of How Blockchain Oracles Undermine the Advantages of Decentralized Ledger Systems ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 12 décembre 2017. <https://papers.ssrn.com/abstract=3382343>.

The way the Oracle retrieves its data depends on whether it relies on human involvement or functions completely automated.

⁷⁸⁸ Ibid.

“Automated Oracles” provide data in a three-step-process. First, the Smart contract sends an inquiry to the Oracle. The Oracle then accesses a “data source” and retrieves the required data. Finally, the Oracle sends the data to the Smart contract. Hence, it is important to differentiate between the Oracle and the data source.

⁷⁸⁹ Chaperon, Meg. « Parametric Flood Insurance ». Descartes. Consulté le 4 avril 2023. <https://www.descartesunderwriting.com/whitepaper/parametric-flood-paper/>.

Descartes monitors the exposure with our new technologies, and when pre-agreed thresholds, such as river water levels, amount of rainfall, and wind speed are reached or exceeded, it directly leads to pay-out.

contract⁷⁹⁰ afin qu'il relâche les fonds séquestrés à l'acheteur⁷⁹¹.

289. Contenu de la clause d'oracle. Dans l'hypothèse où l'oracle, à la fois, créé l'information et l'approvisionne au *smart contract*, la relation entre la personne responsable de ce dernier et une ou l'ensemble des parties pourra être encadrée par des stipulations semblables à celles trouvées dans les conventions de niveaux de service⁷⁹². Elle s'engagera sur la disponibilité de la donnée, sa vitesse de transfert, son délai maximal d'interruption, son délai de restauration, la sécurité de son système, etc⁷⁹³.

Concernant la tâche d'acheminement de la donnée, qui est donc le véritable travail d'un oracle, les parties veilleront à engager, en sus, l'entité responsable sur ces engagements précis :

- récupérer l'information d'une source précise,
- garantir son intégrité pendant l'acheminement de l'information,
- livrer la donnée dans un délai et une fréquence déterminées,
- fournir la donnée à l'adresse précise du *smart contract* dans un format déterminé.

La garantie d'intégrité pourra être accompagnée d'une obligation générale de sécurité du processus. Nous savons que les oracles figurent parmi les vecteurs principaux d'attaques des smart contracts⁷⁹⁴, à ce titre les parties peuvent tenter d'obliger l'oracle à garantir la sécurité des données et/ou son

⁷⁹⁰ Ozan Oguz Ceran. « Enhancing letters of credit with blockchain and smart contracts » p33. Master Thesis, Tilburg, 2019.

Some authors argue that, in blockchain-based letters of credit, IOT based terms can be included in letter of credit such as temperature of goods, oracles that transmits GPS signals from the goods or any condition regarding to goods ...

⁷⁹¹ V., *infra*, §91

⁷⁹² V., *infra*, §122

⁷⁹³ Thibault Verbiest. « Le Service Level Agreement dans les contrats informatiques ». Droit & Technologies (blog), 11 novembre 2003.

<https://www.droit-technologie.org/actualites/le-service-level-agreement-dans-les-contrats-informatiques/>.

Le cœur du SLA réside dans les clauses fixant le niveau des services attendus, à savoir le taux de disponibilité et de fiabilité du service (par exemple : les heures et les jours pendant lesquels le service sera disponible). Sera également précisé le temps de réponse qui devra être octroyé au fournisseur en cas de plainte pour dysfonctionnement (par exemple, prévoir que 95% des problèmes seront résolus après une heure à compter de la plainte).

⁷⁹⁴ Eskandari, Shayan, Mehdi Salehi, Wanyun Catherine Gu, et Jeremy Clark. « SoK: Oracles from the Ground Truth to Market Manipulation ». In Proceedings of the 3rd ACM Conference on Advances in Financial Technologies, 127-41. Arlington Virginia: ACM, 2021. <https://doi.org/10.1145/3479722.3480994>.

An oracle with deprecated or even malicious contents can have disastrous effects on all processes connected to the data feed. In practice, manipulated data feeds can cause significant damage, from unwarranted liquidations to malicious arbitrage trades. The following sections provide examples illustrating common vulnerabilities and malfunctions involving oracles.

processus de fourniture au *smart contract* ; dans le but de faciliter l'engagement de sa responsabilité.

X s'engage auprès des Parties à récupérer la donnée Y nécessaire à l'exécution du Smart contract à partir d'une source précise spécifiée dans l'annexe du présent contrat. X s'assurera de l'exactitude et de l'intégrité de la donnée récupérée avant de l'acheminer vers le Smart contract. Il s'assurera de la mise en place de toutes les mesures nécessaires pour prévenir toute altération, modification ou manipulation non autorisée de la donnée pendant son acheminement vers le Smart contract. Il s'engage à livrer la donnée au Smart contract dans un délai de et à une fréquence de ... La donnée sera fournie à l'adresse précise du Smart contract dans le format suivant :

b) Oracle-humain

290. Rédaction de la clause oracle-humain. Un humain peut aussi être chargé de fournir une information à un *smart contract* afin que ce dernier déclenche une action. Par exemple, un arbitre peut être tâché d'indiquer dans le contrat de séquestre intelligent laquelle des parties a eu gain de cause afin que le *smart contract* lui délivre les actifs séquestrés⁷⁹⁵. Dans une bonne mesure, les engagements de l'oracle-humain seront identiques à ceux de la personne responsable d'un oracle programme, avec quelques adaptations dues au fait qu'il s'agira d'une personne humaine qui devra interagir avec le *smart contract*. Ainsi la clause précisera que l'individu-oracle devra s'assurer d'approvisionner le *smart contract* à partir de sa clef privée et donc veiller qu'il s'agisse bien de son adresse *blockchain* personnelle.

B – Défaillance et responsabilité

291. Défaillance du *smart contract*. Les parties pourront faire le choix de faire un renvoi vers la clause d'imprévisibilité pour gérer les conséquences d'une défaillance technique de l'oracle. Elles veilleront ainsi à ajouter à la liste d'évènements susceptibles de déclencher la procédure d'imprévision, la défaillance de l'oracle que pourrait être la fourniture d'informations erronées ou altérées ou encore le défaut de fourniture de données dans les délais et fréquences prévus.

292. Responsabilité. Concernant la responsabilité, les parties pourront également faire un renvoi vers la clause de responsabilité ou la prévoir spécifiquement dans la clause d'engagement de

⁷⁹⁵ V., *infra*, §77

l'oracle⁷⁹⁶. Celle-ci sera alors chargée d'organiser le régime de responsabilité en cas de défaillance de la mission de l'oracle⁷⁹⁷. Les faits générateurs de responsabilité seront la violation des obligations précédemment abordées. En cas de survenance de l'un d'entre eux, les parties pourront imposer une obligation d'indemniser les dommages directs survenus suite au dysfonctionnement de l'oracle. Ceux-ci peuvent notamment consister en la compromission des actifs manipulés par le *smart contract*, de données personnelles et le retard ou la paralysie du *smart contract*.

En cas d'inexécution de l'un des engagements de X précédemment stipulés, celui-ci s'engage à indemniser les parties des dommages directs résultant de sa défaillance. Ces dommages peuvent être, sans qu'il y soient limités, la compromission des actifs ..., des Données personnelles ou le dysfonctionnement du Smart contract. La responsabilité de X ne pourra pas excéder le montant prévu dans la clause de responsabilité. En tout état de cause, elle ne saurait être caractérisée en cas de force majeure tel que défini à l'article....du présent contrat.

§ II - Clause de traitement de données personnelles

293. Traitement des données personnelles dans un contrat intelligent. Il est fréquent qu'une opération économique encadrée par un contrat mette en œuvre un traitement de données personnelles⁷⁹⁸. Celui-ci peut avoir lieu alors que les parties sont dans un rapport de responsables de

⁷⁹⁶ Blycha, Natasha, et Ariane Garside. « Smart Legal Contracts - A Model for the Integration of Machine Capabilities into Contracts ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 7 décembre 2020. <https://papers.ssrn.com/abstract=3743932>.

We recommend the inclusion of a clause setting out the mechanism for handling a situation where the Incorporated Code or Data Source does not perform as expected or intended. This clause should address: that assessment of whether a clause is performing as expected or intended should be done by reference to the natural language component of the Conjoined Term; (if relevant) nomination of Data Source Party and any relevant costs in respect of installation, running or maintenance of Data Source or Malfunctioning Data Source.

⁷⁹⁷ Mekki, Mustapha. « Blockchain : l'exemple des smart contracts ». Consulté le 20 mai 2023. <https://mustaphamekki.openum.ca/publications/1314/>.

En attendant, son intervention doit être minutieusement encadrée par un ensemble de stipulations contractuelles. Les clauses vont définir dans une lettre de mission la nature de ses obligations et leur intensité. Ces stipulations doivent également prévoir les conséquences d'une information qui fait défaut ou d'une information erronée : le contrat est-il anéanti et les parties récupèrent-elles les éventuels fonds déboursés ou séquestrés ?

⁷⁹⁸ Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE (règlement général sur la protection des données) Article 4 – 2§ - <http://data.europa.eu/eli/reg/2016/679/oj/fra>.

«traitement», toute opération ou tout ensemble d'opérations effectuées ou non à l'aide de procédés automatisés et appliquées à des données ou des ensembles de données à caractère personnel, telles que la collecte, l'enregistrement, l'organisation, la structuration, la conservation, l'adaptation ou la modification, l'extraction, la consultation, l'utilisation, la communication par transmission, la diffusion ou toute autre forme de mise à disposition, le rapprochement ou l'interconnexion, la limitation, l'effacement ou la destruction;

traitement⁷⁹⁹ et sous-traitants⁸⁰⁰ de données personnelles entre elles, ou dans une situation où l'une est responsable de traitement et l'autre une personne concernée par le traitement mis en œuvre dans le contrat. Dans ces cas-là, une clause relative aux données personnelles pourra être rédigée et contenir l'ensemble des informations qu'exige le règlement général sur la protection des données personnelles (RGPD) : les mentions exigées de l'article 28 du RGPD⁸⁰¹ dans le cadre d'une opération de sous-traitance ou celles de l'article 12 au titre des mentions d'informations que le responsable de traitement doit communiquer à une personne concernée par un traitement⁸⁰². La conformité au RGPD se réalisant surtout sur le plan technique⁸⁰³, la clause servira également à relater les choix d'implémentation afin de démontrer comment ces derniers satisfont les exigences du règlement. Nous l'étudierons en abordant les stipulations relatives à la gestion des données personnelles (A) puis celles relatives à leur transfert (B).

⁷⁹⁹ Ibid. Article 4 – 7§ - <http://data.europa.eu/eli/reg/2016/679/oj/fra>.

«responsable du traitement», la personne physique ou morale, l'autorité publique, le service ou un autre organisme qui, seul ou conjointement avec d'autres, détermine les finalités et les moyens du traitement; lorsque les finalités et les moyens de ce traitement sont déterminés par le droit de l'Union ou le droit d'un État membre, le responsable du traitement peut être désigné ou les critères spécifiques applicables à sa désignation peuvent être prévus par le droit de l'Union ou par le droit d'un État membre;

⁸⁰⁰ Ibid. Article 4 – 8§ - <http://data.europa.eu/eli/reg/2016/679/oj/fra>.

«sous-traitant», la personne physique ou morale, l'autorité publique, le service ou un autre organisme qui traite des données à caractère personnel pour le compte du responsable du traitement;

⁸⁰¹ Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE (règlement général sur la protection des données). Article 28 <http://data.europa.eu/eli/reg/2016/679/oj/fra>.

(...) 3. Le traitement par un sous-traitant est régi par un contrat ou un autre acte juridique au titre du droit de l'Union ou du droit d'un État membre, qui lie le sous-traitant à l'égard du responsable du traitement, définit l'objet et la durée du traitement, la nature et la finalité du traitement, le type de données à caractère personnel et les catégories de personnes concernées, et les obligations et les droits du responsable du traitement.

⁸⁰² Ibid. Article 12 <http://data.europa.eu/eli/reg/2016/679/oj/fra>.

1. Le responsable du traitement prend des mesures appropriées pour fournir toute information visée aux articles 13 et 14 ainsi que pour procéder à toute communication au titre des articles 15 à 22 et de l'article 34 en ce qui concerne le traitement à la personne concernée d'une façon concise, transparente, compréhensible et aisément accessible, en des termes clairs et simples, en particulier pour toute information destinée spécifiquement à un enfant...

⁸⁰³ Six, Nicolas, Claudia Negri Ribalta, Nicolas Herbaut, et Camille Salinesi. « A Blockchain-Based Pattern for Confidential and Pseudo-Anonymous Contract Enforcement ». In 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 1965-70, 2020. <https://doi.org/10.1109/TrustCom50675.2020.00268>.

A – Gestion des données personnelles

294. Données personnelles traitées. Qu'il s'agisse de mentions d'informations à destination d'une personne concernée ou d'une clause de sous-traitance, la clause relative aux données personnelles contiendra nécessairement une liste des données traitées⁸⁰⁴. Un *smart contract* traite, au moins, comme données personnelles, les adresses *blockchain*, qui peuvent identifier indirectement des individus⁸⁰⁵ et les données dites de « transaction » qui sont issues des transferts réalisés à partir de l'adresse *blockchain* d'une personne⁸⁰⁶. Il pourra également être rajouté toute autre donnée personnelle indirectement identifiante pouvant être manipulée par des smart contracts selon la nature du contrat.

295. Traitement de données personnelles. Les traitements de données personnelles effectués à travers les smart contracts pourront être indiqués comme étant :

- la collecte ; les données personnelles peuvent être fournies et/ou récupérées par/à un *smart*

⁸⁰⁴ Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE (règlement général sur la protection des données).
<http://data.europa.eu/eli/reg/2016/679/oj/fra>.

Article 28 : [le contrat entre le responsable de traitement et le sous-traitant] *définit (...) le type de données à caractère personnel et les catégories de personnes concernées...*

Article 14 : *Lorsque les données à caractère personnel n'ont pas été collectées auprès de la personne concernée, le responsable du traitement fournit à celle-ci toutes les informations suivantes : (...) d) les catégories de données à caractère personnel concernées...*

⁸⁰⁵ Michèle Finck. « Blockchain and the General Data Protection Regulation | Can distributed ledgers be squared with European data protection law ? » - p.27 - European Parliamentary Research Service, juillet 2019.

Practice reveals that public keys can enable the identification of a specified natural person. There have been instances where data subjects have been linked to public keys through the voluntary disclosure of their public key to receive funds (...) The combination of such records with the public key could thus reveal the real-world identity that lies hidden behind a blockchain address.

⁸⁰⁶ Ibid

Beyond, public keys may also reveal a pattern of transactions with publicly known addresses that could 'be used to single out an individual user' such as through transaction graph analysis...

contract,

- la diffusion ; elles sont, au moins, affichées en clair dans une *blockchain* publique⁸⁰⁷,
- l'enregistrement ; elles peuvent être stockées *on-chain*⁸⁰⁸,
- et le transfert⁸⁰⁹ ; enfin elles peuvent être envoyées d'une adresse à une autre ou d'un *smart contract* à un autre comme nous le verrons tantôt⁸¹⁰.

296. Finalités des traitements de données personnelles. Les parties pourront décrire la finalité du traitement⁸¹¹ en expliquant simplement le rôle du *smart contract* dans l'exécution du contrat, et pourquoi ce dernier a besoin de traiter des données personnelles pour effectuer ses tâches⁸¹².

297. Base juridique du traitement. La base juridique du traitement pourra être indiquée comme étant l'exécution du contrat, dans le cas où la personne concernée par le traitement est une partie⁸¹³. Dans l'hypothèse où la relation est celle d'un responsable de traitement et un sous-traitant,

⁸⁰⁷ V., *infra*, §2

⁸⁰⁸ V., *supra*, §421

⁸⁰⁹ Michèle Finck. « Blockchain and the General Data Protection Regulation | Can distributed ledgers be squared with European data protection law ? » - p.10 - European Parliamentary Research Service, juillet 2019.

In respect of blockchains, this very broad understanding of what counts as data processing implies that the initial addition of personal data to a distributed ledger, its continued storage and any further processing (such as for any form of data analysis but also to reach consensus on the current state of the network) constitutes personal data processing under Article 4(2) GDPR. Indeed, the European Court of Justice affirmed that personal data processing includes 'any operation or set of operations' performed on personal data.

⁸¹⁰ V., *supra*, §301

⁸¹¹ « Finalité d'un traitement | CNIL ». Consulté le 5 juin 2023. <https://www.cnil.fr/fr/definition/finalite-dun-traitement>.

La finalité du traitement est l'objectif principal de l'utilisation de données personnelles. Les données sont collectées pour un but bien déterminé et légitime et ne sont pas traitées ultérieurement de façon incompatible avec cet objectif initial. Ce principe de finalité limite la manière dont le responsable de traitement peut utiliser ou réutiliser ces données dans le futur.

⁸¹² Alice Barbet-Massin et William O'Rorke. « Fiche pratique n° 4317, Blockchain et données personnelles », LexisNexis360, n° 4317 (15 octobre 2019).

Par exemple, un besoin d'automatisation des prestations de sécurité sociale délivrées à différents bénéficiaires par un smart contract pourrait correspondre à une finalité explicite et légitime du traitement des adresses publiques par une blockchain.

⁸¹³ Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE (règlement général sur la protection des données). <http://data.europa.eu/eli/reg/2016/679/oj/fra>.

Article 6 - Licéité du traitement Le traitement n'est licite que si, et dans la mesure où, au moins une des conditions suivantes est remplie: (...) b) le traitement est nécessaire à l'exécution d'un contrat auquel la personne concernée est

elle dépendra de la mission confiée au sous-traitant. Ce pourra être :

- les obligations légales lorsque les opérations du *smart contract* exécutent des dispositions légales⁸¹⁴ ;
- ou l'intérêt légitime lorsque le traitement ne rentre dans aucune autre catégorie de l'article 6 du RGPD et qu'il ne heurte pas disproportionnellement les droits des personnes concernées par un traitement⁸¹⁵.

298. Durée de conservation des données personnelles. Compte tenu de la nature de la *blockchain*, la durée de conservation des données personnelles en son sein pourra potentiellement être infinie⁸¹⁶. Or, l'article 5 du RGPD impose une durée limitée de conservation des données personnelles⁸¹⁷. La clause relative aux données personnelles devra donc spécifier comment les parties ont œuvré pour satisfaire cette exigence du règlement malgré les caractéristiques de la *blockchain*.

299. Exercice des droits des personnes concernées par un traitement. Dans le cas d'une situation où l'une des parties est responsable de traitements et l'autre personne concernée par un traitement, la clause devra mentionner la manière dont cette personne pourra exercer ses droits sur

partie ou à l'exécution de mesures précontractuelles prises à la demande de celle-ci

⁸¹⁴ Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE (règlement général sur la protection des données).

Article 6 - Licéité du traitement Le traitement n'est licite que si, et dans la mesure où, au moins une des conditions suivantes est remplie : (...) c) le traitement est nécessaire au respect d'une obligation légale à laquelle le responsable du traitement est soumis;

⁸¹⁵ Ibid.

Article 6 - Licéité du traitement Le traitement n'est licite que si, et dans la mesure où, au moins une des conditions suivantes est remplie : (...) f) le traitement est nécessaire aux fins des intérêts légitimes poursuivis par le responsable du traitement ou par un tiers, à moins que ne prévalent les intérêts ou les libertés et droits fondamentaux de la personne concernée qui exigent une protection des données à caractère personnel, notamment lorsque la personne concernée est un enfant.

⁸¹⁶ CNIL. « Premiers éléments d'analyse de la CNIL sur la Blockchain - p.7 », septembre 2018.
https://www.cnil.fr/sites/default/files/atoms/files/la_blockchain.pdf.

L'architecture même de la Blockchain fait que les identifiants seront toujours visibles, car ils sont indispensables à son bon fonctionnement. La CNIL considère donc qu'il n'est pas possible de les minimiser davantage et que leurs durées de conservation sont, par essence, alignées sur celles de la durée de vie de la Blockchain.

⁸¹⁷ Article 5 du RGPD : 1. Les données à caractère personnel doivent être: (...) conservées sous une forme permettant l'identification des personnes concernées pendant une durée n'excédant pas celle nécessaire au regard des finalités pour lesquelles elles sont traitées; les données à caractère personnel peuvent être conservées pour des durées plus longues dans la mesure où elles seront traitées exclusivement à des fins archivistiques dans l'intérêt public, à des fins de recherche scientifique ou historique ou à des fins statistiques conformément à l'article 89, paragraphe 1, pour autant que soient mises en œuvre les mesures techniques et organisationnelles appropriées requises par le présent règlement afin de garantir les droits et libertés de la personne concernée (limitation de la conservation);

ses données personnelles. La CNIL a évoqué dans son rapport les difficultés que pourraient rencontrer ces personnes si le traitement de leurs données est opéré dans une *blockchain* publique. Ainsi, le droit d'effacement et de rectification des données paraissent incompatibles avec la nature immuable du protocole⁸¹⁸, même s'il est évoqué dans ce même rapport, qu'il pourrait être mis en œuvre des schémas aboutissant à un résultat s'approchant des effets d'effacement et de rectification⁸¹⁹. La clause devra relater ces mesures techniques mises en place afin que les personnes puissent exercer leurs droits.

300. Mesures de sécurité. Enfin, la clause relative aux mesures de sécurité devra faire état de l'ensemble des moyens mis en place pour empêcher la violation des données personnelles traitées par les smart contracts. Dans son rapport, la CNIL recommande au responsable de traitement un devoir de vigilance dans le choix de la *blockchain* : il devra notamment s'assurer que celle-ci soit bien décentralisée et avoir installé des logiciels permettant de mettre à jour les smart contracts en cas de bogues de ceux-ci.⁸²⁰

B – Transfert de données personnelles

301. Transfert d'un sous-traitant vers un sous-traitant ultérieur. À l'occasion d'un traitement de données personnelles dans la *blockchain*, il est probable que des données soient transférées d'un *smart contract* vers un autre. Par exemple, des données identifiantes peuvent figurer

⁸¹⁸ CNIL. « Premiers éléments d'analyse de la CNIL sur la Blockchain - », p.9, septembre 2018.
https://www.cnil.fr/sites/default/files/atoms/files/la_blockchain.pdf.

Si l'exercice effectif de certains droits ne semble pas poser de difficultés, l'application du droit à l'effacement, du droit de rectification et du droit d'opposition à la Blockchain méritent une analyse plus détaillée.

⁸¹⁹ Ibid.

La CNIL constate qu'il est techniquement impossible de faire droit à la demande d'effacement de la personne concernée lorsque des données sont inscrites dans la Blockchain. Toutefois, lorsque la donnée inscrite sur la Blockchain est un engagement, une empreinte issue d'une fonction de hachage à clé ou un chiffré utilisant un algorithme et des clés conformes à l'état de l'art, le responsable de traitement peut rendre la donnée quasi inaccessible, et se rapprocher ainsi des effets d'un effacement de la donnée.

⁸²⁰ CNIL. « Premiers éléments d'analyse de la CNIL sur la Blockchain - p.11 », septembre 2018.
https://www.cnil.fr/sites/default/files/atoms/files/la_blockchain.pdf.

Dans le cas des blockchains à permission, la CNIL recommande d'évaluer, en fonction de l'éventuelle divergence ou convergence des intérêts des acteurs participants, un minimum de mineurs permettant d'assurer l'absence de coalition permettant de contrôler plus de 50% des pouvoirs sur la chaîne. La CNIL recommande également de mettre en place des procédures techniques et organisationnelles pour limiter l'impact sur la sécurité des transactions de l'éventuelle défaillance d'un algorithme (notamment cryptographique), y compris un plan d'urgence à mettre en œuvre permettant de modifier les algorithmes lorsqu'une vulnérabilité est identifiée.

dans un NFT et le contrat intelligent peut prévoir que celui-ci sera transféré, à l'accomplissement d'une condition, vers un autre *smart contract*. Dans ce cas-là, il peut légitimement être demandé si ce transfert doit être considéré comme un transfert à un sous-traitant ultérieur⁸²¹, dès lors que celui l'initiant est un sous-traitant de données personnelles.

Or, l'article 28 du RGPD impose que le sous-traitant obtienne une autorisation préalable du responsable de traitement avant de transférer ses données personnelles à un sous-traitant ultérieur⁸²². Celle-ci peut être spécifique ou générale, étant entendu que dans le premier cas, *le sous-traitant doit informer le responsable du traitement de tout changement prévu concernant l'ajout ou le remplacement d'autres sous-traitants, lui donnant ainsi la possibilité d'émettre des objections à l'encontre de ces changements.*⁸²³ La CNIL a énoncé que le responsable de traitement doit déterminer si ce traitement ultérieur est compatible avec la finalité pour laquelle les données ont été initialement collectées ; pour cela le responsable de traitement doit procéder à un test de compatibilité. Elle en déduit qu'une autorisation préalable et générale de réutilisation des données n'est pas légale⁸²⁴. Mais cette vision n'est pas celle de la lettre de l'article 28 du RGPD.

Ainsi, les parties à un contrat intelligent dans une relation de responsable – sous-traitant de données

⁸²¹ Comité européen de la protection des données. « Lignes directrices 07/2020 concernant les notions de responsable du traitement et de sous-traitant dans le RGPD » 1.6 §151, 7 juillet 2021.

Les activités de traitement des données sont souvent effectuées par un grand nombre d'acteurs et les chaînes de sous-traitance deviennent de plus en plus complexes. Le RGPD introduit des obligations spécifiques qui sont déclenchées lorsqu'un sous-traitant (ultérieur) envisage de recruter un autre acteur, ajoutant ainsi un maillon supplémentaire à la chaîne, en lui confiant des activités nécessitant le traitement de données à caractère personnel.

⁸²² Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE (règlement général sur la protection des données).
<http://data.europa.eu/eli/reg/2016/679/oj/fra>.

Article 28 - Sous-traitant : (...) *Le sous-traitant ne recrute pas un autre sous-traitant sans l'autorisation écrite préalable, spécifique ou générale, du responsable du traitement. Dans le cas d'une autorisation écrite générale, le sous-traitant informe le responsable du traitement de tout changement prévu concernant l'ajout ou le remplacement d'autres sous-traitants, donnant ainsi au responsable du traitement la possibilité d'émettre des objections à l'encontre de ces changements.*

⁸²³ « Sous-traitants : la réutilisation de données confiées par un responsable de traitement | CNIL ». Consulté le 31 mai 2023. <https://www.cnil.fr/fr/sous-traitants-la-reutilisation-de-donnees-confiees-par-un-responsable-de-traitement>.

Ce « test de compatibilité » doit être réalisé pour un traitement déterminé, en tenant compte des finalités et des caractéristiques de chaque traitement pour lequel le sous-traitant souhaite réutiliser les données. Cela signifie qu'une autorisation préalable et générale de réutilisation des données n'est pas légale.

⁸²⁴ « Sous-traitants : la réutilisation de données confiées par un responsable de traitement | CNIL ». Consulté le 31 mai 2023. <https://www.cnil.fr/fr/sous-traitants-la-reutilisation-de-donnees-confiees-par-un-responsable-de-traitement>.

Ce « test de compatibilité » doit être réalisé pour un traitement déterminé, en tenant compte des finalités et des caractéristiques de chaque traitement pour lequel le sous-traitant souhaite réutiliser les données. Cela signifie qu'une autorisation préalable et générale de réutilisation des données n'est pas légale.

personnelles pourront prévoir dans leur clause relative aux données personnelles, une autorisation préalable générale du responsable de traitement pour que les données puissent être traitées par des sous-traitants ultérieurs. Etant donné que le comportement du *smart contract* est déterministe⁸²⁵, le sous-traitant devrait savoir à l'avance vers qui les données pourront être transférées et s'assurera que ces derniers respectent les mêmes obligations auxquels il s'est engagé envers le responsable de traitement. Il devra également informer ce dernier de tout ajout de sous-traitants ultérieurs et de la possibilité de s'opposer à l'ajout de nouveaux.

302. Transferts de données personnelles en dehors de l'Union européenne. Les transferts de données personnelles vers des pays se situant en dehors de l'Union Européenne soulèvent également des interrogations dans le cadre d'une exécution contractuelle par *smart contract*. En effet, dans une *blockchain*, les données peuvent circuler d'un validateur vers un autre, qui peut se situer dans un pays en dehors du champ d'application du RGPD⁸²⁶. Lorsque la *blockchain* est privée/permissionnée, il est possible d'identifier ces validateurs et ainsi de les contraindre à respecter des mesures assurant un niveau de protection équivalent à celui conféré par le RGPD⁸²⁷ : règles d'entreprises contraignantes⁸²⁸ et clauses types⁸²⁹. Mais dans le cas d'une *blockchain* publique, les validateurs ne sont pas identifiables et il n'y a donc pas de possibilité, *a priori*, de leur imposer le respect des mesures fournissant une protection d'un niveau équivalent au RGPD.⁸³⁰ Si la relation entre

⁸²⁵ C'est-à-dire que les parties savent à l'avance ce que leur smart contracts fera.

⁸²⁶ Finck, Michèle. « Blockchains and Data Protection in the European Union ». SSRN Scholarly Paper. Rochester, NY, 30 novembre 2017. <https://doi.org/10.2139/ssrn.3080322>.

On permissionless ledgers we can presume that there is always an element of cross-border data processing.

⁸²⁷ CNIL. « Premiers éléments d'analyse de la CNIL sur la Blockchain - p.6 », septembre 2018. https://www.cnil.fr/sites/default/files/atoms/files/la_blockchain.pdf.

Il convient de privilégier une Blockchain à permission qui permet d'avoir une meilleure maîtrise sur la gouvernance de la donnée personnelle, s'agissant notamment des transferts hors UE. Les solutions existantes permettant d'encadrer les transferts hors UE, / tels que les règles d'entreprises contraignantes ou les clauses contractuelles types, sont entièrement applicables dans la Blockchain à permission.

⁸²⁸ Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE (règlement général sur la protection des données). Article 47 du RGPD. <http://data.europa.eu/eli/reg/2016/679/oj/fra>.

⁸²⁹ Ibid. Article 28 du RGPD <http://data.europa.eu/eli/reg/2016/679/oj/fra>.

⁸³⁰ CNIL. « Premiers éléments d'analyse de la CNIL sur la Blockchain - p.6 », septembre 2018. https://www.cnil.fr/sites/default/files/atoms/files/la_blockchain.pdf.

Par exemple, la question des transferts hors de l'Union Européenne (UE) peut s'avérer particulièrement problématique, notamment dans le cadre d'une Blockchain publique.

les parties est celle d'un responsable de traitement vis-à-vis d'une personne concernée par un traitement, elles pourront néanmoins fonder le transfert sur le b) de l'article 49 du RGPD, disposant que celui-ci est autorisé lorsqu'il est nécessaire à la conclusion d'un contrat⁸³¹. Dans le cas d'une relation responsable de traitement / sous-traitant, elles pourront s'appuyer sur le c) de l'article 49 si les données personnelles sont traitées sur la base d'un contrat conclu dans l'intérêt de la personne concernée⁸³². Si cette option n'est pas applicable, il restera éventuellement à solliciter le consentement de la personne concernée au transfert. Elle est à considérer en dernier car la personne concernée est susceptible de retirer son consentement à tout moment, heurtant potentiellement le bon fonctionnement du *smart contract*.⁸³³

303. Conclusion de la section II et du chapitre II. Une fois écrites les stipulations relatives à l'exécution du contrat, les parties auront donc prévu dans d'autres leurs rapports avec des tiers pouvant intervenir dans l'exécution de leurs contrats intelligents : les oracles et les personnes concernées par des traitements de données personnelles. Dans les premiers, les parties veilleront à ce que les oracles s'engagent vis-à-vis d'elles similairement à un prestataire de services *cloud* : en respectant différents niveaux de services. S'agissant des personnes concernées par un traitement de données personnelles, les parties veilleront à respecter les article 12 et 28 du RGPD imposant un

⁸³¹ Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE (règlement général sur la protection des données). Article 45
<http://data.europa.eu/eli/reg/2016/679/oj/fra>.

En l'absence de décision d'adéquation en vertu de l'article 45, paragraphe 3, ou de garanties appropriées en vertu de l'article 46, y compris des règles d'entreprise contraignantes, un transfert ou un ensemble de transferts de données à caractère personnel vers un pays tiers ou à une organisation internationale ne peut avoir lieu qu'à l'une des conditions suivantes: (...) b) le transfert est nécessaire à l'exécution d'un contrat entre la personne concernée et le responsable du traitement ou à la mise en œuvre de mesures précontractuelles prises à la demande de la personne concernée;

⁸³² Ibid.

En l'absence de décision d'adéquation en vertu de l'article 45, paragraphe 3, ou de garanties appropriées en vertu de l'article 46, y compris des règles d'entreprise contraignantes, un transfert ou un ensemble de transferts de données à caractère personnel vers un pays tiers ou à une organisation internationale ne peut avoir lieu qu'à l'une des conditions suivantes: (...) c) le transfert est nécessaire à la conclusion ou à l'exécution d'un contrat conclu dans l'intérêt de la personne concernée entre le responsable du traitement et une autre personne physique ou morale;

⁸³³ Alice Barbet-Massin et William O'Rorke. « Fiche pratique n° 4317, Blockchain et données personnelles », LexisNexis360, n° 4317 (15 octobre 2019).

Enfin, le traitement pourrait être fondé sur l'exécution d'un contrat, notamment, dans le cadre d'une blockchain privée encadrée par un contrat prévoyant les règles applicables aux participants identifiés et à la gouvernance. Il conviendra d'éviter de fonder son traitement sur le consentement en ce qu'il doit être recueilli au préalable auprès de la personne concernée et démontré, conformément à l'article 7.1. du RGPD. De surcroît, ce dernier pourra être retiré à tout moment, au risque de priver le traitement de fondement juridique.

certain nombre de mentions dans les clauses de sous-traitance et de mention d'informations.

Chapitre III - Stipulations terminales du contrat

304. Clauses de fin de contrat. Parmi les stipulations qu'il est courant de trouver à la fin du corps d'une convention, on compte les clauses relatives à l'extinction du contrat (Section I) et celles concernant la résolution des conflits⁸³⁴(Section II). Les mécanismes institués par ces clauses auront la particularité de faire actionner des smart contracts exécutant le contrat intelligent.

Section I - Interruption du contrat

305. Pause et fin du contrat. En sus de l'évènement imprévisible, la force majeure est un autre évènement susceptible d'interrompre le contrat de manière provisoire ou définitive (§1). Ces différents types d'interruption pourront être détaillés dans une clause spécifique qui organisera sa mise en œuvre avec les smart contracts exécutant le contrat intelligent (§2I).

§ I - Clause de force majeure

306. Qualification et régime dans la clause de force majeure. La force majeure est définie comme un événement irrésistible, extérieur et imprévisible rendant impossible l'exécution des obligations d'un contrat⁸³⁵. Contrairement à l'évènement de la clause d'imprévision, la conséquence de la qualification d'un événement de force majeure est l'impossibilité de poursuivre l'exécution des obligations, ce qui suspend nécessairement l'exécution du contrat si l'évènement est temporaire ou le termine si l'évènement a un caractère permanent⁸³⁶. La partie débitrice d'une obligation victime

⁸³⁴ Ainsi, dans le modèle universel de contrat d'affaire de Francis Lefebvre, on trouve les clauses relatives à l'extinction du contrat et celles relatives à la force majeure vers la toute fin de la convention (articles 274 et 258 respectivement).

« Modèle Universel De Contrat d'Affaires », Francis Lefebvre, 11 décembre 2019.

⁸³⁵ Article 1218 du code civil : *Il y a force majeure en matière contractuelle lorsqu'un événement échappant au contrôle du débiteur, qui ne pouvait être raisonnablement prévu lors de la conclusion du contrat et dont les effets ne peuvent être évités par des mesures appropriées, empêche l'exécution de son obligation par le débiteur.*

⁸³⁶ Cécile BIGUENET-MAUREL. « Force majeure et imprévision : des outils de droit commun pour faire face au Covid-19 », La quotidienne Francis Lefebvre - Affaires et exécution, 15 avril 2020.

La force majeure et l'imprévision ont en commun l'imprévisibilité de la survenance d'un événement postérieur au contrat, mais elles se distinguent en ce que la force majeure rend impossible l'exécution du contrat tandis que l'imprévision la rend excessivement onéreuse (Rapp. Sén. n° 22 relatif à la loi 2018-287 du 20-4-2018).

empêchée par une force majeure sera ainsi exonérée de sa responsabilité contractuelle⁸³⁷.

Une clause de force majeure pourra servir à aménager, dans la mesure de ce qui est légalement possible, le régime légal disposé à l'article 1218 du Code civil. Elle pourra être utilisée afin de convenir quels seront les événements constituant automatiquement des événements de force majeure (A) et déclenchant la procédure à suivre pour gérer les conséquences de leur survenance⁸³⁸ (B).

A – Qualification d'un événement de force majeure

307. Définition abstraite et non exhaustive des événements constituant une force majeure. Les parties seront donc avisées de définir par elles-mêmes les événements liés à la *blockchain* et aux smart contracts qui seront constitutifs de cas de force majeure. Elles pourront procéder selon la même méthode que pour la clause d'imprévision : à savoir définir de manière générale ce que sera un événement de force majeure, puis lister non exhaustivement les différents cas⁸³⁹.

L'inexécution de tout ou partie de ses obligations par l'une ou l'autre des parties ne pourra engager sa responsabilité si l'inexécution est due à un événement de force majeure, tel que prévu à l'article 1218 du Code civil. Par dérogation à ce texte et de convention expresse, les événements suivants seront réputés constitutifs de cas de force majeure, indépendamment des critères d'irrésistibilité, d'imprévisibilité et d'extériorité s'ils sont indépendants de la volonté des parties et même s'ils ne sont que partiels⁸⁴⁰ : ...

Plusieurs des événements constitutifs d'un cas de force majeure auront déjà été listés comme des

⁸³⁷ Article 1351 du code civil : *L'impossibilité d'exécuter la prestation libère le débiteur à due concurrence lorsqu'elle procède d'un cas de force majeure et qu'elle est définitive, à moins qu'il n'ait convenu de s'en charger ou qu'il ait été préalablement mis en demeure.*

⁸³⁸ Fabrice GRÉAU. « Répertoire de droit civil - Chapitre 4 - Force majeure – Les aménagements contractuels de la force majeure - 105 Clauses relatives à la force majeure », Dalloz, juin 2017.

L'article 1351 du code civil précise ainsi que la libération du débiteur à raison de l'impossibilité d'exécuter joue « à moins qu'il n'ait convenu de s'en charger ». La phrase est quelque peu réductrice car ces clauses peuvent certes avoir pour objet d'étendre les obligations du débiteur lorsque celui-ci accepte de prendre en charge la force majeure (clause de garantie), mais elles peuvent également tenter d'élargir la force majeure, soit par une définition plus accueillante que celles qui ont cours dans la loi ou en jurisprudence, soit et plus fréquemment par l'énumération d'événements qui doivent être contractuellement considérés comme une force majeure alors même que ferait défaut l'un des caractères habituellement requis ...

⁸³⁹ V., *infra*, §274

⁸⁴⁰ Law Insider. « Force Majeure | Sample Clauses ». Consulté le 13 juin 2023.
<https://www.lawinsider.com/fr/clause/force-majeure>.

événements imprévisibles⁸⁴¹. Toutefois les parties devront les mentionner à nouveau dans leur clause de force majeure afin qu'ils déclenchent les effets spécifiques liés à celle-ci : soit la suspension automatique des obligations et l'exonération de la responsabilité contractuelle. Ceux-ci peuvent donc être :

- une attaque ou un bogue prolongé du consensus de la *blockchain* sur laquelle sont déployés les smart contracts,
- une défaillance d'un *smart contract* avec lequel des parties interagissent,
- une mesure ou décision légale imposant une prohibition de l'usage des smart contracts et/ou d'une *blockchain*⁸⁴²,
- une compromission d'un logiciel⁸⁴³ essentiel au fonctionnement et à l'usage des smart contracts comme un *wallet* ou un client.

B – Régime juridique de la force majeure à suivre

308. Mise en œuvre de la clause de force majeure. La clause de force majeure prévoira également la procédure à suivre par les parties en cas de survenance d'un tel événement⁸⁴⁴. Si ce dernier est de nature temporaire, le débiteur affecté sera dispensé de l'exécution de ses obligations, tandis qu'en cas de permanence de la durée de l'évènement, les parties auront toutes les deux la

⁸⁴¹ V., *infra*, §274

⁸⁴² Ainsi dans la proposition d'accord juridique de Gabriel Shapiro, un cas de MAC est celui d'une décision légale qui interdit l'usage d'un *smart contract*.

_g4brielShapiro0. « SCoDA – Simple Code Deference Agreement », 25 avril 2023. <https://github.com/lex-node/SCoDA-Simple-Code-Deference-Agreement->.

(o) “Material Adverse Exception Event” means that one or more of the following has occurred, is occurring or would reasonably be expected to occur: (...) (v) the Designated Smart contract[, any of the Parties] or the Subject Property is subject to a Legal Order that prohibits the Designated Smart contract [(or that, if the Designated Smart contract were a Person, would prohibit the Designated Smart contract)] from executing any function or operation it would otherwise reasonably be expected to execute.

⁸⁴³ Mekki, Mustapha. « Les mystères de la blockchain ». Recueil Dalloz, n° 37 (2 novembre 2017): 2160.

Il va donc falloir penser à rédiger des clauses encadrant ce risque : clause de force majeure pour imputer le risque d'une indisponibilité du réseau, d'une cyber attaque, d'une corruption de données lors de l'hébergement ou lors du transport de ces données.

⁸⁴⁴ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires – Clause de force majeure - §VI – Conseils rédactionnels 755 - . Les Intégrales. LGDJ, 2018.

La clause devra ensuite détailler le régime applicable à la force majeure. On peut notamment préciser l'effet exonératoire et le sort du contrat en cas d'inexécution temporaire ou partielle, et le comportement à adopter par les parties durant cette période de suspension : obligation d'information – notification, preuve - de diligence, de limitation du dommage ou de la charge des frais engagés par exemple.

possibilité de résilier le contrat. Les smart contracts exécutant le contrat intelligent seront alors également suspendus ou terminés⁸⁴⁵.

En cas de survenance d'un événement de Force Majeure, la partie affectée par cet événement sera temporairement dispensée de l'exécution de ses obligations contractuelles, ce qui entraînera la suspension du contrat ainsi que celle du Smart contract des Parties dans les conditions prévues à l'article [...] des présentes. La partie affectée devra immédiatement notifier à l'autre partie, par [...], de la survenance de l'événement de Force Majeure et de l'impact prévu sur l'exécution du contrat. Si l'événement de Force Majeure persiste pendant une période continue de [...] jours, chaque partie aura le droit de résilier le présent contrat et mettre fin au fonctionnement du Smart contract des Parties dans les conditions prévues à l'article [...] des présentes.

Aucune des parties ne pourra être tenue responsable envers l'autre partie pour tout manquement ou retard dans l'exécution de ses obligations contractuelles causé par un événement de Force Majeure, à condition que ladite partie ait notifié l'événement de Force Majeure conformément à la présente clause.

§ II - Clauses de suspension et résiliation

309. Suspension et résiliation du contrat. Plusieurs événements peuvent mener à l'arrêt temporaire ou définitif du contrat intelligent. À ce titre, les parties seront avisées de prévoir des clauses de suspension (A) et de résiliation (B) dans lesquelles elles décriront leurs mise en œuvre et y feront renvoi dans le contrat dès que cela sera nécessaire.

A - Clause de suspension

310. Définition de la clause de suspension. La suspension du contrat permet aux parties de se dégager d'obligations dont elles ne veulent ou peuvent plus assurer la charge⁸⁴⁶. Le but du mécanisme est de faire survivre le contrat jusqu'à ce que l'évènement bouleversant son exécution disparaisse⁸⁴⁷. La clause sert donc à décrire la manière précise dont sera mis en œuvre cette suspension

⁸⁴⁵ V., *supra*, §446 et §449

⁸⁴⁶ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires – Clause de suspension - §I – Objet et utilité 1751 - . Les Intégrales. LGDJ, 2018.

(...) Elle sert le plus souvent à permettre aux parties de se dégager d'obligations dont elles ne peuvent ou ne veulent pas assurer la charge lorsque se produit un évènement qui alourdit les obligations ou rend le contrat moins profitable, tout en assurant la pérennité du contrat.

⁸⁴⁷ Christophe Jamin, Marc Billiau, et Jacques Ghestin. Les effets du contrat. 3e éd. N°398, 2001.

: combien de temps le contrat sera interrompu, quelles obligations devront continuer à être exécutées et comment le contrat sera rétabli.

311. Causes de suspension. La clause commencera par énumérer les causes de suspension. Certaines auront déjà été mentionnées dans la clause d'imprévision⁸⁴⁸ et de force majeure⁸⁴⁹, mais elles pourront être définies, une nouvelle fois, de façon abstraite dans cette clause comme :

*Tout événement quelconque imprévu empêche l'exécution des obligations incombant à l'une ou à l'autre partie*⁸⁵⁰.

312. Procédure de la clause de suspension. Surtout la clause décrira le régime de la suspension. Elle sera activée de concert avec le *smart contract* et devra donc prévoir dans son contenu les modalités de sa mise en œuvre⁸⁵¹. La spécificité de cette clause dans notre contexte est qu'elle occasionnera, également, la suspension du *smart contract*. Les parties pourront faire un renvoi vers l'annexe technique pour préciser de quelle manière concrètement elles devront s'y prendre pour pauser le *smart contract* puis le remettre en fonctionnement.

La suspension s'appliquera aux obligations que les parties désigneront au moment où l'événement se produira. Elles pourront suspendre les smart contracts des Parties selon les modalités décrites en annexe technique des présentes. La suspension ne concernera pas les obligations suivantes : [...]. Elle cessera de produire effet lorsque l'événement paralysant l'exécution du contrat dépassera une durée de [...] à compter du jour où la suspension aura pris effet ; les parties jugeant qu'une prolongation de la suspension au-delà de cette durée compromettrait l'économie et les objectifs du contrat, déclarent expressément que le contrat sera caduc de droit à l'échéance du terme. La reprise de l'exécution du Contrat, notamment la remise ou en fonctionnement ou le redéploiement du Smart contract des Parties, sera réglée par les parties au moment de la fin de la suspension avec, au besoin,

[La suspension] est ainsi un mécanisme d'adaptation du contrat au service de sa force obligatoire.

⁸⁴⁸ V., *infra*, §274

⁸⁴⁹ V., *infra*, §307

⁸⁵⁰ « Modèle Universel De Contrat D'affaires - Suspension de l'exécution du contrat pour empêchement momentané d'exécution », Francis Lefebvre, 11 décembre 2019.

(...) il y a lieu à suspension partielle ou complète de l'exécution du Contrat dès lors qu'un événement quelconque imprévu empêche l'exécution des obligations incombant à l'une ou à l'autre partie.

⁸⁵¹ Lord Chancellor and Secretary of State for Justice. « Smart legal contracts - Advice to Government » p.212, novembre 2021.

(...) Similarly, to avoid a scenario where the code performs pending the outcome of a dispute, parties would be well advised to provide a mechanism for suspension of performance of the code in their smart legal contract.

B – Clause de résiliation

313. Définition de la clause de résiliation. La clause de résiliation peut être le terme donné aux différents types de clauses visant à terminer un contrat⁸⁵³. Cette fin peut être provoquée au titre d'une sanction de l'inexécution d'une obligation ; la clause peut alors être résolutoire⁸⁵⁴ et avoir des effets rétroactifs⁸⁵⁵. Elle peut aussi se faire au titre d'une rupture unilatérale du contrat, par exemple pour cause d'imprévision ou de force majeure. Quelle que soit la cause, elle a pour but de contrôler les modalités de terminaison du contrat⁸⁵⁶.

314. Procédure de fin du contrat intelligent. La clause de résiliation appellera un traitement particulier dans notre contexte puisqu'elle sera mise en œuvre conjointement avec les smart contracts exécutant le contrat : elle sera ainsi accompagnée de la désactivation de ces derniers⁸⁵⁷. Le mécanisme institué par la clause devra donc détailler les conséquences de la mise à l'arrêt du *smart*

⁸⁵² « Modèle Universel De Contrat D'affaires - Suspension de l'exécution du contrat pour empêchement momentané d'exécution », Francis Lefebvre, 11 décembre 2019.

⁸⁵³ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires – Clause de résiliation - §I – Objet et utilité 1617 - . Les Intégrales. LGDJ, 2018.

Sous l'appellation « clause de résiliation », se dissimulent en pratique plusieurs types de clauses qui n'ont en commun que leur principal effet, mettre fin au contrat.

⁸⁵⁴ Article 1224 du code civil : *La résolution résulte soit de l'application d'une clause résolutoire soit, en cas d'inexécution suffisamment grave, d'une notification du créancier au débiteur ou d'une décision de justice.*

⁸⁵⁵ La jurisprudence emploie pourtant indifféremment les termes de résolution et résiliation.

Cour de cassation, civile, Chambre civile 3, 21 décembre 2017, 16-10.583, Publié au bulletin.

⁸⁵⁶ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires – Clause résolutoire - §I – Objet et utilité 1642 - . Les Intégrales. LGDJ, 2018.

La clause résolutoire est l'une des plus usitées en pratique, en ce qu'elle aménage contractuellement les conséquences de l'inexécution.

⁸⁵⁷ Jeffrey D. Neuburger, Wai L. Choy, et Kevin P. Milewski. « Smart contracts: Best Practices », Thomson Reuters - Practical Law, 2019, 11-19.

Kill Mechanism. As a fail-safe mechanism, parties may want to include in their smart contract a kill function that the parties can exercise if an issue warrants it. The parties should carefully consider the circumstances in which a party should be able to run that function (as, for example, the ability to activate the kill function could amount to a termination for convenience right).

contract⁸⁵⁸ en sus de celle de l'accord juridique.

Une partie pourra résilier le contrat en cas de survenance d'un évènement de force majeure tel que défini à l'article [...] ou d'un évènement prévisible tel que défini à l'article [...]. Elle pourra également résilier le contrat en cas d'inexécution d'une obligation contractuelle par la partie débitrice d'une obligation d'une des présentes, après l'avoir mis en demeure de s'exécuter dans un délai de [...] par LRAR sous peine de résiliation du contrat.

Dans tous les cas, la résiliation du contrat devra être suivie par l'Actionnement de la fonction « finDuContrat » du Smart contract des Parties qui Transférera les actifs numériques éventuellement séquestrées selon les modalités décrites dans l'annexe technique.

315. Conclusion de la section I. Les parties pourront donc situer, à la fin du contrat, les clauses relatives à son interruption. Dans une clause de force majeure, elles listeront, similairement à la clause d'imprévision, les évènements constitutifs de tels cas, et spécifiques au contexte d'une exécution dans la *blockchain* ; puis elles détailleront les conséquences contractuelles de la survenance de l'un d'entre eux. Ces conséquences pourront être la suspension ou la résiliation du contrat, qui seront également détaillés dans des clauses spéciales dans lesquelles sera précisée la manière dont les smart contracts seront, eux aussi, suspendus ou terminés.

Section II – Clauses relatives à la résolution des conflits

316. Résolution des conflits. Les parties ont à leur disposition deux types de manières de résoudre les conflits pouvant naître à l'occasion de l'exécution de leur contrat intelligent : elles peuvent recourir à des modes alternatifs de règlements des conflits (MARC), en prévoyant une clause de médiation et/ou compromissoire dans leur accord écrit (§1) ; et dans les cas où elles ne pourraient pas faire usage de ces procédés de résolution de conflits, elles seront avisées de ménager, dans la mesure du possible, la loi et la juridiction applicables pour régler leurs litiges (§2).

⁸⁵⁸ Lord Chancellor and Secretary of State for Justice - « Smart legal contracts - Advice to Government » §5.126, p.139 novembre 2021.

As a practical matter, the party who elects to terminate the contract may not have the power to terminate performance of the code, particularly if the code is recorded on an immutable distributed ledger. This may lead to practical difficulties if the code continues to execute transactions or confer benefits after the contract has been terminated for breach (...) In any event, the ability to stop performance of the code may be necessary or desirable, both to accommodate termination for breach, and also to ensure performance of the code is ended in other contexts discussed in this chapter, such as frustration.

§ I - Clauses de modes alternatifs de règlement des conflits

317. Le recours à la médiation et/ou l'arbitrage. Compte tenu des spécificités des différends soulevés à l'occasion de l'exécution d'un contrat intelligent, les parties auront grand intérêt à faire recours aux modes alternatifs de règlement des conflits (MARC) (A), adaptés à une exécution du contrat dans la *blockchain* (B).

A – Intérêt des MARC pour un contrat intelligent

318. Définition des MARC. Les MARC désignent des modes alternatifs de résolution des conflits ne faisant pas intervenir l'office du juge⁸⁵⁹. Ils comprennent la médiation⁸⁶⁰, la conciliation⁸⁶¹, la transaction⁸⁶² et l'arbitrage⁸⁶³. Les clauses de recours aux MARC organisent donc, dans le contrat, la manière dont les parties géreront un différend entre elles, sans recourir au juge⁸⁶⁴. Leurs vertus sont multiples : elles permettent aux parties de bénéficier d'une solution de règlement de leurs conflits

⁸⁵⁹ Devigny, Emmanuelle. « Médiation, Conciliation, Arbitrage, Négociation : C'est quoi les MARC ? » Officiel de la Médiation Professionnelle et de la Profession de Médiateur (blog), 16 janvier 2017.

Les modes alternatifs de règlement des conflits (MARC) sont des processus volontaires par lesquels les parties résolvent leurs différends, sans procédure judiciaire, mais avec l'aide d'un tiers.

⁸⁶⁰ Debard, Thierry, et Serge Guinchard. Lexique des termes juridiques 2020-2021 - 28e ed. Médiation conventionnelle. Edition 2020-2021. Dalloz, 2020.

(...) désigne la clause d'un contrat par laquelle les parties s'engagent à recourir à un tiers, dénommé médiateur, pour tenter de trouver une solution amiable dans l'hypothèse où un différend surviendrait entre elles à propos du contrat et ne saisir le juge qu'en cas d'échec de cette médiation.

⁸⁶¹ Ibid. Conciliation.

(...) désigne la clause d'un contrat par laquelle les parties s'engagent à tenter de trouver une solution amiable avec d'un tiers dénommé conciliateur, dans l'hypothèse où un différend surviendrait entre elles et à ne saisir le juge qu'en cas d'échec de la tentative de conciliation ou pour faire homologuer leur accord.

⁸⁶² Ibid. Transaction

Contrat par lequel les parties terminent une contestation née ou préviennent une contestation à naître en se consentant des concessions réciproques.

⁸⁶³ Ibid. Arbitrage

Contrat par lequel les parties terminent une contestation née ou préviennent une contestation à naître en se consentant des concessions réciproques.

⁸⁶⁴ Bernard-Ménoret, Ronan. « Les clauses de recours aux MARC : les pièges à éviter ». Les Petites Affiches, Petites affiches, n° 241 (3 décembre 2009): 20.

Tout d'abord, le recours à un MARC exclut le juge judiciaire et souvent le droit de la résolution du litige, qu'il s'agisse d'une conciliation ou d'un arbitrage en équité ou par amiable composition.

plus rapide et confidentielle que la justice ordinaire⁸⁶⁵, et potentiellement de meilleure qualité puisque les tiers participant à leur résolution peuvent être choisis sur la base de leur expertise et ainsi être plus compétents sur certaines questions que des juges classiques.⁸⁶⁶

319. MARC dans le contexte blockchain. Or la résolution des différends pouvant naître de l'exécution d'un contrat dans la *blockchain* nécessite une expertise et un procédé d'intervention particuliers, auxquels les parties seraient très avisées d'y recourir, si elles en ont la possibilité. En effet, en choisissant un tiers spécialement expert dans la *blockchain* pour régler leurs différends, les parties peuvent s'assurer que celui-ci sera bien mieux renseigné qu'un juge ordinaire et rendra en conséquence une décision potentiellement de meilleure qualité. Il pourra comprendre le fonctionnement d'un *smart contract*⁸⁶⁷ et être, par exemple, mieux à même de dire si son dysfonctionnement a été causé par un bogue fortuit ou une faute grave du concepteur⁸⁶⁸. En sus, avec la possibilité de prescrire dans la clause la manière dont il résoudra le conflit, elles s'assureront que le tiers interviendra selon le procédé le plus opportun compte tenu des spécificités de leur contrat intelligent. Par exemple, elles pourront préciser comment il devra s'y prendre pour halter un *smart contract* et décider du sort des actifs que ce dernier séquestre⁸⁶⁹. Les parties joindront ainsi le meilleur

⁸⁶⁵Jarrosson, Charles. « Les modes alternatifs de règlement des conflits. Présentation générale » §6. Revue internationale de droit comparé 49, n° 2 (1997): 325-45. <https://doi.org/10.3406/ridc.1997.5434>.

Le fait est que, dans de nombreuses matières, la réflexion sur les autres mode de règlement des conflits et leur pratique progresse (...). Les vertus qui leurs sont prêtées ne manquent pas d'attrait : souplesse, rapidité, économie et confidentialité, absence de juridisme inutile...

⁸⁶⁶ Flore Poloni et Thibaud Roujou de Boubée. « [Avis d'expert] Quatre raisons de privilégier l'arbitrage dans les litiges sur de nouvelles technologies », 6 juin 2021. <https://www.usinenouvelle.com/editorial/avis-d-expert-quatre-raisons-de-privilégier-l-arbitrage-dans-les-litiges-sur-de-nouvelles-technologies.N1099754>.

Les arbitres sont choisis par les parties et de ce fait, leur désignation se fait en fonction des spécificités du dossier et de l'expertise des arbitres candidats (par exemple en matière de développement de logiciels ou progiciels, de propriété intellectuelle, de protection des données, de déploiement d'ERP, etc.).

⁸⁶⁷ Lord Chancellor and Secretary of State for Justice. « Smart legal contracts - Advice to Government » p.148 §5.156, novembre 2021.

Disputes involving smart legal contracts can vary depending on, amongst other things, the type of smart legal contract, the underlying technology, the sophistication of the parties, and the factual background. The UKJT Rules appear particularly well-suited for disputes involving smart legal contracts. First, they make appropriate provision for the appointment of experts, which is particularly important in deciphering the meaning of coded terms.

⁸⁶⁸ V., *infra*, §278

⁸⁶⁹ Schmitz, Amy J., et Colin Rule. « Online Dispute Resolution for Smart contracts ». SSRN Scholarly Paper. Rochester, NY, 26 juin 2019. <https://papers.ssrn.com/abstract=3410450>.

In this way, the ODR clause in the smart contract can operate like an escrow arrangement. Instead of only two parties to the agreement, the inclusion of the ODR clause creates a role for a third party, the dispute resolution service provider. If either of the first two parties presses the Andon button, the role of the third party is automatically invoked.

des deux mondes en instituant un mécanisme légal de médiation et/ou d'arbitrage dans leur accord, adapté toutefois aux particularités d'une exécution dans la *blockchain* : un mécanisme de résolution des conflits *on* et *off-chain*⁸⁷⁰.

B – Contenu de la clause

320. Conditions de fond de la clause de MARC. En matière de médiation, les parties veilleront à ne pas rédiger la clause de sorte à obliger la partie consommatrice d'y recourir avant de pouvoir saisir un juge⁸⁷¹. Tandis qu'en matière d'arbitrage, la clause ne pourra pas être opposée à la partie consommatrice⁸⁷². En sus, les clauses de médiation et/ou d'arbitrage devront être rédigées de façon expresse et non équivoque dans le contrat, si les parties les souhaitent efficaces et opposables⁸⁷³.

321. Organisation de la procédure d'un MARC. L'organisation de la procédure de médiation ou d'arbitrage pourra être la suivante : lorsqu'un différend sur l'allocation d'actifs séquestrés surviendra entre les parties, celles-ci disposeront de la possibilité d'envoyer ces derniers dans un programme séquestre dans lequel les médiateurs et/ou arbitres désignés pourront actionner

⁸⁷⁰ Buchwald, Michael. « Smart contract Dispute Resolution: The Inescapable Flaws of Blockchain-Based Arbitration ». p.1420 §IV Optimal path forward. University of Pennsylvania Law Review 168, n° 5 (1 janvier 2020): 1369.

In the event that widespread, large-scale blockchain adoption were to occur, contracting parties would do well to build in contractual mechanisms to (1) reveal party identities and (2) migrate resolution off-chain...

⁸⁷¹ Article L.612-4 du code de la consommation : Est interdite toute clause ou convention obligeant le consommateur, en cas de litige, à recourir obligatoirement à une médiation préalablement à la saisine du juge.

⁸⁷² Article 2061 du code civil : La clause compromissoire doit avoir été acceptée par la partie à laquelle on l'oppose, à moins que celle-ci n'ait succédé aux droits et obligations de la partie qui l'a initialement acceptée. Lorsque l'une des parties n'a pas contracté dans le cadre de son activité professionnelle, la clause ne peut lui être opposée.

⁸⁷³ Pour la clause compromissoire : article 1443 du code de procédure civile : La clause compromissoire doit, à peine de nullité, être stipulée par écrit dans la convention principale ou dans un document auquel celle-ci se réfère.

Pour la clause de médiation :

Cour de cassation, civile, Chambre commerciale, 29 avril 2014, n°12-27.004, Publié au bulletin :

Attendu que la clause contractuelle prévoyant une tentative de règlement amiable, non assortie de conditions particulières de mise en oeuvre, ne constitue pas une procédure de conciliation obligatoire préalable à la saisine du juge, dont le non-respect caractérise une fin de non-recevoir s'imposant à celui-ci ;

une fonction transférant les actifs litigieux vers les personnes qu'elles auront choisies⁸⁷⁴.

Concernant la médiation, la clause décrira classiquement le processus à suivre pour tenter de trouver un accord. Dans le cas où il en serait trouvé, il sera stipulé que le médiateur pourra déclencher le transfert des actifs séquestrés. S'il n'est pas trouvé, les actifs continueront à rester en séquestre le temps que les arbitres ou le juge interviennent. Dans le cas d'un arbitrage, qui peut donc intervenir après la médiation, le mécanisme sera alors sensiblement le même : l'arbitre cherchera une solution, conformément au règlement d'arbitrage qui aura été convenu entre les parties⁸⁷⁵, puis pourra déclencher le transfert des actifs selon ce qu'il aura jugé.

Enfin, quel que soit le MARC choisi, les parties veilleront à préciser que la résolution des différends pourra demander des actions dépassant la simple réallocation des actifs séquestrés et que les parties s'engageront à respecter ces prescriptions *off-chain*⁸⁷⁶.

Les parties conviennent expressément qu'en cas de différend susceptible de naître du présent contrat, celles-ci s'efforceront de le régler de bonne foi par la voie de la médiation administrée par [...]. Si un accord satisfaisant n'est pas atteint dans les [...] jours suivant le début de la procédure de médiation, le litige relèvera d'une procédure d'arbitrage soumise au règlement de [...] auquel les parties adhèrent sans réserve.

§ II - Clauses attributive de juridiction et loi applicable

322. L'office du juge. Lorsque les parties ne peuvent recourir aux MARC ou qu'elles souhaitent soumettre la résolution de leurs différends à l'office d'un juge ordinaire, elles seront en

⁸⁷⁴ OpenLaw. « OpenCourt: Legally Enforceable Blockchain-Based Arbitration ». Medium, 19 octobre 2018. <https://media.consensys.net/opencourt-legally-enforceable-blockchain-based-arbitration-3d7147dbb56f>.

Once configured, OpenCourt will send the smart contract notice of a confirmed dispute once invoked. The smart contract will then transfer any identified digital assets to a virtual escrow account, thus locking these assets until an arbitral decision is reached.

⁸⁷⁵ Ceux-ci peuvent être les *Digital Dispute Resolution Rules* publiées par le *UK Jurisdiction Taskforce*.

Lawtech UK. « UKJT Digital Dispute Resolution Rules & Guidance », 8 mai 2021.

⁸⁷⁶ Lawtech UK. « UKJT Digital Dispute Resolution Rules & Guidance », 8 mai 2021.

4. Automatic dispute resolution. The outcome of any automatic dispute resolution process shall be legally binding on interested parties. 5. Submission to arbitration. Any dispute between interested parties arising out of the relevant contract or digital asset which was not the subject of an automatic dispute resolution process shall be submitted to arbitration in accordance with the version of these rules which is current at the time of submission; but any expert issue shall be determined by an appointed expert acting as such and not as an arbitrator.

tous cas avisées de prévoir une clause attributive de juridiction (A) et une clause de loi applicable (B) afin de neutraliser en amont les insécurités juridiques que peuvent faire naître la nature internationale des smart contracts (B).

A - Clause attributive de juridiction

323. Définition de la clause attributive de juridiction. Les clauses attributives de juridiction ont pour objet de désigner par avance le tribunal compétent en cas de survenance d'un litige entre des parties à un contrat⁸⁷⁷. Elles leur permettent ainsi de préciser les règles applicables en matière de procédure civile et/ou de les aménager, dans la mesure de ce qui est légalement possible, selon leur convenance⁸⁷⁸.

324. Clause attributive de juridiction dans notre contexte. Dans tous les cas où il ne sera pas possible pour les parties de régler leurs conflits par le biais de MARC⁸⁷⁹, elles devront passer par le juge et pourront être avisées, dès lors, de prévoir une clause attributive de juridiction. En effet, étant donné le caractère transnational⁸⁸⁰ des smart contracts, il sera opportun pour elles de s'assurer de la maîtrise de la compétence territoriale de la juridiction plutôt que de laisser faire le droit commun. Elles rédigeront donc une clause spécifiant quel tribunal sera compétent pour traiter les litiges naissant de leur contrat intelligent⁸⁸¹. En sus, la clause pourra contenir des indications spéciales pour le juge

⁸⁷⁷ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires – Clause attributive de juridiction - §I Objet et utilité - §125 Intérêts. . Les Intégrales. LGDJ, 2018.

La clause attributive de juridiction permet aux parties de déroger aux règles de compétence internationale en désignant, d'un commun accord, le juge compétent pour connaître d'un litige né ou à naître à l'occasion d'un rapport de droit déterminé.

⁸⁷⁸ Les parties peuvent déroger aux règles des articles 46 et suivants du code de procédure civile seulement si les clauses ont été stipulées de façon très apparentes et que les personnes ont contracté en qualité de commerçant.

Article 48 du code de procédure civile : *Toute clause qui, directement ou indirectement, déroge aux règles de compétence territoriale est réputée non écrite à moins qu'elle n'ait été convenue entre des personnes ayant toutes contracté en qualité de commerçant et qu'elle n'ait été spécifiée de façon très apparente dans l'engagement de la partie à qui elle est opposée.*

⁸⁷⁹ V., *infra*, §320

⁸⁸⁰ Jeffrey D. Neuburger, Wai L. Choy, et Kevin P. Milewski. « Smart contracts: Best Practices », Thomson Reuters - Practical Law, 2019, 11-19.

Specifying governing law, jurisdiction, and venue in a traditional contract will bypass the issue of deciphering which jurisdiction applies to judicial interpretation of smart contracts, which could become complicated given the borderless nature of blockchain.

⁸⁸¹ Mustapha Mekki. « Blockchain : l'exemple des smart contracts, Entre innovation et précaution » §31 – p.13, 2018. <https://lesconferences.openum.ca/files/sites/97/2018/05/Smart-contracts.pdf>.

lui indiquant comment il pourra intervenir dans le *smart contract*. Ainsi les parties pourront lui prévoir un accès spécial afin qu'il puisse actionner une fonction transférant des actifs séquestrés en attente de la résolution d'un litige.⁸⁸²

*Dans le cas où des différends viendraient à naître à propos de la validité, de l'interprétation, de l'exécution ou de l'inexécution, de l'interruption ou de la résiliation du présent contrat intelligent, les parties soumettront le litige au tribunal de [...].*⁸⁸³

Le juge en charge de la résolution du litige survenu entre les parties disposera de la faculté d'interrompre le Smart contract et d'éventuellement décider de la réallocation des actifs séquestrés par celui-ci selon des modalités précisés dans l'annexe technique du présent contrat.

B - Clause de loi applicable

325. Définition de la clause de loi applicable. La clause de loi applicable a pour but de permettre aux parties de choisir la loi qu'elles veulent voir appliquer à leur contrat, et qui sera donc suivie par le juge pour la résolution de leurs litiges⁸⁸⁴. À défaut d'une telle clause, c'est l'article 4 du règlement Rome I qui permet de déterminer la législation applicable au contrat⁸⁸⁵.

Pour éviter toute discussion et tout contentieux inutile, il faut encourager la rédaction de clauses relatives au droit applicable et des clauses désignant la juridiction compétente.

⁸⁸² Lord Chancellor and Secretary of State for Justice. « Smart legal contract, Advice To Government » §5.143 p.143, novembre 2021.

In response to the call for evidence, Herbert Smith Freehills said that the paper could usefully analyse the “jurisdictional basis for any enforcement and/or interim relief activities” undertaken by the courts of England and Wales in relation to smart legal contracts. For example, what powers (if any) may the court have to suspend the operation of a piece of code pending the final determination of a dispute?

⁸⁸³ Collectif. « Dictionnaire Permanent Droit des affaires - Contrat de vente ». ELNET, juin 2023.

⁸⁸⁴ Frédéric BUY, Marie LAMOUREUX, Jacques MESTRE, et Jean-Christophe RODA. Les principales clauses des contrats d'affaires – Clause de droit applicable - §1 – Objet et utilité - §550. Les Intégrales. LGDJ, 2018.

L'insertion d'une clause de droit applicable au sein d'un contrat international permet aux parties de choisir la loi qui régit ce contrat. L'intérêt d'une telle clause est multiple. (...) si les parties n'ont pas désigné la loi applicable, la lex contractus est déterminée par une règle de conflit de lois subsidiaire qui énonce un rattachement objectif afin de localiser le contrat dans un ordre juridique étatique.

⁸⁸⁵ Règlement (CE) n° 593/2008 du Parlement européen et du Conseil du 17 juin 2008 sur la loi applicable aux obligations contractuelles (Rome I), 177 OJ L § (2008). Article 4. <http://data.europa.eu/eli/reg/2008/593/oj/fra>.

À défaut de choix exercé conformément à l'article 3 et sans préjudice des articles 5 à 8, la loi applicable au contrat suivant est déterminée comme suit: ...

326. Contenu de la clause de loi applicable. Comme déjà évoqué, la nature du *smart contract* peut soulever des questions sur le corps de loi applicable au contrat qui l'utilise pour son exécution⁸⁸⁶. Dans le but d'éviter des discussions inutiles, les parties à un contrat intelligent pourront donc prévoir une clause de leur accord réglant sans ambiguïté la question. L'article 3 du règlement Rome I⁸⁸⁷ leur octroie une grande liberté de choix bien qu'elle reste bornée par la prohibition de la fraude : les parties ne pourront faire élection d'une loi qui leur permettrait de ne pas être soumises à ce qui serait normalement applicable à leur contrat⁸⁸⁸.

Ainsi l'article 3 du règlement Rome I prévoit que *Lorsque tous les autres éléments de la situation sont localisés, au moment de ce choix, dans un pays autre que celui dont la loi est choisie, le choix des parties ne porte pas atteinte à l'application des dispositions auxquelles la loi de cet autre pays ne permet pas de déroger par accord*. Autrement dit, les parties ne pourront user de leur liberté pour échapper aux règles d'ordre public interne. Il en est de même pour les lois dites de police⁸⁸⁹ ; selon l'article 9 du règlement Rome I *Les dispositions du présent règlement ne pourront porter atteinte à l'application des lois de police du juge saisi*. Le juge français pourra forcer l'application d'une loi de police nationale, nonobstant la désignation d'une loi étrangère par les parties pour régir leur contrat⁸⁹⁰.

⁸⁸⁶ V., *infra*, §23

⁸⁸⁷ Ibid. Article 3

Le contrat est régi par la loi choisie par les parties. Le choix est exprès ou résulte de façon certaine des dispositions du contrat ou des circonstances de la cause. Par ce choix, les parties peuvent désigner la loi applicable à la totalité ou à une partie seulement de leur contrat.

⁸⁸⁸ Aliénor FEVRE. « La détermination de la loi applicable dans les contrats ». CMS Francis LEFEBVRE (blog). Consulté le 12 juin 2023. <https://cms.law/fr/fra/news-information/la-determination-de-la-loi-applicable-dans-les-contrats>.

Cependant, cette grande liberté ne doit pas permettre aux parties d'écarter des règles de droit qui devraient normalement s'appliquer à elles et d'opérer ainsi une fraude à la loi. Le choix des parties est considéré comme frauduleux si, par ce choix, elles essayent de se soustraire à des dispositions impératives d'une loi qui serait naturellement applicable à leur contrat.

⁸⁸⁹ Règlement (CE) n° 593/2008 du Parlement européen et du Conseil du 17 juin 2008 sur la loi applicable aux obligations contractuelles (Rome I), 177 OJ L § (2008). Article 9. <http://data.europa.eu/eli/reg/2008/593/oj/fra>.

Une loi de police est une disposition impérative dont le respect est jugé crucial par un pays pour la sauvegarde de ses intérêts publics, tels que son organisation politique, sociale ou économique, au point d'en exiger l'application à toute situation entrant dans son champ d'application, quelle que soit par ailleurs la loi applicable au contrat d'après le présent règlement.

⁸⁹⁰ Aliénor FEVRE. « La détermination de la loi applicable dans les contrats ». CMS Francis LEFEBVRE (blog). Consulté le 12 juin 2023. <https://cms.law/fr/fra/news-information/la-determination-de-la-loi-applicable-dans-les-contrats>.

Par la voie de ce mécanisme dérogatoire, le juge peut donc écarter certaines dispositions de la loi choisie par les parties. A titre d'exemple, la loi n° 75-1334 du 31 décembre 1975 relative à la sous-traitance, en ses dispositions protectrices du sous-traitant, a été qualifiée de loi de police (Cass. mixte, 30 novembre 2007, n° 06-14.006, arrêt Agintis). Dès lors, si

Enfin, l'article 21 du règlement Rome I dispose que la liberté de choix des parties sera également mise en échec par des dispositions d'ordre public international⁸⁹¹.

327. Conclusion de la section II et du chapitre III. Les dernières clauses du document *fiat* d'un contrat intelligent seront donc consacrées à l'interruption du contrat et la résolution des conflits issus de celui-ci. S'agissant du traitement des litiges, les parties seront avisées, toutes les fois qu'elles le peuvent, de recourir aux MARC, afin de bénéficier d'une solution spécialement experte et adaptée à la résolution de leurs différends. Dans les cas où elles ne voudraient ou ne pourraient pas recourir à ces modes, il leur sera recommandé de prévoir des clauses attributives de juridiction et de loi applicable afin de régler en amont toutes les incertitudes pouvant découler de la nature internationale du *smart contract* exécutant la convention.

un juge français est saisi d'un litige international présentant un lien de rattachement avec la France (notamment si l'ouvrage en question est situé en France), il devra faire application de ces dispositions quand bien même les parties auraient valablement désigné une autre loi pour régir leur contrat.

⁸⁹¹ Règlement (CE) n° 593/2008 du Parlement européen et du Conseil du 17 juin 2008 sur la loi applicable aux obligations contractuelles (Rome I), 177 OJ L § (2008). Article 21. <http://data.europa.eu/eli/reg/2008/593/oj/fra>.

L'application d'une disposition de la loi désignée par le présent règlement ne peut être écartée que si cette application est manifestement incompatible avec l'ordre public du for. La liberté de choix des parties est quasi-totale. Cependant, cette grande liberté ne doit pas permettre aux parties d'écarter des règles de droit qui devraient normalement s'appliquer à elles et d'opérer ainsi une fraude à la loi. Le choix des parties est considéré comme frauduleux si, par ce choix, elles essayent de se soustraire à des dispositions impératives d'une loi qui serait naturellement applicable à leur contrat. Il existe trois mécanismes en droit international privé qui permettent de contrer toute tentative de fraude : les dispositions d'ordre public interne, les lois de police et l'ordre public international.

Titre II – Développement technique du contrat intelligent

328. **Aspects essentiels de l'implémentation technique.** Une fois le contrat *fiat* rédigé, les parties pourront passer à l'étape de l'implémentation technique du contrat intelligent. Comme mentionné, cette phase peut abriter de nombreuses tâches⁸⁹² ; mais nous avons fait le choix de nous focaliser uniquement sur deux aspects essentiels de l'élaboration technique d'un contrat intelligent : la détermination de la *blockchain* sur laquelle seront déployés les smart contracts (Chapitre I) et le développement de ces derniers chargés d'exécuter les stipulations du contrat *fiat* (Chapitre II).

⁸⁹² V., *infra*, §232

Chapitre I – La détermination de la *blockchain* comme infrastructure d'exécution

329. Les infrastructures d'exécution. Quelles sont les blockchains, ou plus exactement les infrastructures d'exécution⁸⁹³, qui conviennent le mieux pour des parties à un contrat intelligent ? Nous proposons de passer en revue plusieurs d'entre elles, que nous avons sélectionnées pour leur adéquation à supporter l'exécution des contrats intelligents (Section I) ; avant de recommander et justifier celles sur lesquelles le choix des parties devra porter en fonction de la spécificité des contrats intelligents qu'elles souhaitent implémenter (Section II)⁸⁹⁴.

Section I – Les différentes infrastructures

330. Critères de sélection de la *blockchain*. Il existe un nombre pléthorique de *blockchain* sur lesquelles des parties peuvent choisir d'exécuter leurs contrats intelligents. Nous avons sélectionné celles que nous estimons être les meilleures, en nous basant sur certains critères :

- l'ancienneté ; qui est gage de nombreuses qualités pouvant intéresser des parties engagées dans une démarche de création d'un contrat intelligent. L'ancienneté peut en effet présumer que la *blockchain* sera pérenne⁸⁹⁵ et bénéficiera d'une technologie mature ;
- le caractère *turing complet*. Cela signifie l'aptitude pour les blockchains à pouvoir accueillir des programmes pouvant en théorie réaliser n'importe quelle tâche informatique⁸⁹⁶. Cela

⁸⁹³ Nous utilisons ce terme car nous verrons tantôt que les infrastructures sur lesquelles peuvent être déployés des smart contracts ne sont pas nécessairement des blockchains au sens strict.

⁸⁹⁴ Eric A. Caprioli. « Mythes et légendes de la blockchain face à la pratique », Dalloz IP/IT 2019. 429.

III - Détermination du type de blockchain à utiliser - Pour déterminer de quel type de blockchain on a besoin, il est fondamental de se poser une série de questions avec une méthodologie bien précise fondée sur une arborescence en partant du cas d'usage mentionné ci-dessus.

⁸⁹⁵ Il est difficile d'anticiper sur l'existence future d'une *blockchain* (qui est une propriété fondamentale pour des parties dans notre contexte) mais un élément pouvant permettre de la présumer est son ancienneté. Si une *blockchain* existe et fonctionne depuis un certain nombre d'années, on peut gager qu'elle a amassé un certain effet de réseau garantissant qu'elle sera là encore quelque temps.

⁸⁹⁶ Yaga, Dylan J., Peter M. Mell, Nik Roby, et Karen Scarfone. « Blockchain Technology Overview ». NIST, 3 octobre 2018. <https://www.nist.gov/publications/blockchain-technology-overview>.

exclut des blockchains mono-usage comme *Bitcoin* qui ne servent, en principe, qu'à transférer des crypto-monnaies, et où il est impossible ou très difficile de déployer des programmes pouvant exécuter les clauses d'un contrat⁸⁹⁷ ;

- la maturité et le développement de leur écosystème. Une *blockchain* est dans une large mesure une infrastructure sociale⁸⁹⁸. Plus celle-ci bénéficie d'une communauté propre de nombreux utilisateurs et de développeurs, plus elle est un environnement attrayant pour y faire héberger ses smart contracts. Ses usagers pourront, en effet, s'appuyer sur des nombreuses applications et outils de développement, de la documentation, des forums et des chats pour produire de manière plus commode et sécurisée leurs programmes.

Ces infrastructures sélectionnées seront alors évaluées sur ces critères :

- leur résilience : si les parties font le choix de recourir à une *blockchain*, c'est dans le but de s'approprier ses propositions de valeurs principales. Or une des plus cardinales d'entre elles est sa capacité à constituer une infrastructure particulièrement résiliente et disponible pour héberger des programmes⁸⁹⁹. En exécutant leur contrat sur une *blockchain*, les parties doivent être assurées que celle-ci fonctionnera contre presque toutes circonstances ;
- la sécurité de leurs smart contracts : nous évaluerons également les blockchains sur tous les

Turing complete : A system (computer system, programming language, etc.) that can be used for any algorithm, regardless of complexity, to find a solution.

⁸⁹⁷ De Filippi, Primavera, et Aaron Wright. *Blockchain and the Law: The Rule of Code*. Emplacement 527. Harvard University Press, 2018.

Bitcoin excelled as platform to facilitate the exchange of digital currency, but without updating the underlying protocol, it could not be used for much more.

⁸⁹⁸ Voir la notion de *social layer* que de nombreuses analystes avancent pour exprimer le fait qu'une *blockchain* est en fait sécurisée en dernier ressort par sa communauté de parties prenantes.

Moore, Galen. « The Social Layer Is Ironically Key to Bitcoin's Security ». TechCrunch (blog), 19 janvier 2019. <https://techcrunch.com/2019/01/19/bitcoin-social-layer/>.

The story goes: in the event of a security failure, Bitcoin's community of developers, investors, miners and users are an ultimate layer of defense. We, Bitcoin's community, have the option to fork the protocol—to port our investment of time, capital and computing power onto a new version of Bitcoin. It's our collective commitment to a trust-minimized monetary system that makes Bitcoin strong.

⁸⁹⁹ Alexandre Stachtchenko. « Blockchain, résilience et souveraineté : tribune dans Investir ». Investir, 11 juillet 2020. <https://blockchainpartner.fr/2020/07/16/blockchain-resilience-et-souverainete/>.

La blockchain par essence est à qualifier de système plutôt résilient qu'efficient. (...) Mais si un nœud (serveur) tombe, aucun problème sérieux ne se pose pour le réseau. La base de données est en effet répliquée en des dizaines de milliers d'endroits. C'est l'incarnation d'un système décentralisé.

éléments au sein de leur écosystème qui concourront à faire diminuer le risque d'apparition de bogues et portes dérobées dans le développement des smart contracts. Cela peut comprendre les caractéristiques du langage de programmation, la largesse communauté de développeurs, et les pratiques et des outils de développement ;

- leur commodité : nous entendons l'ensemble des outils et applications à disposition des parties leur permettant de produire efficacement un programme exécutant leur contrat. Le recours à la *blockchain* pour exécuter un contrat ne doit, en effet, pas être beaucoup plus lourd à mettre en œuvre qu'une solution centralisée ou même qu'une solution « non-automatisée »⁹⁰⁰. Les parties doivent trouver dans son écosystème une myriade d'éléments leur permettent de créer rapidement un *smart contract* qui exécutera leur convention de manière optimale ;
- leur performance : il est courant que les blockchains offrent leurs atouts uniques en l'échange de performances suboptimales par rapport à ce qui est possible *off-chain*⁹⁰¹. C'est-à-dire que ses programmes s'exécutent de manière plus lente et coûteuse que des programmes ordinaires. Cependant les parties doivent tout de même privilégier les infrastructures dites mises à l'échelle, soit des infrastructures proposant des performances s'approchant de serveurs centralisés et capables d'exécuter des programmes rapidement pour un coût nul ou très peu élevé⁹⁰² ;
- leur personnalisation : les parties doivent enfin choisir les infrastructures suffisamment polyvalentes pour pouvoir implémenter toutes les spécificités d'un projet de contrat intelligent. L'exécution de certains contrats peut en effet nécessiter une grande plasticité de

⁹⁰⁰ V., *infra*, §31

⁹⁰¹ Alexandre Stachtchenko. « Blockchain, résilience et souveraineté : tribune dans Investir ». Investir, 11 juillet 2020. <https://blockchainpartner.fr/2020/07/16/blockchain-resilience-et-souverainete/> .

Ces dernières années, les technologies blockchain se sont heurtées à ce modèle de pensée dominant favorisant l'efficacité. Le premier réflexe des professionnels découvrant ces technologies a été de chercher à faire rentrer un carré dans un rond : « d'accord pour la blockchain, mais si possible dans mon propre serveur, en plus rapide, et moins énergivore ».

⁹⁰² « Blockchain Scalability Approaches | Chainlink ». Consulté le 28 juin 2023. <https://chain.link/education-hub/blockchain-scalability>.

Blockchain scalability is the ability of a blockchain to process transactions, store data, and reach consensus as additional users are added to the network.

l'infrastructure d'accueil des programmes.

Nous proposons de classer ces infrastructures entre celles que nous nommons les généralistes (§1) et les spécialisées (§2). Les premières sont des blockchains destinées à accueillir tout type de *smart contract*, et correspondent aux infrastructures habituelles où ils sont déployés⁹⁰³. Les secondes, qui se développent depuis plus récemment, sont des blockchains que nous nommerons *ad-hoc*, conçues pour accueillir une application unique⁹⁰⁴.

§ I – Les infrastructures généralistes

331. Les différentes infrastructures d'exécutions généralistes. Au sein des infrastructures d'exécution généralistes, il est encore possible de faire la distinction entre les blockchains dites de première couche (les *layer 1* ou L1) (A) et celles de secondes couches (*layer 2* ou L2) (B). Les L1 généralistes sont des blockchains classiques telles qu'on se les représente couramment (comme Bitcoin ou Ethereum). Pour améliorer leur performance, des infrastructures spéciales connectées à ces L1 ont été construites ; qu'on nomme les L2⁹⁰⁵.

⁹⁰³ Staff, Block Telegraph. « General-Purpose vs App-Specific L1s - Block Telegraph », 7 septembre 2022. <https://blocktelegraph.io/general-purpose-vs-app-specific-l1s/>, <https://blocktelegraph.io/general-purpose-vs-app-specific-l1s/>.

General-purpose L1s are the Swiss Army Knives of Web3 infrastructure. They are blockchains that are not optimized for any specific application, instead, they allow a variety of decentralized applications to be built on top of them. Examples of general-purpose L1s include Ethereum, Solana, Avalanche, and BSC. These chains allow protocols to share (and compete) for users, liquidity, and blockspace.

⁹⁰⁴ Staff, Block Telegraph. « General-Purpose vs App-Specific L1s - Block Telegraph », 7 septembre 2022. <https://blocktelegraph.io/general-purpose-vs-app-specific-l1s/>, <https://blocktelegraph.io/general-purpose-vs-app-specific-l1s/>.

App-specific L1s are blockchains that have been developed exclusively for one or a few decentralized applications. These L1s can customize every aspect of their tech stack, such as their programming language, development frameworks, and consensus mechanisms to best suit their protocol(s) needs. Cosmos Zones are perfect examples of app-specific L1s. These chains are fully customizable and are completely sovereign within their own chains.

⁹⁰⁵ Ethereum.org. « Layer 2 ». Consulté le 28 juin 2023. <https://ethereum.org>.

Generalized layer 2s behave just like Ethereum — but cheaper. Anything that you can do on Ethereum layer 1, you can also do on layer 2. Many dapps have already begun to migrate to these networks or have skipped Mainnet altogether to deploy straight on a layer 2.

A – Les premières couches (L1)

a) Ethereum

332. Présentation de Ethereum. Naturellement, le premier L1 à présenter est Ethereum, la *blockchain turing complète* la plus ancienne et connue. La machine virtuelle exécutant ses smart contracts se nomme l'*Ethereum Virtual Machine* (EVM) et son langage de programmation principal est *Solidity*. Son mécanisme de consensus est la preuve d'enjeu : un mécanisme peu énergivore permettant la validation des blocs de transactions par des individus sélectionnés en fonction de la quantité de jetons qu'ils mettent en jeu dans le protocole⁹⁰⁶.

333. Résilience de Etherum. Ethereum tire de son ancienneté un nombre élevé de nœuds et validateurs participant à son mécanisme de consensus⁹⁰⁷. Cette caractéristique fait d'elle une des deux *blockchain* les plus décentralisées au monde⁹⁰⁸. Or plus une *blockchain* est décentralisée, moins elle dispose de points centraux de faillibilité : ses nombreux validateurs et nœuds mutualisent sa disponibilité et son inaltérabilité. Ces éléments font donc d'Ethereum un environnement excellent stable et pérenne pour déployer des smart contracts. Néanmoins son processus de gouvernance peut relativiser sa stabilité. En effet, l'adoption des évolutions techniques dans le protocole (par *softfork* ou *hardfork*) se fait par le *rough consensus*⁹⁰⁹ : c'est-à-dire qu'une fonctionnalité est ajoutée ou supprimée à la *blockchain* qu'après qu'un corps inorganisé de développeurs ait estimé quel était le sentiment général de l'ensemble de la communauté sur celle-ci⁹¹⁰. Il s'agit d'un processus de

⁹⁰⁶ Ibid. « Proof-of-Stake (PoS) ». Consulté le 28 juin 2023. <https://ethereum.org>.

Ethereum uses proof-of-stake, where validators explicitly stake capital in the form of ETH into a smart contract on Ethereum. This staked ETH then acts as collateral that can be destroyed if the validator behaves dishonestly or lazily.

⁹⁰⁷ Il est compté plus de 800 milles validateurs au moment où sont écrites ces lignes.

Etherscan Beacon Chain (Phase 0) Ethereum 2.0 Explorer. « Statistics - Validators | Mainnet Beacon Chain (Phase 0) Ethereum 2.0 Explorer ». Consulté le 28 juin 2023. <https://beaconscan.com/stat/validator>.

⁹⁰⁸ Drew Mailen. « Why Ethereum Is More Decentralized After the Merge ». Blockworks, 7 novembre 2022, sect. Education, Sponsored. <https://blockworks.co/news/ethereum-decentralization-after-the-merge>.

Decentralization in POS vs. POW. One of the fairest litmus tests to test Ethereum's decentralization is to compare it to the proof-of-work consensus mechanism that Bitcoin uses. Looking at Lido once again, after Lido's validators are accounted for, it requires more collusion than Bitcoin's PoW.

⁹⁰⁹ « Rough Consensus ». In Wikipedia, 29 juin 2023. https://en.wikipedia.org/w/index.php?title=Rough_consensus&oldid=1162457456.

Rough consensus is a term used in consensus decision-making to indicate the "sense of the group" concerning a particular matter under consideration. It has been defined as the "dominant view" of a group as determined by its chairperson.

⁹¹⁰ « How Does Governance Work in Ethereum | How Do Bitcoin and Crypto Work ? » Consulté le 29 juin 2023.

gouvernance assez informel et imprévisible alors que les modifications qu'il engendre peuvent avoir des incidences importantes sur les smart contracts déployés⁹¹¹. Une absence de visibilité sur ces potentielles modifications nuance donc la stabilité du protocole malgré sa grande décentralisation.

334. Sécurité et commodité dans Ethereum. L'autre grand avantage de Ethereum, lui aussi tiré de sa relative vieillesse, est la richesse de son écosystème qui est elle-même procuratrice de sécurité et commodité pour ses développeurs. Etant la première *blockchain* à accueillir des smart contracts, elle a su créer un large effet de réseau⁹¹² autour d'elle qui se manifeste par :

- un outillage fourni et divers pour le développement de *smart contract* (une multitude de logiciels de développements, de tests et de déploiements),
- un langage de programmation vivant, très utilisé, sur lequel s'est formé une grande expertise,
- de nombreuses bibliothèques *open source*,
- de nombreux standards de développement,
- des forums et des groupes de chat actifs.

Autant d'éléments qui non seulement facilitent la vie des développeurs en leur permettant d'être plus efficaces dans leurs tâches, mais aussi sécurisent leurs applicatifs car ils se basent sur des outils

<https://www.bitcoin.com/get-started/how-does-governance-work-in-ethereum/>.

Ethereum integrates a formalized process for proposing, debating, and integrating upgrades to its protocol. At the core of this process lies the Ethereum Improvement Proposal (EIP). Broadly speaking, individuals or teams within the Ethereum developer community draft EIPs and the wider community debates their merits vigorously. Proposals are amended, resubmitted, and debated further until rough consensus is achieved amongst the most active participants in the community.

⁹¹¹ Par exemple le *hardfork* Istanbul a intégré une modification du protocole qui a eu pour effet d'augmenter les frais d'interaction avec certaines fonctions de *smart contract*.

H, Renaud. « Ethereum : le hard fork Istanbul, qu'est-ce que ça change ? » Journal du Coin, 7 décembre 2019. <https://journalducoin.com/ethereum/ethereum-hard-fork-istanbul-definition/>.

EIP 1884 : MODIFICATION DU PRIX EN GAS DE CERTAINES FONCTIONS. Dans un smart contract, chaque action réalisée (création d'une transaction, modification d'une information, etc.) a un coût en gas. L'EIP 1884 vise à rétablir un équilibre raisonnable entre le prix en gas de certaines actions (liées aux fonctions de ces contrats) et la puissance de calcul nécessaire à la réalisation de ces actions dans l'EVM. D'un point de vue plus technique, le coût en gas sera augmenté pour les actions SLOAD, BALANCE et EXTCODEHASH : en effet, elles sont peu chères pour le moment, alors qu'elles consomment beaucoup de puissance de calcul.

⁹¹² Cointelegraph. « What Is the Network Effect? », 19 février 2023. <https://cointelegraph.com/news/what-is-the-network-effect>.

The network effect is a phenomenon where the value of a product or service increases as more people use it. As the user base grows, there are more opportunities for interactions, which can lead to increased benefits and positive outcomes for each user. The network effect is a powerful driver of growth and adoption for many technologies and platforms, including social media, messaging apps and marketplaces.

nombreux, éprouvés et audités par une large communauté. La maturité de l'écosystème se manifeste aussi par les nombreuses applications déployées dans la *blockchain* auxquels des smart contracts peuvent se connecter afin d'exploiter leurs fonctionnalités. Par exemple, c'est sur Ethereum qu'on trouve le plus grand nombre de jetons stables, d'oracles ou encore de protocoles de finance décentralisés. Ce sont autant d'applications utiles que des développeurs pourront mobiliser dans leurs projets.

335. Performance de Ethereum. Le revers de l'ancienneté de Ethereum est le retard technique qu'il accuse sur d'autres L1 sortis plus récemment. Dans le trilemme des blockchainns posé par Vitalik Buterin, ce dernier évoque que Ethereum a compromis sur la performance afin de pouvoir être décentralisé et sécurisé⁹¹³. Cela se manifeste par le fait que le déploiement et les interactions avec ses smart contracts prennent un certain temps et coûtent chers. La raison est que la *blockchain* dispose d'un mécanisme la faisant ralentir et augmenter ses frais lorsqu'elle est trop sollicitée (afin de neutraliser les attaques de déni de service)⁹¹⁴. Or comme Ethereum est extrêmement populaire, la demande est constamment élevée.

336. Personnalisation de Ethereum. Ces limitations techniques empêchent donc ce L1 de supporter n'importe quel type de programme : les applications gourmandes en ressources seront à proscrire. De manière générale, les programmes hériteront, pour le pire comme pour le meilleur, des propriétés de Ethereum : ils seront immuables, transparents et accessibles librement mais lents et chers à actionner.

b) Tezos

337. Présentation de Tezos. Tezos est un L1 *turing complet* dit de *troisième génération* : c'est-à-dire sorti quelques années après Ethereum, et constituant une version prétendument exempt

⁹¹³ Vitalik Buterin. « Why sharding is great: demystifying the technical properties », 7 avril 2021. <https://vitalik.ca/general/2021/04/07/sharding.html>.

The scalability trilemma says that there are three properties that a blockchain try to have, and that, if you stick to "simple" techniques, you can only get two of those three. The three properties are: Scalability (...) ; Decentralization (...) Security (...). Traditional blockchains - including Bitcoin, pre-PoS/sharding Ethereum, Litecoin, and other similar chains. These rely on every participant running a full node that verifies every transaction, and so they have decentralization and security, but not scalability.

⁹¹⁴ Ethereum.org. « Gas and Fees ». Consulté le 29 juin 2023. <https://ethereum.org>.

In short, gas fees help keep the Ethereum network secure. By requiring a fee for every computation executed on the network, we prevent bad actors from spamming the network (...). High gas fees are due to the popularity of Ethereum. Performing any operation on Ethereum requires consuming gas, and gas space is limited per block (...); If there's too much demand, users must offer a higher tip amount to try and outbid other users' transactions. A higher tip can make it more likely that your transaction will get into the next block.

de ses défauts⁹¹⁵. Cette *blockchain* a la caractéristique d’avoir été cocréée par un français et bénéficie à ce titre d’une grande communauté de développeurs en France⁹¹⁶. Elle utilise une variante du mécanisme de consensus de preuve d’enjeu, appelée *preuve d’enjeu liquide*, et le langage de programmation de ses smart contracts est *Michelson*.

338. Résilience de Tezos. En dépit de son ancienneté, Tezos n’est pas une *blockchain* très décentralisée comparée à d’autres L1 pourtant plus récents. Cela pourrait s’expliquer par le fait que son mécanisme de consensus favorise la concentration et plus généralement son manque de popularité par rapport à d’autres blockchains (dont nous donnerons des éléments d’explication plus bas). Actuellement, le protocole compte environ 400 validateurs (appelés *bakers*⁹¹⁷), ce qui est extrêmement peu par rapport à Ethereum mais toujours bien plus décentralisé que des solutions privées et centralisées. La *blockchain* est donc un environnement assez faiblement pérenne pour déployer des smart contracts, comparés à d’autres L1.

Tezos a néanmoins pour particularité d’avoir une gouvernance de son protocole *on-chain* : l’évolution de sa *blockchain* est décidée par vote de ses *bakers*⁹¹⁸. La transformation du protocole ne fait ainsi pas recours au *rough consensus*. Celui-ci est amendé si une proposition d’évolution arrive au terme d’un processus formalisé et déterminé. Une telle gouvernance rend plus prévisible l’évolution de la *blockchain* pour les développeurs, ce qui est, dans une certaine mesure, gage de stabilité et de pérennité.

339. Sécurité et commodité de Tezos. Un autre atout de Tezos est la sécurité fournie par sa machine virtuelle et son langage d’écriture de *smart contract*, *Michelson*. Il s’agit d’un langage de

⁹¹⁵ « Exaion, EDF Group Subsidiary, Becomes a Tezos Baker », 21 juin 2022. <https://exaion.edf.fr/en/exaion/our-news/exaion-edf-group-subsidiary-becomes-a-tezos-baker>.

Tezos is a third-generation decentralized blockchain dedicated to the creation and management of assets and distributed applications.

⁹¹⁶ Ingrid Vergara. « Tezos, la blockchain aux racines françaises qui veut concurrencer Ethereum ». LEFIGARO, 9 décembre 2020, sect. Tech & Web. <https://www.lefigaro.fr/secteur/high-tech/tezos-la-blockchain-aux-racines-francaises-qui-veut-concurrencer-ethereum-20201209>.

Cofondé par le Français Arthur Breitman, développé avec l'aide d'une équipe de chercheurs dans l'hexagone, Tezos permet d'échanger de la valeur et de créer des applications décentralisées.

⁹¹⁷ Data, Blockwatch. « Bakers Tezos sur TzStats ». Consulté le 30 juin 2023. <https://tzstats.com/bakers>.

⁹¹⁸ Arluck, Jacob. « Amending Tezos ». Tezos (blog), 13 mai 2020. <https://medium.com/tezos/amending-tezos-b77949d97e1e>.

Tezos is a self-amending blockchain network which incorporates a formal, on-chain mechanism for proposing, selecting, testing, and activating protocol upgrades without the need to hard fork.

programmation fonctionnel sur lequel peut être réalisé de la vérification formelle : soit un moyen de prouver mathématiquement qu'un programme est dépourvu de bugs. Cela confère, en théorie, aux smart contracts une sécurité bien meilleure que celle fournie par des outils de tests traditionnels⁹¹⁹.

Mais le langage souffre d'une difficile prise en main dû à son originalité. Il requiert une courbe d'apprentissage plus longue que *solidity* ; ce qui a constitué probablement une des causes pour lesquelles la *blockchain* manque de popularité. Nous verrons que beaucoup de L1 ont fait le choix de copier l'EVM afin de permettre aux nombreux développeurs de l'écosystème Ethereum de transiter facilement vers le leur. A cet égard, il peut être affirmé que Tezos a quelque peu compromis sur la commodité en faveur de la sécurité. La communauté de développeurs de Tezos a su toutefois développer des librairies permettant d'écrire des smart contracts dans des langages plus commodes (comme python⁹²⁰) pouvant ensuite être transpilés en *Michelson*.

Enfin, bien que incomparable avec l'effet de réseau et l'écosystème de Ethereum, Tezos figure parmi les L1 les plus anciens et tire aussi, dans une certaine mesure, bénéfice de sa maturité :

- on y compte un certain nombre de librairies de *smart contract* ;
- il y existe des nombreux standards pour créer différents types de *smart contract* (jetons fongibles et non-fongibles..) ;
- il y est déployé un assez grand nombre d'applications que des développeurs peuvent exploiter pour enrichir les fonctionnalités de leurs programmes ;
- de manière générale, Tezos bénéficie d'une assez large communauté de développeurs dévouée depuis un certain temps à améliorer les pratiques de développement et de sécurité dans son écosystème.

340. Performance de Tezos. Tezos est mieux mis à l'échelle que Ethereum. Son mécanisme de consensus fonctionne de sorte à réduire le nombre de validateurs nécessaires pour valider les blocs de transactions, ce qui centralise mais accélère le processus. Le fait également que la *blockchain* ne soit pas surutilisée rend le déploiement et l'interaction avec un *smart contract* bien

⁹¹⁹ Hillard, Frank. « Tezos - Vérification formelle ». OCTO Talks !, 4 septembre 2020. <https://blog.octo.com/tezos-verification-formelle/>.

La blockchain Tezos apporte plusieurs innovations notamment en rendant possible de la vérification formelle. La blockchain Tezos utilise un langage de smart contract (Michelson) qui a été formellement prouvé. Cette preuve du langage Michelson est une librairie appelée Mi-cho-coq. En s'appuyant sur l'isomorphisme Curry-Howard, qui assure la correspondance entre un programme et un théorème), on peut traduire un script Michelson en une forme logique équivalente (un théorème).

⁹²⁰ « SmartPy | Tezos ». Consulté le 30 juin 2023. <https://tezos.com/developers/smartpy/tezos.com/developers/smartpy>.

moins chers et plus rapides que sur Ethereum. Enfin l'intégration de solutions de mises à l'échelle, notamment les *rollup* que nous aborderons plus bas⁹²¹, promet d'améliorer considérablement la performance du protocole⁹²².

341. Personnalisation de Tezos. Le L1 de Tezos permet d'accueillir les même types d'application que celui d'Ethereum. Les développeurs devront composer avec la transparence, l'immutabilité et la libre accessibilité de leurs programmes.

b) Avalanche

342. Présentation de Avalanche. Avalanche est une *blockchain* relativement récente fondée par Emin Gün Sirer⁹²³. Elle a pour singularité d'utiliser un mécanisme de consensus en preuve d'enjeu qui permettrait d'obtenir la validation finale des blocs de transaction en un temps très réduit, en sus d'être capable d'accueillir un grand nombre de validateurs, qui n'ont pas besoin de ressources informatiques importantes pour mener à bien leurs missions⁹²⁴.

Avalanche est subdivisée en 3 *blockchain* interconnectées⁹²⁵ :

- la *X-chain* servant seulement à transférer des crypto-monnaies,
- la *C-chain* qui est un L1 généraliste où on peut classiquement déployer et interagir avec des smart contracts ; qui retiendra notre attention,
- et la *P-chain* servant à coordonner les validateurs du protocole et d'infrastructure d'ancrage pour les *subnets* ; qui sont les infrastructures privées d'Avalanche sur lesquelles nous aurons

⁹²¹ V., *supra*, §357

⁹²² Cryptonio.tez. « Rollups on Tezos [Part I] ». Medium, 3 juin 2023. <https://news.tezoscommons.org/rollups-on-tezos-part-i-cdd7b70d53da>.

⁹²³ Cointelegraph. « Emin Gün Sirer: Founder and CEO of Ava Labs | #28 | Cointelegraph Top 100 ». Consulté le 30 juin 2023. <https://cointelegraph.com/top-people-in-crypto-and-blockchain-2022/emin-gun-sirer>.

Emin Gün Sirer is a Turkish-American computer scientist, software engineer and thought leader whose research over the last 20 years has focused primarily on network security, operating and distributed systems, and digital currencies.

⁹²⁴ « Avalanche Consensus | Avalanche Dev Docs ». Consulté le 30 juin 2023. <https://docs.avax.network/learn/avalanche/avalanche-consensus>.

Avalanche Consensus is a consensus protocol that is scalable, robust, and decentralized. It combines features of both classical and Nakamoto consensus mechanisms to achieve high throughput, fast finality, and energy efficiency.

⁹²⁵ « What Are the Differences between the X, P, and C-Chains? | Avalanche Support ». Consulté le 30 juin 2023. <https://support.avax.network/en/articles/6077308-what-are-the-differences-between-the-x-p-and-c-chains>.

également l'occasion de revenir.

La *C-chain* d'Avalanche est celle qui nous intéresse. Elle utilise l'EVM pour faire fonctionner ses smart contracts, ce qui signifie qu'un *smart contract* sur Ethereum est identique à un *smart contract* sur Avalanche.

343. Résilience de Avalanche. Malgré sa jeunesse, Avalanche est une *blockchain* relativement bien décentralisée. Au moment où sont écrites ces lignes, elle dispose en effet de plus de 1200 validateurs participant à son mécanisme de consensus⁹²⁶. La gouvernance de Avalanche est également *on-chain* afin de faire évoluer les paramètres cruciaux de son protocole⁹²⁷. Elle a pour particularité de limiter les modifications possibles de ses paramètres : c'est-à-dire que les propositions d'évolutions ne peuvent se faire évoluer que dans un cadre prédéterminé. De sorte que les parties prenantes peuvent mieux anticiper les changements à venir de la *blockchain*.

Ces éléments plaident en faveur d'une bonne résilience et stabilité de cette *blockchain*, bien que quelques éléments viennent la relativiser. D'abord, on constate que son nombre de validateurs a stagné, voir régressé, avec le temps, ce qui est un mauvais signe sur la capacité de la *blockchain* à rester décentralisée dans le futur⁹²⁸. Il faut ajouter que le nombre en apparence élevé de validateurs n'équivaut pas forcément aux nombres d'individus détenant un nœud. A ce titre 1200 validateurs peut sembler peu lorsqu'on sait que Ethereum, par exemple, compte plus de 800 milles validateurs.

344. Sécurité et commodité d'Avalanche. En choisissant d'instancier l'EVM, la *C-chain* d'Avalanche a fait le choix de permettre aux développeurs d'Ethereum de migrer sans friction vers son protocole. En effet, ces derniers bénéficient dans Avalanche du même langage de programmation de *smart contract*, du même outillage, des mêmes pratiques de développement et des standards, et plus généralement de tout ce qui fait la richesse de l'écosystème ETH et qui contribue à faciliter et sécuriser le travail des développeurs⁹²⁹. Ces derniers bénéficieront aussi des nombreuses applications déjà existantes sur Ethereum, qui n'ont pas eu de difficultés à migrer vers Avalanche pour ces mêmes

⁹²⁶ « Validators | Earn Staking Rewards on Avalanche ». Consulté le 30 juin 2023. <https://www.avax.network/validators>.

⁹²⁷ Michael @ CryptoEQ. « Is Avalanche (AVAX) Governed More by On-Chain or Off-Chain?? », 1 juin 2022. <https://www.publish0x.com/cryptoeq/is-avalanche-avax-governed-more-by-on-chain-or-off-chain-xnnedee>.

⁹²⁸ Twitter. « xTotem :: web3sec sur Twitter », 22 décembre 2022. <https://twitter.com/OxTotem/status/1606269251727151105>.

Let's be honest the number of [Avalanche] validator is down ~20 since beginning of December

⁹²⁹ V., *infra*, §334

raisons.

345. Performance de Avalanche. La *C-chain* d'Avalanche peut accueillir un très haut débit de transactions. On avancerait une capacité de 4500 transactions par seconde⁹³⁰. En sus, le fait qu'elle ne soit pas aussi sursollicitée qu'Ethereum permettrait d'interagir avec elle de manière rapide et peu chère. Mais comme beaucoup d'autres L1 instanciant l'EVM, elle ne règle pas fondamentalement les problèmes de mise à l'échelle. Ces blockchains courent le risque de voir les frais d'interaction avec leurs smart contracts grimper, dès lors que la popularité de la *blockchain* augmente (avec le temps de validation). Autrement dit, des développeurs sur Avalanche prennent le risque d'utiliser une solution temporaire aux problèmes de mise à l'échelle qu'ils fuyaient sur des L1 plus lents.

346. Personnalisation de Avalanche. La *C-chain* de Avalanche est presque en tous points similaires au L1 d'Ethereum ; cela signifie que les développeurs sont étreints par les mêmes limites pour un projet d'automatisation contractuelle. En revanche, Avalanche propose des *subnets* : des solutions permettant de créer des *blockchain* privées interconnectés entre elles (et également à la *P-Chain* et *C-Chain*) qui peuvent être personnalisées à l'envie par leur concepteur⁹³¹. Nous aurons l'occasion de revenir sur ces infrastructures⁹³².

c) Gnosis Chain

347. Présentation de Gnosis Chain. Comme évoqué, Avalanche n'est pas le seul L1 ayant adopté l'EVM ; de nombreuses autres *blockchain* ont également choisi de profiter de l'écosystème Ethereum par ce moyen, dont la *blockchain* *Gnosis Chain*. Avant de porter ce nom, elle existait sous l'appellation *xDai* et consistait en une *sidechain* (sur laquelle nous aurons l'occasion de revenir) adossée à Ethereum depuis 2019 qui permettait de déployer et d'interagir avec des smart contracts de manière instantanée et quasiment sans frais⁹³³. Grâce à sa popularité et au développement de sa

⁹³⁰ Seq. « What Sets Avalanche Apart From Other Blockchains? » Medium (blog), 17 octobre 2021. <https://cryptoseq.medium.com/what-sets-avalanche-apart-from-other-blockchains-3c5f4a4c0889>.

Avalanche can do 4500 tps on just 2 cores, 4 GB memory, whilst also able to scale to unlimited number of subnets.

⁹³¹ « What Is a Subnet? | Avalanche Dev Docs ». Consulté le 30 juin 2023. <https://docs.avax.network/learn/avalanche/subnets-overview>.

A Subnet is a sovereign network which defines its own rules regarding its membership and token economics.

⁹³² V., *supra*, §374

⁹³³ Utorg. « What Is XDai On The Gnosis Chain? » Consulté le 30 juin 2023. <https://utorg.pro/blogs/what-is-xdai-on-the-gnosis-chain/>.

communauté, elle s'est muée en véritable L1 développé par l'organisation *Gnosis*, qui se trouve derrière les portefeuilles *multisig*, sur lesquelles nous reviendrons⁹³⁴. Elle fonctionne avec un mécanisme de preuve d'enjeu nécessitant peu de ressources de la part des validateurs pour y participer ; ce qui lui a permis de réunir rapidement un grand nombre de validateurs, la mettant sur le devant de la scène.

348. Résilience de Gnosis Chain. Comme évoqué, le premier atout de la *Gnosis chain* est donc son grand nombre de validateurs. Elle en compte actuellement plus de 120 000, ce qui fait d'elle, en théorie, une des blockchainss les plus décentralisées au monde après Ethereum⁹³⁵. Cette caractéristique seule en fait un environnement très attractif en matière de pérennité.

Reste que la *blockchain* demeure extrêmement jeune⁹³⁶, et doit encore performer sans accroc pendant quelques temps pour lever tous les doutes sur sa stabilité. En sus, bien qu'elle détienne plus de 120 000 validateurs, nous avons vu que ce nombre n'équivaut pas forcément à celui des réels individus détenant un dispositif de validation⁹³⁷. En réalité, la *blockchain* est donc sans doute bien moins décentralisée que laisse suggérer sa quantité de validateurs. Enfin, la *Gnosis Chain* ne bénéficie pas d'un mécanisme de gouvernance *on-chain* : ses décisions sont prises par le biais du *rough consensus* des parties prenantes de l'écosystème, comme pour Ethereum⁹³⁸.

349. Sécurité et commodité de Gnosis Chain. Puisqu'elle exploite l'EVM, la *Gnosis Chain* offre aux développeurs une expérience identique à celle d'Ethereum, Avalanche et bien d'autres. Cela signifie que ces derniers jouissent des mêmes avantages de sécurité et commodité dans la construction de leurs programmes. En revanche, la jeunesse du protocole implique qu'il manquera

Gnosis Chain (previously, it was called xDai Chain) was created as the world's first stable dollar blockchain and went live in October 2018. It is designed for low-cost, anonymous, and speedy transactional payments with a fixed fee.

⁹³⁴ V., *supra*, §408

⁹³⁵ « Gnosis Chain ». Consulté le 30 juin 2023. <https://www.gnosis.io/>.

By allowing contributors around the globe to easily run a node, Gnosis Chain is secured by over 120k validators.

⁹³⁶ Haig, Samuel. « Gnosis Executes Its Own Merge in Shift to PoS in Boost for Staking ». The Defiant, 8 décembre 2022. <https://thedefiant.io/gnosis-to-execute-its-own-merge-in-shift-to-pos>.

⁹³⁷ V., *infra*, §343

⁹³⁸ Gnosis. « Who Can Govern Gnosis Chain », 18 mars 2022. <http://forum.gnosis.io/t/who-can-govern-gnosis-chain/4133>.

Gnosis Chain does not have a formalized upgrade mechanism (on-chain governance). Instead, it is a decentralized blockchain with many stakeholders. Similar to Ethereum upgrades a hard fork without a chain split can only happen if all those stakeholders simultaneously agree to an upgrade.

d'applications déployées en son sein ; ce qui réduira d'autant plus les possibilités pour les programmeurs d'enrichir leurs smart contracts de différentes fonctionnalités comme sur Ethereum, Tezos ou Avalanche.

350. Performance de Gnosis Chain. La *blockchain* étant récente et encore relativement peu utilisée, elle a l'avantage (temporaire) de ne pas être congestionnée et donc de pouvoir permettre des transactions rapides et peu chères, voir quasiment gratuites. Cependant comme beaucoup de L1 instanciant l'EVM, elle n'innove pas sur ses capacités à se mettre à l'échelle. Cela signifie qu'en cas de bond de la demande, elle courra le risque d'être aussi impraticable que Ethereum.

351. Personnalisation de Gnosis Chain. Au niveau du L1, la *Gnosis Chain* ne présente aucune possibilité d'implémenter quelque chose de fondamentalement différent que sur Ethereum. Les parties pourront créer des applications plus gourmandes en ressources et nécessitant plus de vélocité mais, comme déjà dit, elles prendront le risque de s'exposer à une baisse de performance si l'infrastructure devient trop populaire.

d) Solana

352. Présentation de Solana. Solana figure parmi les blockchains récentes les plus populaires, malgré ses nombreuses différences avec d'autres L1. Elle dispose d'une machine virtuelle propre, la Serum Virtual Machine (SVM), qui permet d'écrire des smart contracts en C, C++ et Rust⁹³⁹. Elle possède également un mécanisme de consensus original, qui mélange preuve d'enjeu et preuve d'histoire, lui permettant d'atteindre la validation théorique de plus de 65 000 transactions par seconde⁹⁴⁰.

353. Résilience de Solana. Solana est une *blockchain* paraissant être bien décentralisée, puisqu'elle compterait plus de 3000 validateurs au moment où sont écrites ces lignes⁹⁴¹. Malgré ce chiffre honorable, les requis techniques pour participer à son mécanisme de consensus de son protocole sont très élevés ; ce qui réduit le nombre de personnes potentielles pouvant être validateurs et donc centralise le réseau. En sus, malgré (et à cause de) ses grosses performances, la *blockchain* a

⁹³⁹ Qui sont des langages de programmation populaires servant à développer bien d'autres logiciels que des smart contracts.

⁹⁴⁰ Syscoin, Corey Crypto-. « Solana: The 65,000 TPS Blockchain: Warp Speed ». Medium (blog), 9 juillet 2020. <https://coreycrypto.medium.com/solana-the-65-000-tps-blockchain-warp-speed-ab34d3ebb85c>.

⁹⁴¹ Throuvalas, Andrew. « Is Solana Really Decentralized? A Validator Health Report ». CryptoPotato (blog), 16 août 2022. <https://cryptopotato.com/is-solana-really-decentralized-a-validator-health-report/>.

Per the foundation's report on Wednesday, Solana currently consists of more than 3400 validators across six continents.

eu à accuser de nombreux dénis de services au cours de son existence⁹⁴².

Les frais de transactions étant si bas, et leur validation si rapide, que des développeurs ont pris l'habitude de coder des robots actionnant un nombre très élevé de transactions (afin faire de l'arbitrage financier ou récupérer des NFT en masse, par exemple) au point de saturer le réseau. A plusieurs reprises, la *blockchain* a donc dû être contrainte de s'arrêter fonctionner pendant plusieurs heures pour être "redémarrée" ensuite. Ces éléments font de la résilience et la disponibilité des points faibles de Solana.

354. Sécurité et commodité de Solana. Solana fait partie des rares L1 qui ont su développer un effet de réseau important autour d'eux, sans exploiter l'EVM. La *blockchain* est pourvue d'un écosystème propre et développé permettant aux codeurs de *smart contract* sur le SVM d'aller vite et bien dans leur tâche. Bien que le langage d'écriture des smart contracts, *Rust*, soit plus complexe à prendre en main que *Solidity*,⁹⁴³ il a su toutefois l'être par une large communauté de développeurs qui partagent les meilleures pratiques et contribuent à sa sécurité.

355. Performance de Solana. Le grand avantage de Solana est sa performance. Grâce à son mécanisme de consensus, la *blockchain* serait ainsi capable de valider plus de 65 000 transactions par seconde⁹⁴⁴ ; ce qui la classe, théoriquement, comme le L1 le plus performant au monde. Les transactions y prenant lieu sont validées instantanément et ne coûtent quasiment rien. Ce sont ces propriétés qui ont permis la popularité de cette *blockchain*, mais ils viennent, comme déjà mentionné, avec des inconvénients : lorsque la *blockchain* est saturée, ses smart contracts courent le risque d'être tout simplement interrompus.

356. Personnalisation de Solana. A l'instar d'Ethereum, Tezos, Avalanche et Gnosis Chain, les smart contracts développés sur Solana seront par défaut accessibles librement, transparents et immuables. Cependant les performances de Solana lui permettront d'accueillir des programmes

⁹⁴² Gilbert, Aleksandar. « Solana To Focus On Stability In 2023 After Repeated Outages ». The Defiant, 2 mars 2023. <https://thedefiant.io/solana-focus-stability-2023>.

⁹⁴³ « Solidity vs. Rust: Everything You Need to Know ». Consulté le 2 juillet 2023. <https://www.alchemy.com/overviews/solidity-vs-rust>.

Solidity is a high-level language that offers a high level of abstraction from the computer system architecture. Because of this, Solidity is simpler to learn and use, which makes it a more user-oriented language. In contrast, Rust is a low-level language that is closer to the computer's hardware and offers good memory efficiency and speed, making it a more machine-oriented language.

⁹⁴⁴ Syscoin, Corey Crypto-. « Solana: The 65,000 TPS Blockchain: Warp Speed ». Medium (blog), 9 juillet 2020. <https://coreycrypto.medium.com/solana-the-65-000-tps-blockchain-warp-speed-ab34d3ebb85c>.

particulièrement gourmands en ressources. Aussi, toutes les fois que des utilisateurs souhaiteront créer des applications à destination d'un grand nombre d'individus et offrant une expérience similaire à celle trouvée dans le web ordinaire, elles ne trouveront donc pas de meilleur candidat parmi les L1 que Solana.

B - Les deuxièmes couches (L2)

357. **Sidechain et rollup.** Les deuxième couches (ou L2) sont le nom donné aux *blockchain* spécialement créées pour améliorer la performance d'autres L1⁹⁴⁵. Parmi celles existantes, nous porterons exclusivement notre attention sur les *rollup* et les *sidechain* ; qui sont toutes deux des solutions offrant un nouvel environnement d'exécution aux smart contracts d'un L1, tout en restant connectées à ceux-ci de différentes manières.

Les *rollup* sont des infrastructures dédiées à décharger les L1 de l'exécution de leurs smart contracts. Autrement dit, ils sont chargés de traiter les opérations de mouvement de cryptoactifs en leur sein, mais postent toujours, dans un format compressé (*rolled-up*) et à intervalles réguliers, le solde issu de ces mouvements dans la *blockchain* (le L1) qu'ils mettent à l'échelle⁹⁴⁶.

Le fait alors de n'utiliser le L1 que pour l'enregistrement du solde et la fourniture du mécanisme de consensus améliore grandement la performance. Les interactions avec un *smart contract* deviennent très peu chères car leurs couts ne correspondent qu'à l'enregistrement des données et leur exécution est quasi-instantanée. Surtout, ce gain de performance ne s'obtient pas au prix de la sécurité et la décentralisation : les données des opérations effectués sur le L2 demeurent constamment enregistrées sur le L1. Il est dit alors que la sécurité des *rollup* est héritée d'un L1 car les utilisateurs peuvent à tout moment retirer leurs cryptoactifs à partir de celui-ci, qui est généralement décentralisé et sécurisé⁹⁴⁷.

Au côté de ces solutions de *rollup*, se trouve les *sidechain* qui constituent, pour leur part, des L1 à

⁹⁴⁵ V., *infra*, §331

⁹⁴⁶ Ethereum.org. « Scaling ». Consulté le 3 juillet 2023. <https://ethereum.org>.

Rollups perform transaction execution outside layer 1 and then the data is posted to layer 1 where consensus is reached. As transaction data is included in layer 1 blocks, this allows rollups to be secured by native Ethereum security.

⁹⁴⁷ Ethereum.org. « Scaling ». Consulté le 3 juillet 2023. <https://ethereum.org>.

This category of off-chain solutions derives its security from Mainnet Ethereum. Layer 2 is a collective term for solutions designed to help scale your application by handling transactions off the Ethereum Mainnet (layer 1) while taking

part entière mais très liés par des ponts (*bridge*) à d'autres L1 dont elles visent à améliorer la performance. Elles sont des blockchainss généralistes classiques nées du besoin de fournir une alternative rapide, peu chère et temporaire aux problèmes de congestion rencontrés par certains L1 populaires (comme Bitcoin et Ethereum). Elles atteignent leur vitesse et la quasi-gratuité de l'interaction avec leurs smart contracts en compromettant clairement sur la décentralisation : ce sont des blockchainss où le nombre de validateurs est très peu élevé. Ce compromis est compensé par le fait qu'elles offrent des nombreuses passerelles vers les L1 dont elles visent à améliorer la performance : ce qui permet aux utilisateurs de transférer aisément leurs cryptoactifs vers des environnements plus décentralisés.

Notre attention portera sur deux types de *rollup* que sont les *optimistic rollup* (OR) (a) et les *zero knowledge rollup* (ZKR) (b), puis les *sidechain* (c).

a) *Optimistic rollup*

358. Présentation des OR. Un *rollup* est donc une *blockchain* abritant l'exécution de *smart contract*, qui va agréger les données des transactions effectuées par ces derniers, pour les enregistrer dans un L1. Dans les *rollup* dits "optimistes", les individus chargés de réaliser cette tâche sont appelées les agrégateurs. Le système par lequel ils s'organisent ressemble fort à celui d'un mécanisme classique de consensus en preuve d'enjeu⁹⁴⁸. En effet, pour mettre à jour le solde des opérations des smart contracts, un agrégateur est sélectionné en partie à raison de la quantité de jetons qu'il a mis en jeu. Il propose alors un solde des opérations qui est présumé valide pendant une période de 7 jours. Si après l'écoulement ce délai, les autres agrégateurs n'ont pas invalidé sa proposition, il récupère sa mise en plus d'une récompense⁹⁴⁹.

Ainsi, ce type de *rollup* est en partie sécurisé par des mécanismes d'incitations économiques. L'appellation *Optimistic* provient du fait qu'en raison de ceux-ci, chacun est incité à agir selon les règles du système et donc les propositions d'agrégats sont considérées (de façon optimiste) comme

advantage of the robust decentralized security model of Mainnet.

⁹⁴⁸ V., *infra*, §332

⁹⁴⁹ Ethereum.org. « Optimistic Rollups ». Consulté le 3 juillet 2023. <https://ethereum.org>.

After a rollup batch is submitted on Ethereum, there's a time window (called a challenge period) during which anyone can challenge the results of a rollup transaction by computing a fraud proof. If the fraud proof succeeds, the rollup protocol re-executes the transaction(s) and updates the rollup's state accordingly. The other effect of a successful fraud proof is that the sequencer responsible for including the incorrectly executed transaction in a block receives a penalty. If the rollup batch remains unchallenged (i.e., all transactions are correctly executed) after the challenge period elapses, it is deemed valid and accepted on Ethereum.

valides jusqu'à preuve du contraire. Parmi les OR les plus connus et fiables, on compte *Arbitrum*⁹⁵⁰ et *Optimism*⁹⁵¹ qui officient sur Ethereum. La *blockchain* Tezos également bénéficie d'OR et d'autres L1 que nous n'avons pas présenté ici⁹⁵².

359. Résilience des OR. En raison de leur fonctionnement, les OR héritent dans une très large mesure de la résilience du L1 sur lequel elles enregistrent leurs données ; qui sont eux-mêmes des infrastructures lentes mais décentralisées et sécurisées comme Ethereum. Ils forment donc, par héritage, des environnements pérennes et stables pour abriter des smart contracts. Mais comme ils sont des solutions assez récentes et complexes, ils demeurent encore porteurs d'éléments de centralisation qui relativisent leur résistance à la censure.

En effet, chez la plupart des OR, il n'y a encore qu'un seul agrégateur chargé de mettre à jour les données, alors que le système est censé être décentralisé. Cela signifie que les usagers du protocole peuvent ne pas voir leurs transactions acceptées par ce seul agrégateur, si celui-ci est compromis ou les refuse (bien qu'ils peuvent toujours, *in fine*, retirer leurs fonds à partir du L1). Au-delà de cet aspect, les équipes chargées des développements des OR disposent souvent de pouvoirs exorbitants pour les premiers instants de l'existence de leurs infrastructures : pouvoir d'arrêt et de mise à jour unilatéral de l'infrastructure, par exemple⁹⁵³. Si ces derniers sont compromis, c'est toute l'infrastructure qui le devient aussi ; ce qui met à risque la pérennité des smart contracts qui y sont déployés.

360. Sécurité et commodité des OR. Les OR peuvent aussi hériter de l'écosystème des L1 qu'ils mettent à l'échelle. En effet, il est très courant que ces derniers copient l'environnement de développement des L1 afin de fournir aux développeurs qui migrent vers eux, une expérience identique. Ainsi *Arbitrum* et *Optimism* disposent de machines virtuelles équivalentes à l'EVM⁹⁵⁴, ce qui signifie que leurs développeurs peuvent mobiliser les mêmes outils pour améliorer leur productivité et la sécurité de leurs applicatifs que ceux d'Ethereum, Avalanche, Gnosis Chain et

⁹⁵⁰ <https://arbitrum.io/>

⁹⁵¹ <https://www.optimism.io/>

⁹⁵² https://tezos.gitlab.io/alpha/smart_rollups.html

⁹⁵³ <https://l2beat.com/scaling/projects/optimism#risk-analysis>

⁹⁵⁴ TokenInsight. « Optimism vs. Arbitrum — A Complete Comparison ». Medium (blog), 4 juillet 2022. <https://tokeninsight.medium.com/optimism-vs-arbitrum-a-complete-comparison-f504f727e4df>.

Also, while Optimism and Arbitrum are both EVM compatible, Optimism uses Ethereum's EVM, whereas Arbitrum runs its own Arbitrum Virtual Machine (AVM). This results in Optimism having only a Solidity compiler, while Arbitrum supports all EVM compiled languages (Vyper, Yul, etc).

consorts.

361. Performance des OR. La raison d'être des OR est la performance accrue qu'ils apportent aux L1. Sur ce point, ils fournissent des résultats très satisfaisants : actuellement les débits de transaction des OR comme *Arbitrum* et *Optimism* seraient de 40 000 transactions par secondes⁹⁵⁵, tandis que sur Tezos, ils permettraient à la *blockchain* d'atteindre les 1 000 000 TPS⁹⁵⁶. En sus, ces solutions sont constamment en voie d'optimisation et accroissent régulièrement cette capacité de validation de transactions par secondes. Il en résulte une expérience pour les utilisateurs où les interactions coûtent très peu chères et sont validées instantanément. Néanmoins comme les données d'un *rollup* sont stockées sur le L1, lorsque celui-ci est submergé, le prix des transactions sur le L2 s'en trouve affecté ; bien que cette corrélation s'amoinde avec le temps⁹⁵⁷.

362. Personnalisation des OR. Pour les OR, tels que *Arbitrum* et *Optimism*, qui ont fait le choix de calquer leur machine virtuelle sur celle d'Ethereum, ceux-ci partagent les mêmes possibilités et étreintes que les L1 qu'ils mettent à l'échelle. A la différence près que des parties pourront déployer des smart contracts bien plus rapides et moins chers que ceux d'Ethereum. Il faut toutefois noter que les *rollup* sont une couche d'exécution de *smart contract* qui n'a vocation qu'à poster des données sur un L1, la manière dont ils sont exécutés dans cette infrastructure peut être protéiforme. Autrement dit, un OR peut embarquer n'importe quel type de machine virtuelle permettant à des parties d'écrire des smart contracts dans divers types de langages de programmation⁹⁵⁸.

b) Les *zk rollup*

363. Présentation des ZKR. Les *zk rollup* (ZKR) sont une autre classe de *rollup*, qui

⁹⁵⁵ Stevens, Decrypt / Robert. « What Is Arbitrum? Speeding Up Ethereum Using Optimistic Rollups ». Decrypt, 20 mars 2023. <https://decrypt.co/resources/what-is-arbitrum-speeding-up-ethereum-using-optimistic-rollups/>.

While Ethereum manages a mere 14 transactions per second, Arbitrum races ahead at 40,000 TPS. Transactions cost several dollars to complete on Ethereum, while they cost about two cents on Arbitrum.

⁹⁵⁶ « The Road to a Million TPS (and beyond): Smart Rollups Are Coming ». Consulté le 3 juillet 2023. <https://research-development.nomadic-labs.com/smart-rollups-are-coming.html>.

⁹⁵⁷ L'acceptation prochaine de l'EIP 4844 dans le protocole Ethereum réduirait très significativement les frais d'interaction avec les *rollup*.

Ethereum Improvement Proposals. « EIP-4844: Shard Blob Transactions ». Consulté le 12 juillet 2023. <https://eips.ethereum.org/EIPS/eip-4844>.

⁹⁵⁸ Labs, Offchain. « Hello, Stylus ». Medium (blog), 7 février 2023. <https://offchain.medium.com/hello-stylus-6b18fecc3a22>.

Stylus will enable users to deploy programs written in popular programming languages to Arbitrum One and Arbitrum Nova. That's right: Rust, C, C++, and more, side-by-side with existing Solidity dApps on the same Arbitrum blockchain.

différent des OR par le fait que les données des opérations effectuées dans leur environnement d'exécution soient soumises aux L1 avec une preuve cryptographique de leur validité. C'est-à-dire que les opérations effectuées par les smart contracts dans ces L2 sont assurées d'être en concordance avec leur soldes postés dans le L1 grâce à la technologie de preuve à divulgation non nulle ("zero knowledge proof"- d'où son appellation)⁹⁵⁹. Cette classe de *rollup* est généralement considérée comme supérieure aux OR puisqu'elle serait plus sécurisée ; en raison du fait que son fonctionnement est garanti uniquement par de la cryptographie et non pas par des comportements humains (même incités économiquement)⁹⁶⁰. Il existe un mécanisme de consensus sur le postage des données (lui aussi calqué sur celui de la preuve d'enjeu), mais qui vise seulement à garantir que le solde soit régulièrement enregistré.

364. Résilience des ZKR. Les ZKR héritent de la résilience des L1 sur lesquels ils postent leur données de la même manière que les OR. Cela signifie qu'ils constituent également des infrastructures pérennes et stables pour accueillir des smart contracts lorsque ces derniers sont décentralisés et sécurisés. En revanche, étant aussi (voir plus) immature encore que les OR, ils sont également porteurs des mêmes éléments de centralisation et de leurs inconvénients associés : comme la possibilité de mise à jour et d'interruption unilatéral par l'équipe en charge du développement⁹⁶¹. Il faut rajouter comme spécificité des ZKR que la génération des preuves à divulgation nulle est une tâche beaucoup plus nécessiteuse en ressources que celle des agrégateurs des OR. Autrement dit, les conditions pour pouvoir participer au mécanisme de postage de données dans le L1 seront plus lourdes

⁹⁵⁹ Ethereum.org. « Zero-Knowledge Rollups ». Consulté le 3 juillet 2023. <https://ethereum.org>.

Zero-knowledge rollups (ZK-rollups) are layer 2 scaling solutions that increase throughput on Ethereum Mainnet by moving computation and state-storage off-chain. ZK-rollups can process thousands of transactions in a batch and then only post some minimal summary data to Mainnet. This summary data defines the changes that should be made to the Ethereum state and some cryptographic proof that those changes are correct.

⁹⁶⁰ Gluchowski, Alex. « Optimistic vs. ZK Rollup: Deep Dive ». Medium, 7 avril 2021. <https://blog.matter-labs.io/optimistic-vs-zk-rollup-deep-dive-ea141e71e075>.

In a ZK Rollup, every state transition is verified by the Rollup smart contract before it becomes effective. It is strictly not possible for operators to steal the funds or corrupt the Rollup state. ZKR relies on the censorship-resistance of L1 only for its liveness, not for its security. There is no need for anyone to monitor the ZKR: after a block is verified, user funds are always guaranteed to be eventually retrievable even if operators refuse to cooperate.

⁹⁶¹ Gluchowski, Alex. « Optimistic vs. ZK Rollup: Deep Dive ». Medium, 7 avril 2021. <https://blog.matter-labs.io/optimistic-vs-zk-rollup-deep-dive-ea141e71e075>.

Most ZK-rollups use a "supernode" (the operator) to execute transactions, produce batches, and submit blocks to L1. While this ensures efficiency, it increases the risk of censorship: malicious ZK-rollup operators can censor users by refusing to include their transactions in batches.

pour des ZKR que pour les OR, ce qui aggrave la centralisation et ses désavantages⁹⁶².

365. Sécurité et commodité des ZKR. Tous comme les OR, la plupart des ZKR ont fait le choix d'utiliser une machine virtuelle équivalente à celles des L1 avec qui ils fonctionnent. On compte ainsi de nombreux ZKR qui instancient des machines virtuelles équivalentes à l'EVM afin de fournir aux développeurs un environnement identique à celui d'Ethereum : on les nomme les zkEVM. Ces derniers hériteront donc de la toute la sécurité et commodité caractéristiques de cet environnement⁹⁶³.

366. Performance des ZKR. En théorie, les ZKR sont une technique de mise à l'échelle plus performante que les OR car ils nécessitent de poster moins de données sur les L1 : ce qui signifie des interactions moins coûteuses et une validation plus rapide. Aujourd'hui, on estime que leur débit de transaction serait capable d'atteindre plus de 100 000 transactions par seconde⁹⁶⁴. Tout comme les OR, ils bénéficient en sus de constantes optimisations laissant promettre des performances encore meilleures.

367. Personnalisation des ZKR. Les zkEVM, comme leur nom le suggère, sont des ZKR calés sur l'EVM et ne permettent donc que le déploiement de *smart contract* compatibles avec cette machine virtuelle. Néanmoins comme pour les OR, les ZKR peuvent embarquer différentes machines virtuelles pour poster les données sur L1. Même si la technologie est encore balbutiante au moment où sont écrites ces lignes, il est noté différentes initiatives, comme celles de starkNet⁹⁶⁵, qui offre aux développeurs sur cette technologie de plus amples possibilités de personnalisation de leurs smart

⁹⁶² Certaines initiatives visent à tacler de problème en proposant des schémas de décentralisation innovants.

HackMD. « Decentralized zk-Rollup ». Consulté le 12 juillet 2023.
<https://hackmd.io/@yezhang/SkmyXzWMY#Decentralized-zk-Rollup>.

zk-Rollup requires prover to generate a succinct proof for a batch of transactions off-chain. However, the proof generation process is costly for complicated smart contract transactions. We introduce a Layer-2 proof outsourcing mechanism which can incentivize rollers to generate proofs for us using GPU/ASIC.

⁹⁶³ Il est pensé à ZkSync Era (<https://zksync.io/>) et Polygon zkEVM (<https://polygon.technology/polygon-zkevm>)

⁹⁶⁴ Nambiampurath, Rahul. « What Are ZK-Rollups? » The Defiant, 1 décembre 2022. <https://thedefiant.io/what-are-zk-rollups>.

This all changed in 2022 with zkSync 2.0, offering both EVM compatibility and up to 100,000 transactions per second (tps) speed. By leveraging Ethereum's security, as the largest DeFi network, ZK-rollups provide superior security than other scaling solutions.

⁹⁶⁵ Starkware. « Tech Stack ». Consulté le 3 juillet 2023. <https://starkware.co/tech-stack/>.

StarkWare defines the Cairo-Architecture and implements it both as a native VM, and as a proving-optimized algebraic form. StarkWare develops additional developer tools around the Cairo programming language, such as compiler from higher-level languages, debugger, IDE support, and more.

contracts.

c) Les *sidechain*

368. Présentation des *sidechain*. Comme évoqué, les *sidechain* sont des L1 ordinaires qui se rattachent, par divers moyens, à d'autres L1 dont ils visent à améliorer la performance. Ils ont donc la même structure qu'une *blockchain* généraliste classique (avec un mécanisme de consensus et une machine virtuelle propre) mais le but de leur existence est de mettre à l'échelle une *blockchain* lente. Ils n'ont donc pas vocation à exister de manière autonome, mais davantage comme une solution annexe et relativement temporaire aux problèmes de mise à l'échelle d'un L1. Pour cette raison, leur qualification de L2 est souvent contestée: ils n'utilisent pas le L1 pour enregistrer les données de transaction⁹⁶⁶ donc ils ne peuvent être qualifiés de L2. La *sidechain* la plus connue est Polygon, elle utilise le mécanisme de consensus de la preuve d'enjeu et instancie l'EVM⁹⁶⁷.

369. Résilience des *sidechain*. Pour parvenir à soutenir un grand débit de transactions, les *sidechain* opèrent un compromis clair sur la décentralisation : les validateurs de leur mécanisme de consensus de preuve d'enjeu sont peu nombreux. Elles sont donc toujours très centralisées et ne constituent pas, à ce titre, des environnements très pérennes et stables pour faire résider des smart contracts⁹⁶⁸. Néanmoins, c'est une situation normalement connue des utilisateurs de cette *blockchain*, qu'ils ont accepté compte tenu du fait qu'ils peuvent transférer facilement leurs cryptoactifs vers des infrastructures plus résilientes.

370. Sécurité et commodité des *sidechain*. En règle générale, les *sidechain* fournissent le même environnement de développement des L1 qu'elles tentent de mettre à l'échelle. Dans Polygon, les développeurs bénéficient de la même expérience que sur Ethereum et peuvent ainsi mobiliser les mêmes outils pour aller plus vite dans leurs tâches et produire des applicatifs aussi sécurisés. En sus, ces *sidechain* parviennent parfois à créer un large effet de réseau autour d'elles qui se manifeste par un grand nombre d'applications déjà déployés en leur sein avec lesquels des développeurs peuvent

⁹⁶⁶ user610620. « Is Polygon (Matic) a layer-2 or a sidechain? » Forum post. Ethereum Stack Exchange, 29 mars 2022. <https://ethereum.stackexchange.com/q/125024>.

Polygon PoS can be considered as a combination of sidechain and layer2 solution because it is a separate chain that has its own consensus mechanism but also uses the main Ethereum layer for some features in its consensus mechanism to improve security.

⁹⁶⁷ <https://polygon.technology/>

⁹⁶⁸ Adejumo, Oluwapelumi. « Crypto Investments Fund Founder Says Polygon Is “Highly Insecure & Centralized” ». CryptoSlate (blog), 16 août 2022. <https://cryptoslate.com/crypto-investments-fund-founder-says-polygon-is-highly-insecure-centralized/>.

faire interagir leur programme⁹⁶⁹. Cela renforce d'autant plus la commodité de l'écosystème.

371. Performance des *sidechain*. Comme déjà évoqué, les *sidechain* sont marquées par un grand débit de transactions qu'elles obtiennent en compromettant sur la décentralisation. En déployant leur programme dans ce type de *blockchain*, les développeurs ont la garantie que leur smart contracts s'exécuteront de manière extrêmement rapide et peu chère. Toutefois, ces infrastructures ne règlent pas fondamentalement les problèmes de mise à l'échelle et ne sont donc pas étrangères aux problèmes de congestion : des pics de demande peuvent faire monter haut les frais d'interaction avec les smart contracts et provoquer des ralentissements⁹⁷⁰.

372. Personnalisation des *sidechain*. Les *sidechain* telles que Polygon ne peuvent pas accueillir des programmes différents de ceux de la machine virtuelle des L1 qu'elles mettent à l'échelle. Mais à l'instar de toutes *blockchain* performantes, elles sont à même d'héberger des applications plus demandereses en ressources.

§ II - Les *infrastructures* spécifiques

373. Blockchain et *rollup* spécifiques. Au côté des infrastructures généralistes, que nous venons de présenter, se trouve celles que nous nommons les infrastructures "spécifiques" (ou *ad-hoc*) ; que nous appelons ainsi car elles sont des blockchainss conçues et optimisées spécialement pour une application ou un seul type d'application. Elles n'ont donc pas vocation à accueillir toutes sortes de programmes et les forcer à partager les ressources de l'infrastructure avec d'autres. Elles prennent généralement la forme de L1 privés (A), bien qu'on assiste progressivement à l'émergence de *rollup* spécifiques (B).

A - Les blockchains spécifiques

374. Présentation des blockchainss spécifiques. Les blockchains spécifiques sont des L1 taillées-sur-mesure pour les besoins d'un porteur de projet. Dans celles-ci, il peut implémenter son

⁹⁶⁹ Ainsi, la *sidechain* Polygon aurait été la blockchain la plus utilisée en 2023.

Brenda Mary. « Polygon Became the Most Widely Used Blockchain in 2023 », 18 mai 2023.
<https://crypto.news/polygon-became-the-most-widely-used-blockchain-in-2023/>.

⁹⁷⁰ Chawla, Vishal. « NFT Gamers Are Clogging Up Polygon ». Crypto Briefing, 5 janvier 2022.
<https://cryptobriefing.com/nft-gamers-are-clogging-up-polygon/>.

Polygon is experiencing congestion because Sunflower Farmers, a play-to-earn game, is placing high demands on the network.

propre mécanisme de consensus, sa propre machine virtuelle et ses propres règles d'interaction avec sa *blockchain* et ses smart contracts.⁹⁷¹ Même si ce n'est pas une règle, le plus souvent ces blockchains sont dites privées. Cela signifie que :

- leur mécanisme de consensus est hypercentralisé : il s'agit souvent de celui de la preuve d'autorité où seules les personnes disposant d'un certificat, délivré par le porteur de projet, sont autorisées à valider des blocs de transactions⁹⁷² ;
- et l'accès à la *blockchain* et aux smart contracts est limité à certains individus par le porteur de projet.

On compte plusieurs protocoles proposant des outils de création de *blockchain* spécifiques : Cosmos⁹⁷³, Avalanche⁹⁷⁴ et Polygon⁹⁷⁵ figurent parmi les plus connus. Les kits de développement (SDK) qu'ils mettent à disposition permettent de créer des blockchains personnalisées en quelques heures avec différents mécanismes de consensus prêts à l'emploi, des validateurs à la demande et un choix fourni de machines virtuelles.

375. Résilience des blockchains spécifiques. La résilience est donc la principale faiblesse d'une telle infrastructure puisqu'en principe, le porteur projet se lance avec un mécanisme de

⁹⁷¹ « Application-Specific Blockchains | Cosmos SDK ». Consulté le 3 juillet 2023.
<https://docs.cosmos.network/main/intro/why-app-specific>.

Application-specific blockchains are blockchains customized to operate a single application. Instead of building a decentralized application on top of an underlying blockchain like Ethereum, developers build their own blockchain from the ground up.

⁹⁷² Bellanca, Chloe. « Qu'est-ce que le Proof of Authority ? » Coinhouse (blog), 28 juin 2019.
<https://www.coinhouse.com/fr/academie/blockchain/proof-of-authority/>.

The Proof-Of-Authority (PoA) est une méthode de consensus qui donne à un nombre restreint et désigné d'acteurs d'une blockchain le pouvoir de valider des transactions ou interactions avec le réseau et de mettre à jour son registre plus ou moins distribué.(...) Le Proof-of-Authority est souvent privilégié par les blockchains privées ou de consortium.

⁹⁷³ Cosmos Network. « Cosmos Network - Internet of Blockchains ». Consulté le 12 juillet 2023. <https://cosmos.network>.

Write your custom blockchain. Compose your blockchain application with a mix of prebuilt modules and your own custom modules.

⁹⁷⁴ « What Is a Subnet? | Avalanche Dev Docs ». Consulté le 30 juin 2023.
<https://docs.avax.network/learn/avalanche/subnets-overview>.

A Subnet is a sovereign network which defines its own rules regarding its membership and token economics. Subnets use virtual machines to specify their own execution logic, determine their own fee regime, maintain their own state, facilitate their own networking, and provide their own security.

⁹⁷⁵ « Build Your Own Blockchain, without the Complexity ». Consulté le 12 juillet 2023.
<https://polygon.technology/polygon-supernets>.

Polygon Supernets allows you to create high-performing, customizable App-chains with compliance implementation options, in a fast seamless way.

consensus extrêmement centralisé. Dans ce cas, la *blockchain* n'est alors pas beaucoup plus résiliente qu'un serveur centralisé classique et se trouve dépourvue d'une de ses propositions de valeur principale. Au point tel que pendant longtemps, ces formes de *blockchain* ont même été accusées d'usurper leur titres, tant leur caractéristiques étaient fondamentalement contraires à celles qu'on attendait d'elles⁹⁷⁶. Néanmoins, une *blockchain* spécifique peut être relativement décentralisée si son porteur de projet le souhaite. Ce dernier peut implémenter un mécanisme de consensus réunissant plusieurs validateurs, même si il les a choisis lui-même, et ouvrir progressivement les conditions d'accès pour le rejoindre⁹⁷⁷.

376. Sécurité et commodité des blockchainns spécifiques. En choisissant d'implémenter sa propre infrastructure, un porteur de projet conçoit un environnement modelé pour ses besoins où il privilégie donc sa commodité. Mais ce faisant, il peut se priver de la richesse et la maturité de l'écosystème d'une *blockchain* public et éprouvée depuis des années. Or, nous avons vu que ces éléments peuvent être fournisseurs d'une commodité de développement et d'une sécurité pour les smart contracts⁹⁷⁸.

Toutefois, ces blockchains peuvent conférer une forme de sécurité qu'on ne retrouve pas dans les infrastructures généralistes. En optimisant une *blockchain* pour une seule application, la surface d'apparitions de bogues ou de possibilités de piratages s'en trouve mécaniquement diminuée. En sus, les développeurs ne se retrouvent pas obligés de composer avec les défauts des machines virtuelles qu'ils n'ont pas choisies, ils peuvent les modifier pour ne réaliser que les opérations qui les intéressent⁹⁷⁹.

377. Performance des blockchainns spécifiques. Grâce à leur architecture centralisée et

⁹⁷⁶ Hussey, Decrypt / Matt. « Private vs Public Blockchains | What Is The Difference ? » Decrypt, 21 janvier 2019. <https://decrypt.co/resources/private-blockchains/>.

Private blockchains are not without their critics. Here are some issues: Unnecessary - Public blockchains are designed to work in a trustless environment. If the nodes are trusted, it may be simpler and cheaper to just use a database.

⁹⁷⁷ Un porteur de projet peut créer sa blockchain avec pour mécanisme de consensus le PoA et ajouter plusieurs validateurs au cours de son existence jusqu'à faire évoluer son mécanisme de consensus vers un PoS ouvert à tous.

⁹⁷⁸ V., *infra*, §330

⁹⁷⁹ MARIN, Gautier. « Why Application-Specific Blockchains Make Sense ». Medium, 8 février 2019. <https://blog.cosmos.network/why-application-specific-blockchains-make-sense-32f2073bfb37>.

The attack-surface of a Virtual-machine blockchain is large. Most of it comes from the complexity of the Virtual-machine itself. The security analysis is simpler in application-specific blockchains because you only have to consider how the different parts of your application interact with each other. You don't have to worry about the interactions between the application and the Virtual-machine mechanics. The complexity of the Virtual-machine mechanics is usually the cause of bugs like call stack limit (DAOBug), DelegateCall (Parity Bug #1), contract suicide (Parity Bug #2), etc.

optimisée pour un seul but, ces blockchains peuvent également être très performantes. La raison étant que la seule application qui y est déployée n'a pas besoin de rentrer en compétition avec des milliers d'autres pour exploiter les ressources de la *blockchain*. Il s'agit donc d'une infrastructure qui convient, mieux que tout autre, aux applications qui seraient très gourmandes en ressources et nécessiteraient des interactions à haute fréquence avec une *blockchain*⁹⁸⁰.

378. Personnalisation des blockchains spécifiques. La proposition de valeur principale de ces infrastructures est qu'elles offrent de larges possibilités de personnalisation aux porteurs de projets. Comme évoqué, ces derniers peuvent choisir leur propre machine virtuelle (et donc le langage de programmation de leur programme), configurer leur propre mécanisme de consensus et même les caractéristiques d'accessibilité de leur *blockchain*. Ce contrôle est un atout pour des personnes souhaitant s'exposer, seulement dans la mesure qu'elles souhaitent, aux bénéfices de la *blockchain*.

B - Les *rollup* spécifiques

379. Présentation des *rollup* spécifiques. Les *rollup* spécifiques sont des *rollup* conçus, comme les blockchains *ad-hoc* précédemment décrites, pour accueillir qu'une seule ou un seul type d'application. Ils ont ainsi le fonctionnement d'un *rollup* classique, qu'il soit OR ou ZKR, avec la particularité qu'ils soient configurés pour convenir aux besoins particuliers d'un porteur de projet. Ils découlent d'une philosophie d'architecture de plus en plus populaire dans le milieu de la *blockchain*, appelée la « modularité », consistant à extirper et confier l'exécution de tâches habituellement toutes effectuées par une *blockchain* à différentes infrastructures spécialisées dans leur réalisation⁹⁸¹. Ainsi,

⁹⁸⁰ Infura Blog | Tutorials, Case Studies, News, Feature Announcements. « The Benefits and Tradeoffs of Application-Specific Blockchains », 17 janvier 2023. <https://blog.infura.io/post/the-benefits-and-tradeoffs-of-application-specific-blockchains>.

Performance is another reason to deploy on an application-specific blockchain. Many blockchains have gas limits to prevent certain DDoS attack vectors and reduce hardware requirements for blockchain nodes. But these designs limit throughput (measured in transactions per second) and result in degraded UX when an application surges in usage. With an appchain, developers can modify gas limits and other runtime parameters to optimize performance for dapps. As most appchains can be permissioned-by-choice, it's easy to enlist validators that meet specific hardware requirements.

⁹⁸¹ Liesl Eichholz. « Beyond Monolithic: The Modular Blockchain Paradigm ». Fuel, 5 octobre 2022. <https://fuel-labs.ghost.io/beyond-monolithic-the-modular-blockchain-paradigm/>.

The core functions of a blockchain are: Execution - Transaction processing and computation. Settlement - Dispute resolution and bridging. Consensus - Transaction ordering. Data availability - Ensures data is available. Traditionally, blockchain designs have been monolithic. This means that all the functions of the blockchain are handled on a single layer. The thesis of modular blockchains is that a single blockchain doesn't need to handle all these components on its own. Instead, by disaggregating these core components, individual blockchains can focus on specializing in a specific area, leading to significant optimizations.

au lieu qu'un L1 soit, tout à la fois, en charge de la fourniture d'un mécanisme de consensus, l'exécution des smart contracts et l'enregistrement des données, il serait plus rationnel qu'une infrastructure soit constituée pour le stockage de données, une autre chargée de valider les blocs de transactions et une dernière pour exécuter un programme ou un type de programmes. Ethereum, Tezos et Celestia font partie de ces L1 qui se sont explicitement engagés dans cette voie, en visant à devenir des infrastructures spécialisées dans la fourniture de service de consensus et/ou d'enregistrement de données⁹⁸². Des *rollup* pourraient alors s'y connecter afin d'être dédiés à l'exécution d'applications uniques.

380. Résilience des *rollup* spécifiques. La spécificité (et supériorité) des *rollup* spécifiques sur leurs équivalents *blockchain* est que les premiers tirent, comme tout *rollup*, leur sécurité d'autres L1. Ils ne souffrent donc pas comme les blockchains spécifiques des risques de centralisation (et partant de faible résilience) puisque leur mécanisme de consensus sont ceux des L1 qui sont, en général, très décentralisées ; ce qui garantit un environnement pérenne et stable pour le programme qu'ils hébergent.

381. Sécurité et commodité des *rollup* spécifiques. L'environnement d'un *rollup* spécifique est modélisable selon les grés du porteur du projet. Comme pour les blockchains spécifiques, cela peut être un grand avantage en termes de commodité et de sécurité dans la mesure où le concepteur peut s'abstraire des défauts, limites et failles des environnements d'exécution des L1 afin d'en créer un sécurisé et taillé pour ses propres besoins. Néanmoins, ce faisant il court le risque de se priver de la richesse de l'écosystème des environnements L1 qui prodiguent eux aussi des nombreux éléments sécurisant et facilitant le développement de *smart contract*.

382. Performance des *rollup* spécifiques. Une des raisons d'être des *rollup* spécifiques est d'offrir des performances accrues aux programmes déployés en leur sein par rapport à ceux de L1 ou *rollup* généralistes. Dans un *rollup* spécifique, l'application déployée ne partage pas les ressources de l'infrastructure avec d'autres. Cette dernière est optimisée pour son support ; il est donc possible de créer des applications dotées d'une grande vélocité et de frais d'interaction presque nuls. Néanmoins par définition, un *rollup* enregistre ses données sur un L1, ce qui signifie qu'il y aura toujours une part incompressible de frais d'interaction correspondant à ce coût d'enregistrement. Même si celui-ci peut être minimal, les performances de ces infrastructures peuvent s'en retrouver

⁹⁸² James Strudwick et Dylan Kugler. « A Comprehensive Guide to Rollups and Mina's Place in the Landscape ». Mina Protocol, 4 novembre 2023. <https://minaprotocol.com/blog/guide-to-rollups-and-minas-place-in-the-landscape>.

The first is an application specific rollup that supports only a certain type of application or functionality. In contrast, a general purpose rollup allows projects to customize logic and smart contracts independent of a rollup operator, which is not possible with an application specific rollup. As crypto moves towards a multichain model, application specific rollups can enable apps to distribute across different environments.

inférieures par rapport à celles de L1 spécifiques.

383. Personnalisation des *rollup* spécifiques. Le grand intérêt d'une structure spécifique est de pouvoir être personnalisée pour satisfaire les besoins de son concepteur. Les *rollup* spécifiques ne dérogent pas à la règle en permettant à des porteurs de projets d'implémenter leur propre machine virtuelle, langages de programmation et règles d'interaction avec leur infrastructure. Ils seront toutefois limités dans leur possibilité d'implémenter leur propre mécanisme de consensus, puisque ce dernier sera, en principe, celui du L1 sur lequel les données de leur application seront enregistrés. De plus, les *rollup* spécifiques ne bénéficient pas de la même maturité de leur écosystème que celui des blockchains spécifiques. On ne retrouvera donc pas, aussi aisément, des kits de développement permettant d'en déployer en quelques heures malgré que leur architecture soit complexe à mettre en

œuvre. **Tableau récapitulatif de l'évaluation des différentes infrastructures d'exécution :**

Infrastructure d'exécution	Résilience	Sécurité des smart contracts	Commodité	Performance	Personnalisation
Ethereum	Très élevée	Très élevée	Très élevée	Très faible	Très faible
Tezos	Faible	Elevée	Elevée	Moyenne	Très faible
Avalanche	Moyenne	Très élevée	Elevée	Elevée	Faible
Gnosis Chain	Elevée	Très élevée	Moyenne	Elevée	Faible
Solana	Moyenne	Elevée	Très élevée	Elevée	Faible
Rollup généraliste	Elevée	Très élevée	Très élevée	Très élevée	Faible
Blockchain privée	Très faible	Moyenne	Elevée	Très élevée	Très élevée
Rollup spécifique	Elevée	Moyenne	Très élevée	Très élevée	Très élevée

Section II – La sélection des infrastructures

384. Les choix de *blockchain* par principe et exception. Les différentes infrastructures d'exécutions présentées et évaluées, nous pouvons être en mesure d'établir lesquelles doivent être choisies par les parties pour leur projet de formalisation contractuelle. Selon nous, celles-ci seront avisées, par principe, de déployer leurs smart contracts sur une infrastructure généraliste (§1), mais par exception elles pourront faire le choix d'utiliser une infrastructure spécifique lorsque l'opération

contractuelle qu'elles souhaitent exécuter nécessite des contraintes particulières (§2).

§ I - Les choix par principe

385. Les infrastructures généralistes, par principe. Pour la plupart des contrats que des parties seront intéressées d'exécuter dans la *blockchain*, et particulièrement ceux consistant en des processus de transferts d'actifs comme déjà évoqués⁹⁸³, il nous apparaît que la première couche d'Ethereum (A) ou un *rollup* connecté à celle-ci (B) constitueront les meilleures infrastructures d'exécution.

A – Ethereum

386. Qualités d'Ethereum intéressant un contrat intelligent. Ethereum, en tant que L1, dispose de plusieurs atouts intéressant des parties cherchant à implémenter un contrat intelligent. D'abord, il s'agit d'une *blockchain* particulièrement décentralisée, et donc résiliente. A ce titre, elle fournit aux parties un environnement très pérenne pour héberger les programmes de leur contrat intelligent. Sa décentralisation est aussi synonyme de haute disponibilité: sur Ethereum, les parties auront bien plus de garanties que sur d'autres blockchains que leur programme ne subira pas de déni de service ou d'interruption.

Surtout, l'ancienneté d'Ethereum pourvoit les parties de l'écosystème le plus riche et éprouvé du milieu de la *blockchain* : celui de l'EVM. Cela se traduit par une myriade d'outils qu'elles pourront utiliser afin de produire efficacement les programmes sécurisés qui exécuteront leur contrat. La sécurité étant un sujet majeur lors du recours à un *smart contract*, il est crucial que des parties trouvent dans l'écosystème qu'elles choisissent tous les éléments possibles qui concourront à neutraliser le risque de bogues et piratages : standards, bibliothèques, pratiques *open-source*, éprouvés et audités par une large communauté de développeurs.

387. Limites d'Ethereum pour un contrat intelligent. En revanche, les limitations techniques actuelles d'Ethereum n'en font pas un environnement adapté pour accueillir des programmes destinés à être exécutés rapidement et sans grand frais d'interaction⁹⁸⁴. Il s'agit en effet d'une *blockchain* relativement lente et congestionnée. Cela signifie que les parties devront budgéter leur utilisation des smart contracts et que ceux-ci auront une latence notable.

388. Contrats adéquats pour une exécution sur Ethereum. Ces éléments nous permettent d'affirmer qu'Ethereum convient pour exécuter tous les contrats intelligents auxquels les

⁹⁸³ V., *infra*, §339

⁹⁸⁴ V., *infra*, §335

parties veulent conférer le maximum de sécurité, au détriment du reste. Pour s'approprier plus intelligemment les qualités d'Ethereum, les parties préféreront y déployer des contrats à très fort enjeu et ayant une durée de vie longue. Par fort enjeu, nous entendons les contrats qui opèrent des transferts de sommes d'argent très importantes et/ou qui concernent des processus particulièrement cruciaux. Dans ceux-ci, les bénéfices qu'entendent retirer les parties de l'exécution *on-chain* surpasseront de loin leurs défauts. De plus, la hauteur de l'enjeu du contrat intelligent justifiera que soit utilisé l'écosystème procurant le plus d'éléments sécurisants, et pour lequel il n'existe pas meilleur que Ethereum.

Ce L1 convient également pour l'exécution de contrats dont les obligations peuvent se déclencher plusieurs années après la signature. En effet, dans cette hypothèse les parties auront besoin d'une infrastructure dont elles ont la conviction qu'elle résistera à l'épreuve du temps : autrement dit, que le *smart contract* qu'elles déploieront aura d'excellentes chances de s'exécuter même plusieurs années après son déploiement. Or en raison de sa forte décentralisation, Ethereum est statistiquement une des blockchainss qui a le plus de chance d'avoir la longévité la plus étendue et stable⁹⁸⁵.

A titre d'illustration, voici une liste de clauses et contrats convenant pour être exécutés sur Ethereum:

- les conventions en rapport avec la mort telles que les testaments⁹⁸⁶, les contrats d'assurance-vie et de donation-partage. Nonobstant la faisabilité juridique de la démarche, ce sont des opérations à fort enjeu et qui peuvent déclencher des transferts d'actifs plusieurs années après la conclusion du contrat. A ce titre, les programmes les exécutant doivent résider sur une infrastructure particulièrement sécurisante et résiliente.
- certaines clauses de contrats M&A comme la clause d'ajustement de prix ou de garantie de passif⁹⁸⁷. Ce sont des clauses qui prévoient des obligations pouvant être activées plusieurs années après leur signature et qui impliquent très souvent des sommes élevées⁹⁸⁸.
- des contrats encadrant des opérations de « long dépôts », comme des fiducies ou les contrats

⁹⁸⁵ V., *infra*, §333

⁹⁸⁶ Sreehari, P, M Nandakishore, Goutham Krishna, Joshin Jacob, et V. S. Shibu. « Smart will converting the legal testament into a smart contract ». In 2017 International Conference on Networks & Advances in Computational Technologies (NetACT), 203-7, 2017. <https://doi.org/10.1109/NETACT.2017.8076767>.

⁹⁸⁷ V., *infra*, §148

⁹⁸⁸ Diallo, Abdoulaye. « Fusions-Acquisitions et Smart contracts ». Medium, 9 mars 2022. <https://abdoulaye77124.medium.com/fusions-acquisitions-et-smart-contract-71e7985cf84b>.

d'entiercement logiciel⁹⁸⁹ : là-encore, il s'agit de contrats à fort enjeux et prévoyant des obligations dont les conditions de déclenchement peuvent être tardives. Dans un contrat de séquestre logiciel, un tiers doit impérativement délivrer les actifs lorsque l'entreprise fournisseuse se trouve en liquidation judiciaire ; ce qui est un évènement pouvant arriver plusieurs années après la conclusion du contrat.

B - Un *rollup* généraliste sur Ethereum

389. Qualités d'un *rollup* généraliste sur Ethereum intéressant un contrat intelligent.

Nous avons vu que, dans une large mesure, les *rollup* héritent de toutes les qualités des L1 auxquels ils sont connectés, tout en se débarrassant de leur défaut principal qui est leur performance médiocre⁹⁹⁰. Ceci est d'abord vrai pour la résilience : les *rollup* sur Ethereum, qu'ils soient OR ou ZK, enregistrent leurs données dans un L1 très décentralisé⁹⁹¹ ; si jamais les *rollup* sont compromis, les parties disposent toujours de la possibilité de retirer leurs cryptoactifs à partir de données enregistrées dans l'infrastructure pérenne et stable que constitue Ethereum. Cela fait d'un *rollup*, un environnement qui est, dans cette mesure, aussi résistant que l'infrastructure sur laquelle il poste ses données.

Les *rollup*, comme Arbitrum ou PolygonZkEVM, héritent aussi de l'écosystème de Ethereum. En effet, ils embarquent des machines virtuelles équivalentes à l'EVM, ce qui permet aux parties non seulement de retrouver dans les *rollup* la grande majorité des applications qui existaient sur Ethereum (qui n'ont eu aucun mal à faire la migration) ; mais aussi de mobiliser tout l'outillage de l'EVM pour construire commodément des smart contracts sécurisés.

390. Limites d'un *rollup* généraliste sur Ethereum pour un contrat intelligent.

L'infrastructure d'un *rollup* n'est cependant pas exempt de bogues ou de portes dérobées. En effet, le *smart contract* déployé sur le L1 faisant office de registre de toutes les opérations effectués sur le L2 peut être compromis⁹⁹². Même si ce risque peut être éminemment maîtrisé, il signifie que déployer un *smart contract* sur un *rollup* n'équivaut pas absolument à un déploiement sur Ethereum en termes de sécurité : les parties ajoutent le risque, même minime, que le *rollup* soit bogue ou exploité, en sus

⁹⁸⁹ V., *infra*, §80 et §84

⁹⁹⁰ V., *infra*, §335

⁹⁹¹ V., *infra*, §357

⁹⁹² Il s'agit de l'endroit où sont postées les soldes issus des opérations effectuées dans la *blockchain*.

des risques classiques de déploiement d'un *smart contract* sur une *blockchain*⁹⁹³. De plus, les *rollup* n'en sont encore à leur début d'existence au moment où nous écrivons ces lignes, ils sont donc tous dotés d'éléments de centralisation qui permettent à leurs équipes conceptrices de les faire évoluer unilatéralement⁹⁹⁴. Autrement dit, ils sont encore sous le contrôle d'une poignée d'individus qui peuvent être piratées⁹⁹⁵.

391. Contrats adéquats pour une exécution sur un *rollup* généraliste sur Ethereum.

Nous recommandons aux parties d'utiliser un *rollup* connecté à Ethereum, et en particulier *Arbitrum*, comme infrastructure principale d'exécution de leurs contrats intelligents. Les qualités d'un *rollup* généraliste (performance sans sacrifice de la sécurité et la décentralisation) en font en effet une infrastructure idéale pour accueillir, par défaut, tous les contrats intelligents mettant en œuvre des processus ordinaires : c'est-à-dire les processus de transferts d'actifs ne s'exécutant pas sur une durée très longue⁹⁹⁶.

Par principe donc, les parties seront avisées de déployer leur contrat intelligent sur un *rollup* généraliste comme *Arbitrum*, et si leur contrat est à très fort enjeu et s'exécute sur une durée longue, le L1 Ethereum sera mieux indiqué. En revanche, si le contrat intelligent demande des adaptations spécifiques, les parties pourront choisir, par exception, de le faire exécuter sur une infrastructure du même type.

§ II - Les choix par exception

392. **Le choix de l'infrastructure spécifique.** Lorsqu'elles rechercheront à exécuter un contrat intelligent nécessitant des adaptations très particulières, les parties seront avisées d'utiliser une infrastructure d'exécution spécifique. Elles pourront choisir entre une blockchain *ad-hoc* (A) si

⁹⁹³ Sam Kessler. « Ethereum's Layer 2 Rollups Reduce Costs, but the Risks Are Underappreciated », 26 octobre 2022. <https://www.coindesk.com/tech/2022/10/26/ethereums-layer-2-rollups-speed-things-up-but-the-risks-are-underappreciated/>.

As with other rollups, another security risk when using Optimism and Arbitrum is that their core codebases – the Ethereum-based smart contracts that allow them to operate – are vulnerable to hacks like any other blockchain-based programs

⁹⁹⁴ V., *infra*, §359

⁹⁹⁵ Sam Kessler. « Ethereum's Layer 2 Rollups Reduce Costs, but the Risks Are Underappreciated », 26 octobre 2022. <https://www.coindesk.com/tech/2022/10/26/ethereums-layer-2-rollups-speed-things-up-but-the-risks-are-underappreciated/>.

On Arbitrum, only a select group of hand-picked operators are allowed to submit fraud proofs. Harry Kalodner, the co-founder of Arbitrum builders Offchain Labs, said in an interview the team aims to make it so anyone can submit proofs within the next six months. But for now, Arbitrum users need to trust Arbitrum and its curated group of validators to know their transactions will not be tampered with.

⁹⁹⁶ V., *infra*, §39

elles privilégient le contrôle à tout autre paramètre dans l'exécution de leur contrat ; ou un *rollup* spécifique (B) lorsqu'elles accordent une certaine importance à la résilience et l'interopérabilité.

A – Une *blockchain* privée

393. Atouts d'une *blockchain* privée pour un contrat intelligent. Les blockchains spécifiques sont des infrastructures personnalisables à l'envie par leur créateurs, qui peuvent implémenter leur propre mécanisme de consensus, langage de programmation et règles d'accessibilité. Le plus souvent elles prennent la forme de *blockchain* dites privées où le concepteur dispose d'un contrôle étendu sur son infrastructure. Ces caractéristiques en font un environnement attractif pour des parties qui souhaitent avoir une totale maîtrise sur la manière dont elles souhaitent s'exposer aux propriétés de la *blockchain*.

D'autre part, elle constitue une couche d'exécution où il est plus simple d'être en conformité avec des exigences législatives. En effet, la loi peut imposer des obligations qui s'accordent peu avec les propriétés des infrastructures généralistes publiques. Il devient alors tentant pour des parties de recourir à des alternatives plus centralisées afin de respecter, par exemple, des réglementations en terme de protection des données personnelles, de lutte contre le blanchiment et le financement du terrorisme, ou autres⁹⁹⁷.

En sus, dans une infrastructure *ad-hoc*, l'application unique qui y est hébergée ne partage pas les ressources de la *blockchain* avec d'autres⁹⁹⁸. Cela signifie que celle-ci pourra être très gourmande en ressources, sans forcément subir de baisse de performance. Imaginons que le contrat intelligent consiste en un service proposé par une partie professionnelle à des consommateurs (Uber sur *blockchain*, par exemple⁹⁹⁹) : ces contrats intelligents standardisés peuvent demander une exécution simultanée par des milliers d'individus. Pour gérer techniquement cette demande, le prestataire de service pourra opportunément vouloir mettre sur place une *blockchain* privée optimisée afin de

⁹⁹⁷ « What Is a Subnet? | Avalanche Dev Docs ». Consulté le 30 juin 2023.
<https://docs.avax.network/learn/avalanche/subnets-overview>.

Launch a Network Designed With Compliance In Mind | Avalanche's Subnet architecture makes regulatory compliance manageable. As mentioned above, a Subnet may require validators to meet a set of requirements. Some examples of requirements the creators of a Subnet may choose include: Validators must be located in a given country. | Validators must pass KYC/AML checks. | Validators must hold a certain license.

⁹⁹⁸ V., *infra*, §377

⁹⁹⁹ Takyar, Akash. « Uber Blockchain Platform|Blockchain in Ridesharing ». LeewayHertz - AI Development Company, 20 août 2018. <https://www.leewayhertz.com/blockchain-disrupting-uber-platform/>.

supporter un grand afflux de demandes.

Enfin comme évoqué, les blockchains privées disposent actuellement d'une certaine maturité¹⁰⁰⁰. Les parties pourront donc mobiliser relativement facilement des outils éprouvés et documentés par une large communauté afin de les construire efficacement et avec sécurité. Cosmos, Polkadot, Avalanche et Polygon figurent parmi les protocoles les plus connus fournissant des kits de développement de *blockchain*. Le meilleur, selon nous, est celui de Cosmos, avec qui de nombreuses *blockchain* privées performantes ont été construites ces dernières années¹⁰⁰¹.

394. Limites d'une *blockchain* privée pour un contrat intelligent. Comme une *blockchain* spécifique prend souvent la forme d'une *blockchain* privée¹⁰⁰², elle est, par définition, centralisée et donc peu résiliente. Si des parties utilisent une telle forme de *blockchain*, elles doivent avoir conscience que leur programme n'aura pas une garantie de vie beaucoup supérieure à celle d'un programme ordinaire sur un serveur centralisé ; ce faisant elles perdront une des propositions de valeur essentielle d'une *blockchain*. En sus, le déploiement d'une *blockchain* privée, malgré les kits de développement à disposition, reste une entreprise bien plus fastidieuse, chronophage et coûteuse que celui du déploiement d'un *smart contract* sur une infrastructure généraliste (comme Ethereum ou un *rollup*). Il faudra donc que les parties s'assurent de l'opportunité d'une telle démarche par rapport à déployer ordinairement un programme sur une *blockchain* ou un serveur centralisé.

Enfin, une *blockchain* privée est par essence isolée. Autrement dit, les parties se privent, en y recourant, de la possibilité de s'interfacer librement et aisément avec une myriade d'autres applications déployées dans une *blockchain* publique, ce qui constitue aussi une des propositions de valeurs principales d'une *blockchain*. Concrètement, cela signifie que des parties n'auront pas, nativement, accès à des *stablecoin*, des applications de finance décentralisée ou des solutions d'oracles pré-faites¹⁰⁰³. Elles devront créer et utiliser des ponts vers d'autres blockchains où résident

¹⁰⁰⁰ V., *infra*, §376

¹⁰⁰¹ dYdX fait partie des très grandes entreprises qui ont fait le choix de Cosmos pour lancer leur *blockchain* privée.

dYdX. « Announcing dYdX Chain », 22 juin 2022. <https://dydx.exchange/blog/dydx-chain>.

¹⁰⁰² V., *infra*, §374

¹⁰⁰³ Infura Blog | Tutorials, Case Studies, News, Feature Announcements. « The Benefits and Tradeoffs of Application-Specific Blockchains », 17 janvier 2023. <https://blog.infura.io/post/the-benefits-and-tradeoffs-of-application-specific-blockchains>.

Building an appchain reduces interoperability with other applications and breaks composability (with some exceptions, e.g., Polkadot/Cosmos). Users can still bridge funds from other chains, but atomicity (a quality of blockchains where all parts of a transaction succeed or the entire transaction fails) is lost. As a rule, atomic transactions (e.g., taking out a flash loan to buy tokens on a DEX) work when all applications involved live on the same settlement layer.

ces applications pour y connecter leur programme ; ce qui est une opération lourde en plus d'être porteuse de risques.

395. Contrats adéquats pour une exécution dans une *blockchain* privée. Ces éléments nous permettent de dire que les parties seront avisées de recourir à une *blockchain* privée pour des projets de formalisation contractuelle qui remplissent les conditions cumulatives suivantes :

- le contrat intelligent n'a pas ou très peu besoin d'autres applications décentralisées pour fonctionner,
- il demande une performance excellente,
- et surtout, il nécessite un contrôle maximal d'une ou plusieurs des parties sur le programme exécutant, notamment pour des raisons légales et/ou de sécurité.

Une *blockchain* privée constitue donc une infrastructure par exception, devant n'être utilisée que pour exécuter des programmes sur lesquels les parties ont essentiellement besoin d'avoir un large contrôle. Nous remarquons que les contrats de services publics (ou ceux s'en approchant) se prêtent bien à une exécution sur ce type d'infrastructure. Il s'agit en effet de relations dans lesquelles les services objets de ces contrats peuvent être beaucoup sollicités (ce qui demande une infrastructure performante), ces services n'ont pas ou peu besoin d'être connectés à d'autres blockchains et étant donné la crucialité des services fournis, il est justifié qu'ils soient sous la mainmise des prestataires de ces services (les pouvoirs publics). Voici une liste d'exemple de contrats convenant pour être exécutés sur de telles infrastructures :

- le transport public : chaque fois qu'un usager prend un transport en commun, il contracte avec la SNCF qui est une entreprise publique. Il s'agit d'une relation qui est régulièrement citée comme exemple de convention dont les processus pourraient être formalisés par des smart contracts : pour le paiement automatisé de tickets, ou encore leur remboursement automatique en cas de retard¹⁰⁰⁴. Pour ce service, une *blockchain* privée pourrait être constituée par la SNCF. Elle serait à même de supporter une intense sollicitation, tout en restant sous son contrôle. En même temps, elle apporterait une transparence et une fluidité d'exécution bienvenue à l'exécution de ces contrats ;
- la fourniture d'énergie : semblablement au transport, les contrats entre des usagers et les fournisseurs d'énergie pourraient aussi être automatisés dans une *blockchain* privée; notamment pour le paiement de la consommation d'électricité, la revente éventuelle des

¹⁰⁰⁴ Enescu, Florentina, Fernando Birleanu, Maria Raboaca, Nicu Bizon, et Phatiphat Thounthong. « A Review of the Public Transport Services Based on the Blockchain Technology ». *Sustainability* 14 (12 octobre 2022): 13027. <https://doi.org/10.3390/su142013027>.

réserves non consommées, ou encore l'indemnisation en cas de défaut d'approvisionnement¹⁰⁰⁵ ;

- le péage des autoroutes constitue un autre cas d'usage cité comme automatisable dans une *blockchain*, et pour lequel nous pensons qu'une *blockchain ad-hoc* constituerait une infrastructure d'exécution de choix : elle pourrait supporter le paiement automatique des frais dès lorsqu'une voiture franchirait la barrière d'une route à péage¹⁰⁰⁶.

B – Un *rollup* spécifique sur Ethereum

396. Atouts d'un *rollup* spécifique sur Ethereum pour un contrat intelligent. Un *rollup* spécifique sur Ethereum présente presque toutes les qualités déjà évoquées d'une *blockchain ad-hoc* : les parties peuvent le personnaliser à leur convenance pour qu'il satisfasse tous leurs besoins. L'atout des *rollup* spécifiques sur leur équivalent *blockchain* néanmoins est que ces derniers sont nativement connectés à des L1 comme Ethereum, qui sont décentralisés et matures. Cela signifie que les parties recourant à ces solutions pourront, beaucoup plus facilement que sur des *blockchain*s privées, s'interfacer avec les nombreuses applications déployés dans l'écosystème d'Ethereum. En d'autres termes, l'infrastructure d'exécution, bien que personnalisée et contrôlée dans une grande mesure, ne sera pas aussi isolée qu'une *blockchain* spécifique.

Autre *plus-value* d'un *rollup* spécifique est le fait que celui-ci hérite sa sécurité de Ethereum : c'est une *blockchain* résiliente en plus d'être personnalisable. Il s'agit d'une propriété particulièrement intéressante car elle libère les parties du choix habituel qu'elles doivent opérer entre personnalisation et décentralisation. Les *rollup* spécifiques constituent donc une infrastructure extrêmement intéressante pour des parties souhaitant recourir à une *blockchain* dans laquelle elles ont un contrôle étendu, sans l'extirper de ses propriétés les plus essentielles.

397. Limites d'un *rollup* spécifique sur Ethereum pour un contrat intelligent. Reste que ces solutions ne bénéficient pas encore de la même maturité technique que leurs équivalents L1. Il n'existe pas de kits de développements aussi éprouvés et prêts à l'emploi que ceux proposés, par exemple, par *Cosmos* pour mettre sur pied rapidement des *rollup* spécifiques. Les parties prendront donc significativement plus de temps à créer leur propre *rollup* qu'un L1 *ad-hoc*. Il faut ajouter que

¹⁰⁰⁵ Kirli, Desen, Benoit Couraud, Valentin Robu, Marcelo Salgado-Bravo, Sonam Norbu, Merlinda Andoni, Ioannis Antonopoulos, Matias Negrete-Pincetic, David Flynn, et Aristides Kiprakis. « Smart contracts in Energy Systems: A Systematic Review of Fundamental Approaches and Implementations ». *Renewable and Sustainable Energy Reviews* 158 (1 avril 2022): 112013. <https://doi.org/10.1016/j.rser.2021.112013>.

¹⁰⁰⁶ Tanveer, Hasnain, et Nadeem Javaid. *Using Ethereum Blockchain Technology for Road Toll Collection on Highways*, 2019.

les *rollup* spécifiques n'offrent pas la même étendue de personnalisation et contrôle que les blockchains privées : le mécanisme de consensus sera, par définition, celui hérité du L1.

398. Contrats adéquats pour une exécution sur un *rollup* spécifique. Comme pour les *rollup* généralistes, nous estimons que les *rollup* spécifiques doivent être la solution privilégiée ou par défaut des parties ayant un projet de formalisation contractuelle spécial. L'infrastructure offre en effet un degré de personnalisation et contrôle important sans compromettre sur la composabilité et la résilience, qui constituent l'essentiel de l'intérêt de recourir à une *blockchain* en premier lieu. Même si le contrôle qu'elle donne n'est pas aussi absolu que dans une *blockchain* privée, il reste satisfaisant pour la plupart des besoins des parties.

Un exemple de contrat pertinent à exécuter sur un *rollup* spécifique serait les statuts de société. Actuellement, ces contrats sont exécutés dans des blockchains privées¹⁰⁰⁷, au motif que seules celles-ci permettraient de répondre aux exigences réglementaires notamment celles de l'article R211-9-7 du code monétaire et financier¹⁰⁰⁸. Même si il est possible de coder ses processus dans un *rollup* généraliste, les parties prenantes peuvent légitimement souhaiter avoir un meilleur contrôle sur l'infrastructure hébergeant leur programme, notamment afin de garder confidentielles certaines données relatives aux associées et garantir au maximum sa sécurité en maîtrisant l'accès au programme.

L'inconvénient majeur de choisir une *blockchain* privée pour héberger un tel contrat intelligent est que celui-ci ne sera pas plus pérenne que le L1 centralisé sur lequel il réside. Or, une société ayant vocation à durer, les parties font face à un besoin impératif de résilience : il faut qu'elles soient assurées que le programme exécutant leurs processus statutaires pourra survivre pendant des années, comme leur société. Si le programme réside sur un L1 privée, tenu par quelques entités, sa durée de vie sera celle de cette poignée d'individus. Un *rollup* spécifique répond alors de manière élégante au problème : le programme est aussi résilient que Ethereum mais les parties conservent tout de même

¹⁰⁰⁷ Livia Chartrain. « Quelle blockchain a choisi MonJuridique ? » Consulté le 5 juillet 2023. <https://monjuridique.infogreffe.fr/blog/blockchain-monjuridique>.

En synthèse, Monjuridique.infogreffe a fait le choix d'une blockchain d'entreprise pour répondre à des besoins professionnels. Il s'agit d'une blockchain privée qui permet de bénéficier des apports de la blockchain en matière de preuve et d'opposabilité juridique, tout en répondant aux exigences de sécurité et de confidentialité du monde économique traditionnel.

¹⁰⁰⁸ **Article R211-9-7 du code monétaire et financier** : *Le dispositif d'enregistrement électronique partagé mentionné au deuxième alinéa de l'article L. 211-7 est conçu et mis en œuvre de façon à garantir l'enregistrement et l'intégrité des inscriptions et à permettre, directement ou indirectement, d'identifier les propriétaires des titres, la nature et le nombre de titres détenus. Les inscriptions réalisées dans ce dispositif d'enregistrement font l'objet d'un plan de continuité d'activité actualisé comprenant notamment un dispositif externe de conservation périodique des données. Lorsque des titres sont inscrits dans ce dispositif d'enregistrement, le propriétaire de ces titres peut disposer de relevés des opérations qui lui sont propres.*

un contrôle étendu celui-ci leur permettant de respecter toutes les obligations que la loi leur impose.

399. Conclusion du chapitre. Tableau récapitulatif du choix des infrastructures d'exécution

Infrastructure d'exécution	Type de contrats intelligents	Exemples de contrats intelligents
Ethereum (L1 généraliste)	Contrats à très fort enjeu, mettant en œuvre des processus ordinaires et ayant une durée de vie longue.	Contrats de séquestre logiciel, clause testamentaire, contrats d'assurance-vie.
Arbitrum (<i>rollup</i> généraliste)	Contrats mettant en œuvre des processus ordinaires, ayant une durée de vie de quelques années maximum.	Convient pour la plupart des contrats
Cosmos (L1 spécifique)	Contrats aux besoins spécifiques, nécessitant un contrôle maximal sur toutes ses composantes (pour des raisons légales et/ou de sécurité) et n'ayant pas besoin d'être connectés à d'autres applications dans une <i>blockchain</i> .	Contrats de transport public, de fourniture d'énergie.
StarkEx (<i>rollup</i> spécifique)	Contrats aux besoins spécifiques, nécessitant un contrôle assez élevé sur certaines de ses composantes (pour des raisons réglementaires et/ou de sécurité) mais ayant besoin d'être lié à d'autres applications dans une <i>blockchain</i> et/ou de bénéficier d'une grande résilience.	Statuts de SAS

L'infrastructure d'exécution déterminée, les parties pourront passer à l'étape du développement de des programmes exécutant leur contrats intelligents.

Chapitre II – L’écriture des smart contracts

400. Bases et modèles des smart contracts. Comme évoqué, Ethereum ou un *rollup* connecté à celui-ci, doivent constituer les principales infrastructures d’exécution des contrats intelligents des parties. Ce sont sur celles-ci que les parties pourront déployer des smart contracts qui seront chargés d’exécuter certains des processus stipulés de leurs contrats *fiat*. Aussi, il s’agira pour nous de présenter les bases nécessaires au développement de ces programmes (Section I), avant de proposer des modèles de *smart contract* que des parties pourront utiliser pour leur propre implémentation de contrats intelligents (Section II).

Section I – Prolégomènes

401. Cadre de développement des smart contracts. Avant d’aborder les smart contracts en eux-mêmes (§2), il nous faut décrire le « stack technique »¹⁰⁰⁹ à disposition des parties pour les développer (§1).

§ I - Présentation du *stack technique*

402. Définition du *stack technique*. Nous nommons « stack technique » l’ensemble des technologies utiles pour créer et suivre les programmes exécutant les contrats des parties. Nous pouvons classer ces outils entre ceux visant à créer et interagir avec des smart contracts (A), et ceux servant à stocker et suivre des données issues de ces derniers (B).

A - Nécessaire de création et d’interaction avec des smart contracts

403. Etapes de réalisation d’un *smart contract*. Le processus courant du développement d’un *smart contract* est le suivant :

¹⁰⁰⁹ Jaspard, Audrey. « Qu’est-ce qu’une tech stack ? Définition et exemples ». Consulté le 5 juillet 2023. <https://blog.hubspot.fr/marketing/tech-stack>.

Aussi appelée « technology stack », « écosystème de données » ou encore « pile de technologies », la tech stack est l’environnement technique d’un logiciel ou d’un programme informatique. Elle comprend le ou les langages de programmation utilisés, mais aussi les frameworks et les outils auxquels les développeurs ont recours pour communiquer avec l’application.

- le programme est d'abord codé par le développeur à l'aide d'un langage de programmation ;
- il est ensuite déployé dans une *blockchain* via un « nœud »¹⁰¹⁰, à l'aide d'un *wallet* ;
- une fois le *smart contract* déployé dans la *blockchain*, il peut être interagi avec lui directement ou au travers d'un site web ou d'une application mobile. Lorsque le développeur choisit ces deux derniers modes d'interaction, il est dit qu'il développe une « application décentralisée » (Dapp).¹⁰¹¹

A chaque étape de ce processus de développement d'un *smart contract*, les parties peuvent mobiliser différents outils, nombreux et efficaces, de l'écosystème d'Ethereum.

a) Outils pour le codage des smart contracts

404. Les langages de programmation. Un programme est une suite d'instructions exprimées dans un langage compréhensible par un ordinateur afin que celui-ci les exécute¹⁰¹². Les instructions que la *blockchain* ou un *rollup* Ethereum comprennent sont l'EVM *bytecodes*. Pour qu'elles lui soient communiquées, il est nécessaire au préalable d'écrire le programme dans un idiome spécifique, un langage de programmation, puis de le traduire (on parle plutôt de le compiler¹⁰¹³), afin que la machine virtuelle d'Ethereum exécute les ordres qui lui ont été données.

Les deux principaux langages de programmation pour écrire des smart contracts compréhensibles par l'EVM sont *Solidity* et *Vyper*. Le second est bien moins populaire que le premier et est inspiré du langage de programmation *Python*. Il permet de coder des smart contracts de manière peu verbal : cela signifie qu'il n'y a pas besoin d'écrire beaucoup pour exprimer une instruction. Il est également

¹⁰¹⁰ V., *infra*, §2

¹⁰¹¹ ethereum.org. « Introduction to Dapps ». Consulté le 5 juillet 2023. <https://ethereum.org>.

A decentralized application (dapp) is an application built on a decentralized network that combines a smart contract and a frontend user interface.

¹⁰¹² Techno-Science.net. «  Programme informatique : définition et explications ». Techno-Science.net. Consulté le 5 juillet 2023. <https://www.techno-science.net/definition/5403.html>.

Un programme informatique est une liste d'ordres indiquant à un ordinateur ce qu'il doit faire. Il se présente sous la forme d'une ou plusieurs séquences d'instructions, comportant souvent des données de base, devant être exécutées dans un certain ordre par un processeur ou par processus informatique (cas des systèmes multitâches).

¹⁰¹³ « What Is Compilation? » Consulté le 12 juillet 2023. <https://www.computerhope.com/jargon/c/compilat.htm>.

Compilation is the process the computer takes to convert a high-level programming language into a machine language that the computer can understand. The software which performs this conversion is called a compiler.

mieux optimisé que *Solidity* : il contient moins de fonctionnalités pour une efficacité équivalente. Ces propriétés le rendrait assez sécurisé car plus simple à auditer et bénéficiant d'une surface de bogues et d'attaques réduite¹⁰¹⁴.

Solidity est, quant à lui, inspiré des langages *Java*, *C* et *Javascript*. Il est, de très loin, le langage de programmation le plus utilisé pour écrire des smart contracts et dispose donc de la communauté de développeurs la plus large¹⁰¹⁵. A ce titre, il bénéficie de nombreux standards, bibliothèques *open-source*, kits de développement et d'autres outils facilitant son exploitation. Comme évoqué, ce sont ces éléments qui participent à créer efficacement des programmes sécurisés¹⁰¹⁶ et c'est la raison pour laquelle nous recommandons aux parties de choisir ce langage pour le codage de leur smart contracts. Nos développements seront donc proposés en *Solidity*.

405. Outils de développement. Les parties peuvent utiliser des outils de développements leur permettant de coder, compiler, déboguer et déployer des smart contracts en *Solidity*. Au moment où sont écrites ces lignes, *Truffle*, *Hardhat*, *Remix* et *Foundry* figurent parmi les plus connus et utilisés dans l'écosystème de l'EVM¹⁰¹⁷. Le choix de l'outil dépendra des préférences des parties. Nous conseillons à celles privilégiant une solution en ligne et facile d'utilisation de choisir *Remix*¹⁰¹⁸ et aux autres mettant l'accent sur la performance et la complétude de l'outil de privilégier *Foundry* ou *Hardhat*¹⁰¹⁹.

406. OpenZeppelin. *OpenZeppelin* est une organisation connue dans l'écosystème

¹⁰¹⁴ Chainlink. « Solidity vs. Vyper: Which Smart contract Language Is Right for Me? » Chainlink Blog, 17 octobre 2022. <https://blog.chain.link/solidity-vs-vyper/>.

Vyper is a contract-oriented, pythonic programming language also designed for the EVM. Vyper was designed to improve upon Solidity by aiming to enhance readability and limit certain practices. On a high level, Vyper seeks to optimize the security and auditability of smart contracts.

¹⁰¹⁵ Chainlink. « Solidity vs. Vyper: Which Smart contract Language Is Right for Me? » Chainlink Blog, 17 octobre 2022. <https://blog.chain.link/solidity-vs-vyper/>.

According to DefiLlama, as of right now, in the DeFi space, Solidity smart contracts secure 87% of TVL, while Vyper smart contracts secure 8%.

¹⁰¹⁶ V., *infra*, §387

¹⁰¹⁷ « Remix vs Truffle vs Hardhat vs Foundry - Become Ethereum Blockchain Developer ». Consulté le 6 juillet 2023. <https://ethereum-blockchain-developer.com/124-remix-vs-truffle-vs-hardhat-vs-foundry/00-overview/>.

¹⁰¹⁸ Il s'agit d'un environnement de développement intégré en ligne et particulièrement ergonomique pour les smart contracts. Il est prisé par les débutants.

¹⁰¹⁹ Les deux sont des toolkit sortis bien après *Remix*. *Foundry* est connu pour sa performance et vitesse d'exécution tandis que *Hardhat* est prisé pour la complétude du panel qu'il offre.

Ethereum fournissant la plus large bibliothèque *open-source* de *smart contract* en *solidity*.¹⁰²⁰ Leurs programmes, accessibles librement, ont la particularité d'être codés par des développeurs aguerris et surtout audités par une large communauté d'entre eux. De sorte qu'ils constituent bien souvent des programmes dont n'hésitent pas s'emparer des porteurs de projets pour développer leurs applications en toute quiétude. Les parties seront donc avisées de recourir, toutes les fois qu'elles le peuvent, tant pour des raisons de sécurité que de productivité, aux smart contracts de *OpenZeppelin*. Malgré cela, il n'existe pas de garantie que ces programmes soient exempts de bogues et tout à fait adaptés aux usages des parties¹⁰²¹. Pourtant nous pensons que les parties tireront bien plus d'avantages qu'elles ne courront de risques à y recourir.

b) Outils pour le déploiement et l'interaction avec des smart contracts

407. Les portefeuilles de crypto (wallet). Un *wallet* est un applicatif permettant de gérer son identité et ses cryptoactifs dans une *blockchain*.¹⁰²² En principe, il est le nom donné à tout logiciel stockant une paire de clefs reliées cryptographiquement entre elles permettant :

- pour la clef qu'on appelle « publique », d'identifier un individu dans la *blockchain* ; elle peut être librement communiquée afin notamment de recevoir des cryptoactifs ;
- pour la clef qu'on dit « privée », qui est confidentielle, de signer des transactions afin d'effectuer toute interaction dans la *blockchain*.

408. Types et typologies de wallet. Il existe différents types de *wallet* et de typologies de

¹⁰²⁰ Blog, Moralis. « What Is OpenZeppelin? The Ultimate Guide ». Moralis Web3 | Enterprise-Grade Web3 APIs, 13 novembre 2021. <https://moralis.io/what-is-openzeppelin-the-ultimate-guide/>.

OpenZeppelin is a set of vetted smart contracts. It helps you put precautionary security measures in place for your Web3 apps. Using its audit services, you can ensure your practices will conform to a set of established standards. That way, criminals will not compromise the security of your system.

¹⁰²¹ Kiffer, Lucianna, Dave Levin, et Alan Mislove. « Analyzing Ethereum's Contract Topology », 494-99, 2018. <https://doi.org/10.1145/3278532.3278575>.

This high level of code reuse for user-created contracts suggests users are obtaining their code from a small set of locations, including that if bugs exist in these contracts, the effects could be widespread.

¹⁰²² « MetaMask Learn | What Is a Crypto Wallet? » Consulté le 6 juillet 2023. <https://learn.metamask.io/lessons/what-is-a-crypto-wallet>.

Crypto wallets are a form of digital wallet designed for Web3. They help you manage permissions with whom you share your data, store cryptocurrency, NFTs, and more (...). Your wallet is a means for storing and managing your identity, represented by digital keys. You need these keys to do anything on a blockchain—connect to a dapp, send or receive crypto, buy or sell NFTs, etc. Think of your wallet as a Web3 permissions manager, where you grant access to the apps that you want to use.

wallet. Traditionnellement, il est fait une distinction entre les *hot wallet* et les *cold wallet* :

- les *hot wallet* sont des portefeuilles connectés à internet, qui peuvent prendre la forme d'applications, logiciels ou extensions de navigateur. Ils sont de loin les plus commodes à utiliser et donc les plus populaires¹⁰²³, mais aussi les plus vulnérables aux attaques pirates¹⁰²⁴. Dans cette catégorie de portefeuilles, on fait encore la distinction entre les *wallet* dits *custodial* et *non-custodial*. Dans les premiers, les propriétaires des *wallet* ne sont pas en possession de la paire de clefs cryptographiques précédemment évoquée ; celle-ci est détenue par un prestataire leur mettant à disposition le *wallet*. Celui-ci réalise au nom et pour le compte de son client les interactions que ce dernier souhaite effectuer¹⁰²⁵. La situation des individus ayant ce type de portefeuille est donc très similaire à celle d'une personne ayant de l'argent placé dans un compte en banque : elle n'a pas de véritable souveraineté sur ses cryptoactifs et dispose tout au plus d'un droit de créance sur ceux-ci¹⁰²⁶. Les *wallet non-custodial* sont, *a contrario*, des portefeuilles où leurs possesseurs sont en possession de la paire de clefs cryptographiques et de toute la souveraineté qui y est associée ;
- les *cold wallet* consistent en des dispositifs physiques (prenant souvent la forme de clefs *usb*)

¹⁰²³ Investopedia. « Hot Wallet vs. Cold Wallet ». Consulté le 6 juillet 2023. <https://www.investopedia.com/hot-wallet-vs-cold-wallet-7098461>.

Hot crypto wallets are connected to the internet. They offer a series of features, from storing, sending, and receiving tokens to managing and viewing all available tokens in one place. Hot wallets are accessible from internet-enabled devices such as cellular phones, tablets, and laptops. Hot wallets have been widely adopted because of the ease of transferring and receiving funds on demand.

¹⁰²⁴ Gemini. « Crypto Wallets: Hot vs. Cold Wallets ». Consulté le 6 juillet 2023. <https://www.gemini.com/cryptopedia/crypto-wallets-hot-cold>

A hot wallet is connected to the internet and could be vulnerable to online attacks — which could lead to stolen funds — but it's faster and makes it easier to trade or spend crypto.

¹⁰²⁵ BitPay Blog. « Custodial vs Non-Custodial Wallet - What's the Difference? | BitPay », 24 mai 2023. <https://bitpay.com/blog/non-custodial-wallets-vs-custodial-wallets/>.

Custodial wallets are nearly always web-based, and are usually provided by centralized crypto exchanges like Coinbase (...). With a custodial wallet, a user initiates a transaction through their platform of choice and selects a wallet address to which they'd like to send funds. The custodian of the private key, in this case a crypto exchange, is tasked with "signing" transactions using the private key to ensure they're completed correctly.

¹⁰²⁶ Dans un compte courant bancaire, le titulaire du compte est en fait créancier des sommes inscrites sur celui à l'égard de l'établissement bancaire.

Debard, Thierry, et Serge Guinchard. Lexique des termes juridiques 2020-2021 - 28e ed. Edition 2020-2021. Dalloz, 2020.

Compte courant : Convention par laquelle deux personnes qui sont périodiquement créanciers et débitrices réciproques, font figurer leurs créances et dettes en articles de compte indivisible, seul le solde étant dû après clôture.

dans lesquels est stockée la paire de clefs cryptographiques (ils sont donc par défaut *non - custodial*). Ces engins sont, à la différence des *hot wallet*, complètement déconnectés d'internet. Cela signifie que ce réseau n'est pas, a priori, un vecteur d'attaque pour leurs possesseurs. Donc ils sont, dans cette mesure-là, plus sécurisés que les *hot wallet*. Mais si l'avantage des *cold wallet* serait leur sécurité, leur inconvénient est leur incommodité : pour toute interaction avec la *blockchain*, il est nécessaire de manuellement connecter son dispositif à son ordinateur¹⁰²⁷;

- à côté de ces deux catégories de portefeuilles, a émergé une troisième catégorie relativement récente qu'on appelle les *smart wallet*. Ils consistent en des smart contracts capables de reproduire toutes les fonctionnalités d'un *wallet* ordinaire (identification, réception et envoi de cryptoactifs) et d'autres plus complexes. Par exemple, il est possible de coder une limite sur le montant maximal de cryptoactifs transférables ou encore de créer ses propres listes noires et blanches d'adresses susceptibles de recevoir des cryptoactifs¹⁰²⁸.

Nous recommandons aux parties d'utiliser des *smart wallet*, puisque ces derniers leur permettent de neutraliser le principal risque de compromission du *wallet* de manière décentralisée et commode. En effet, dans un *hot* ou *cold wallet non custodial*, les parties doivent conserver précieusement leur clef privée (sous forme de phrase mnémonique¹⁰²⁹) afin de la recouvrer en cas d'égarement ou de piratage. Cela constitue un risque sécuritaire considérable : si une personne perd cette information, toute son identité et ses cryptoactifs sont également perdus. Or dans un *smart wallet*, les utilisateurs bénéficient d'une fonctionnalité leur permettant, en cas de perte de leur clef privée, de confier à des individus

¹⁰²⁷ Gemini. « Crypto Wallets: Hot vs. Cold Wallets ». Consulté le 6 juillet 2023.
<https://www.gemini.com/cryptopedia/crypto-wallets-hot-cold>.

Generally, cold storage wallets are quite secure. Stealing from a cold wallet usually would require physical possession of or access to the cold wallet, as well as any associated PINs or passwords that must be used to access the funds. Most hardware wallets are cold wallets and live on devices that look like a small to medium-sized USB stick.

¹⁰²⁸ Rajeev Gopalakrishna. « Account Abstraction (EIP-2938): Why & What ». Our Status, 19 novembre 2020.
<https://our.status.im/account-abstraction-eip-2938/>.

Smart contract wallets are implemented on-chain via smart contracts (as the name suggests). Such wallets offer programmable risk mitigation and user-friendly experience by implementing features such as multi-signature security, social or time-based recovery, rate-limiting of transactions or amounts, allow/deny-list of addresses, gasless meta-transactions and batched transactions.

¹⁰²⁹ G, Thomas. « Comment fonctionnent les phrases mnémoniques (seed) ? » Journal du Coin, 13 avril 2018.
<https://journalducoin.com/actualites/phrases-mnemoniques/>.

Phrase mnémonique : série de mots utilisée pour prouver la possession de clés et adresses servant aux transactions cryptographiques. Aussi appelé seed.

qu'ils ont nommées la tâche de choisir (dans un *smart contract*) quelle sera la nouvelle clef privée du *smart wallet*. Ce faisant, les propriétaires n'ont pas à conserver leur clef privée et peuvent recouvrir la perte de leur *wallet* grâce à une liste de personnes de confiance qu'ils auront choisi¹⁰³⁰.

Ils peuvent utiliser d'autres mécanismes de ce style pour sécuriser les interactions avec une *blockchain*. Par exemple, celui de multi-signature permet de conditionner toute interaction avec une *blockchain* à l'approbation préalable d'un seuil de *n* personnes désignées par le propriétaire du *wallet*. Ces mécanismes constituent autant de fonctionnalités qui permettent de ne pas concentrer le risque de compromission sur une seule et unique personne, tout étant *non-custodial* (puisque cela est mis en œuvre par un *smart contract* déployé sur une infrastructure décentralisée). Les parties seront donc avisées de recourir aux *smart wallet* comme ceux de l'entreprise *Argent* ou de *Gnosis*.¹⁰³¹

409. Les nœuds. Comme déjà évoqué¹⁰³², toute interaction avec une *blockchain* se fait à travers un nœud. Ce sont ces structures qui constituent le réseau d'une *blockchain* et sans lesquelles on ne peut ni déployer ni interagir avec des *smart contracts*.¹⁰³³ Or, en pratique, détenir et maintenir un nœud est une opération très fastidieuse qu'on préfère externaliser à un tiers spécialiste. Ces derniers tiennent alors plusieurs nœuds qu'ils mettent à disposition des développeurs de *smart contract* par le biais d'une API, leur permettant de ne se préoccuper que du développement de leurs programmes.

Au moment où sont écrites ces lignes, *Infura* et *Alchemy* semblent être les deux meilleurs fournisseurs de ce service dans l'écosystème Ethereum. Mais le gain en commodité obtenu par leur service ne se

¹⁰³⁰ Vitalik Buterin. « Why we need wide adoption of social recovery wallets », 11 janvier 2021. <https://vitalik.ca/general/2021/01/11/recovery.html>.

This gets us to my preferred method for securing a wallet: social recovery. A social recovery system works as follows: There is a single "signing key" that can be used to approve transactions. There is a set of at least 3 (or a much higher number) of "guardians", of which a majority can cooperate to change the signing key of the account. The signing key has the ability to add or remove guardians, though only after a delay (often 1-3 days).

¹⁰³¹ Kelly, Decrypt / Liam J. « Making Ethereum Wallets Smarter Is the Next Challenge—and Visa Is Among Those Working on It ». Decrypt, 22 mars 2023. <https://decrypt.co/124100/making-crypto-wallets-smarter-ethereum>.

Argent, along with Safe (formerly Gnosis Safe), is at the forefront of the account abstraction movement, making crypto wallets easier, and smarter, to use.

¹⁰³² V., *infra*, §403

¹⁰³³ Ethereum.org. « Nodes and Clients ». Consulté le 7 juillet 2023. <https://ethereum.org>.

Ethereum is a distributed network of computers (known as nodes) running software that can verify blocks and transaction data. The software application, known as a client, must be run on your computer to turn it into an Ethereum node (...). A "node" is any instance of Ethereum client software that is connected to other computers also running Ethereum software, forming a network. A client is an implementation of Ethereum that verifies data against the protocol rules and keeps the network secure.

fait pas sans risques. En effet, en y recourant, des parties introduisent un tiers centralisé dans leur démarche qui peut :

- refuser le déploiement de leurs smart contracts¹⁰³⁴,
- refuser l'interaction avec certains smart contracts,
- subir une interruption ou un défaut de service¹⁰³⁵.

410. Elles devront donc arbitrer entre la commodité procurée par l'utilisation des services de ces tiers et la souveraineté (couplée à une certaine garantie de sécurité) procurée par le maintien de leurs propres nœuds. En pratique, interagir directement avec des smart contracts à travers un nœud demande un niveau de professionnalisme et d'investissement (en tous cas actuellement) que des parties assez profanes ou peu investies dans la *blockchain* préféreront éviter¹⁰³⁶. Celles-ci seront donc avisées d'utiliser les services comme *Infura*¹⁰³⁷ ou *Alchemy*¹⁰³⁸. A l'inverse, celles aguerries et/ou investies dans la *blockchain* (en raison, par exemple, de l'exécution *on-chain* de processus à fort enjeu¹⁰³⁹, préféreront avoir le plus de maîtrise possible en disposant de leur propre infrastructure de communication avec elle.

¹⁰³⁴ Mark Simon. « MetaMask and Infura: The Centralised Infrastructure Behind Crypto ». HelloCrypto (blog), 19 septembre 2022. <https://hellocrypto.com/article/metamask-and-infura-the-centralised-infrastructure-behind-crypto/>.

In response to US sanctions in March of 2022, Infura blocked off a lot of users in Venezuela, Iran and Lebanon, from the network.

¹⁰³⁵ Hayward, Decrypt / Andrew. « MetaMask, Ethereum Apps Down as Infura Suffers Outage ». Decrypt, 22 avril 2022. <https://decrypt.co/98457/metamask-ethereum-apps-down-infura-outage/>.

Infura is a node provider; allowing people to use existing roads and infrastructure connected to the Ethereum Virtual Machine without having to rebuild it every single time. Providing these services to developers allows them to focus on the apps and smart contracts, making it very easy to build new decentralised apps and protocols. It means that you don't have to set up your own node to connect to the blockchain or host a service. Since Infura built all the roads to Ethereum, it can also control the traffic lights to prevent congestion.

¹⁰³⁶ Mark Simon. « MetaMask and Infura: The Centralised Infrastructure Behind Crypto ». HelloCrypto (blog), 19 septembre 2022. <https://hellocrypto.com/article/metamask-and-infura-the-centralised-infrastructure-behind-crypto/>.

Infura is a node provider; allowing people to use existing roads and infrastructure connected to the Ethereum Virtual Machine without having to rebuild it every single time. Providing these services to developers allows them to focus on the apps and smart contracts, making it very easy to build new decentralised apps and protocols. It means that you don't have to set up your own node to connect to the blockchain or host a service.

¹⁰³⁷ <https://www.infura.io/>

¹⁰³⁸ <https://www.alchemy.com/>

¹⁰³⁹ V., *infra*, §388

c) Outils pour la création d'une *Dapp*

411. Bibliothèques d'applications décentralisées. Il arrive très régulièrement que les services fournis par le biais de *smart contract* prennent la forme de *Dapp*. Les parties (ou l'une d'entre elles) peuvent donc choisir de créer une application web à travers laquelle elles interagiront commodément avec leurs programmes, en particulier si l'une d'elles est une profane. Pour ce faire, elles auront à leur disposition un certain nombre d'outils leur permettant de faciliter cette communication entre leur site web et leurs smart contracts.

Ces derniers sont appelés, « bibliothèques web 3 »¹⁰⁴⁰, et les deux plus connues de l'écosystème Ethereum sont, actuellement, *web3.js* et *ethers*. Même si le choix de ces bibliothèques dépendra des affinités personnelles des parties en charge du développement, nous recommandons l'usage de *ethers* pour sa prise en main et documentation supérieure à celle de *web3.js*. De manière générale, nous conseillons aux parties d'exploiter l'écosystème *JavaScript* et en particulier de recourir à ces *framework*¹⁰⁴¹ les plus connus comme *React*¹⁰⁴². Le développement de ce type de site s'est surtout construit autour de ces outils : ce qui signifie qu'ils bénéficient du support communautaire le plus large ; ce qui est gage de sécurité et productivité pour un développeur.

B – Nécessaire de suivi et de stockage

412. Une fois les smart contracts déployés, les parties peuvent mobiliser d'autres outils afin de suivre leur évolution (a) et stocker leur données (b).

a) Explorateurs et *subgraph*

413. Explorateurs de *blockchain*. Un autre élément important du *stack technique* des

¹⁰⁴⁰ Ethereum.org. « JavaScript API Libraries ». Consulté le 7 juillet 2023. <https://ethereum.org>.

In order for a web app to interact with the Ethereum blockchain (i.e. read blockchain data and/or send transactions to the network), it must connect to an Ethereum node. For this purpose, every Ethereum client implements the JSON-RPC specification, so there are a uniform set of methods that applications can rely on. If you want to use JavaScript to connect with an Ethereum node, it's possible to use vanilla JavaScript but several convenience libraries exist within the ecosystem that make this much easier. With these libraries, developers can write intuitive, one-line methods to initialize JSON RPC requests (under the hood) that interact with Ethereum.

¹⁰⁴¹ « Framework ou infrastructure logicielle : définition et traduction », 20 janvier 2019. <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203355-framework/>.

Un framework (ou infrastructure logicielle en français) désigne en programmation informatique un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application. C'est le framework, encore appelé structure logicielle, canevas ou socle d'applications en français, qui établit les fondations d'un logiciel ou son squelette applicatif. Tous les développeurs qui l'utilisent peuvent l'enrichir pour en améliorer l'utilisation.

¹⁰⁴² <https://react.dev/>

parties est l'explorateur d'une *blockchain*, qui est un outil permettant de la scanner afin de suivre les smart contracts qui y sont déployés : une personne peut ainsi fournir l'adresse de déploiement d'un *smart contract* au moteur de recherche de l'explorateur afin d'observer son comportement (vérifier le nombre de jetons qu'il tient en séquestre, les adresses qui ont récupéré les sommes, etc)¹⁰⁴³. En particulier, les explorateurs permettent de faire "vérifier un *smart contract*". Il s'agit d'un moyen de prouver qu'un code source donné correspond bien à celui d'un *smart contract* déployé dans la *blockchain*, dont on a fourni l'adresse. Cette fonctionnalité garantit aux personnes interagissant avec un *smart contract* que le code de celui-ci est bien celui prétendu par son développeur. Il s'agit d'une fonctionnalité fondamentale pour une partie qui n'est pas en charge du développement du contrat intelligent¹⁰⁴⁴, puisqu'elle lui assure qu'elle utilisera un programme dont elle connaît le code¹⁰⁴⁵.

Pour chaque *blockchain*, il existe un explorateur "officiel" dont pourront se servir les parties afin qu'elles puissent commodément suivre leur smart contracts et s'assurer que la version déployée correspond bien à celle qu'elles ont produite ou fait produire. Actuellement, l'explorateur *Etherscan* est le plus avancé en termes de fonctionnalités pour le L1 d'Ethereum. Tandis que pour *Arbitrum*, les parties pourront utiliser *Arbiscan*.

414. Les solutions d'indexation. Les solutions d'indexations permettent aux développeurs de *smart contract* de récolter des informations plus précises sur leur programmes, que celles fournies par des explorateurs¹⁰⁴⁶. Ces derniers peuvent être perçus comme des outils de scan généralistes, qui fournissent des informations sur un *smart contract* que jusqu'à un certain niveau de détail. Tandis que les solutions d'indexation peuvent être spécialisées dans l'analyse fine d'un *smart contract* donné

¹⁰⁴³ Gemini. « What Is a Block Explorer? BTC Block Explorers, Etc. » Consulté le 7 juillet 2023. <https://www.gemini.com/cryptopedia/what-is-a-block-explorer-btc-bch-eth-ltc>

A block explorer is an online tool that enables you to search for real-time and historical information about a blockchain, including data related to blocks, transactions, addresses, and more.

¹⁰⁴⁴ V., *infra*, §229

¹⁰⁴⁵ Etherscan.io. « Verify & Publish Contract Source Code | Etherscan ». Ethereum (ETH) Blockchain Explorer. Consulté le 7 juillet 2023. <http://etherscan.io/verifyContract>.

Source code verification provides transparency for users interacting with smart contracts. By uploading the source code, Etherscan will match the compiled code with that on the blockchain. Just like contracts, a "smart contract" should provide end users with more information on what they are "digitally signing" for and give users an opportunity to audit the code to independently verify that it actually does what it is supposed to do.

¹⁰⁴⁶ DEV Community. « Indexing Smart contract Data Using The Graph Protocol », 27 juillet 2022. <https://dev.to/jamiescript/indexing-smart-contract-data-using-the-graph-protocol-4h95>.

Smart contracts emit events that operate as a rich supply of data which can be aggregated. For instance, when tokens are transferred, an ERC20 token produces a transfer event. A developer or a user might index and aggregate this transfer event data and then run queries against it to learn more about the token's performance, such as the top holders, the volume of transactions, etc.

et la fourniture efficace aux parties d'informations très précises sur celui-ci.

Par exemple, si des parties souhaitent refléter rapidement dans leur *dapp* des informations qui viennent d'être renseignées dans leur smart contracts (comme l'irrespect d'un niveau de service...), elles pourront se servir d'une solution d'indexation qui mettra à disposition ces informations par le biais d'une API. La meilleure solution actuelle dans l'écosystème Ethereum est le protocole décentralisé *TheGraph*¹⁰⁴⁷, où des individus sont incités économiquement à récolter des informations demandées par d'autres en l'échange du paiement d'un jeton.

b) Les solutions de stockage décentralisées

415. Enregistrement des données. Enfin, les parties peuvent avoir besoin d'utiliser des solutions de stockage pour enregistrer et lier des fichiers aux programmes exécutant leurs contrats. Imaginons une cession de droits d'auteurs dans la *blockchain* : les droits sont représentés par un NFT et leur transfert dans la *blockchain* équivaut à une véritable cession de droits¹⁰⁴⁸. Le document décrivant les droits du cessionnaire doit pouvoir être accessible par les parties. Or, il est impossible de le stocker dans la *blockchain* ; les parties peuvent opportunément choisir de l'enregistrer dans une infrastructure du même type destinée au stockage.

En effet, même si il existe pléthore de solutions de stockage centralisées, celles décentralisées et au fonctionnement similaire à celui d'une *blockchain* ont les avantages d'être peu chères, résilientes et mieux connectés aux infrastructures d'exécution des smart contracts¹⁰⁴⁹. Ainsi, les parties seront avisées de stocker les fichiers liés à leur programmes sur des solutions comme *IPFS*¹⁰⁵⁰, *Arweave*¹⁰⁵¹ ou *Filecoin*¹⁰⁵².

Ces protocoles de stockage peuvent encore être combinées avec d'autres spécialisées dans le chiffrement. Imaginons une clause de séquestre de secrets industriels exécutée dans la *blockchain*. Si

¹⁰⁴⁷ <https://thegraph.com/>

¹⁰⁴⁸ Diallo, Abdoulaye. « TokenIP : Une Proposition de Tokenisation Des Droits d'auteurs En Droit Français ». Medium (blog), 30 décembre 2020. <https://abdoulaye77124.medium.com/tokenip-une-proposition-de-tokenisation-des-droits-dauteurs-en-droit-fran%C3%A7ais-43624b402ecc>.

¹⁰⁴⁹ Bispo, Nuno. « The Benefits of Decentralized Storage vs Cloud Storage ». Geek Culture (blog), 6 janvier 2022. <https://medium.com/geekculture/the-benefits-of-decentralized-storage-vs-cloud-storage-f5f01592ed9d>.

¹⁰⁵⁰ <https://ipfs.tech/>

¹⁰⁵¹ <https://www.arweave.org/>

¹⁰⁵² <https://filecoin.io/>

l'entreprise fait faillite, un *smart contract* autorise que ses secrets soient rendus accessibles à un tiers désigné. Les fichiers objets du contrat peuvent être stockés dans une solution de stockage décentralisée et connectés au *smart contract* autorisant leur délivrance, mais les secrets seront publics, visibles à la vue de tous. Les parties pourront alors utiliser une solution comme *Lit Protocol*¹⁰⁵³ ou *LightHouse*¹⁰⁵⁴ pour chiffrer ces documents stockés et prévoir qu'ils seront déchiffrés si le *smart contract* l'autorise. Nous recommandons aux parties de choisir une solution de stockage décentralisée avec de solides garanties de stockage pérenne : c'est-à-dire un nombre élevé de participants, un système économique incitatif et un écosystème mature.

Tableau récapitulatif du stack technique recommandé pour le développement de *smart contract*

Outils pour coder un <i>smart contract</i>	Outils pour le déploiement et l'interaction avec un <i>smart contract</i>	Outils pour la création d'une <i>Dapp</i>	Outils pour le suivi du <i>smart contract</i>	Outils pour le stockage et chiffrement de fichiers liés au <i>smart contract</i>
Langages de programmation : <i>Solidity</i> (avec recours extensif à la bibliothèque <i>OpenZeppelin</i>)	<i>Wallet</i> : <i>Smart wallet Argent</i> ou <i>Gnosis Safe</i>	Bibliothèque de développement front-end : <i>Ethers</i>	Explorateurs : <i>Etherscan</i> pour Ethereum ; <i>Arbiscan</i> pour Arbitrum	Stockage : <i>Filecoin</i> ou <i>Arweave</i>
Outils de développement : <i>Remix</i> ou <i>HardHat</i>	Service de fourniture de nœuds : <i>Infura</i> ou <i>Alchemy</i>	<i>Framework</i> de développement : <i>React</i> , <i>Next.JS</i>	Indexation : <i>theGraph</i>	Chiffrement : <i>Lighthouse</i> ou <i>Litprotocol</i>

§ II – Développement des programmes

416. Structure et éléments de code. Le stack technique décrit, nous pouvons porter notre attention sur les smart contracts. Avant de présenter des éléments de code intéressant l'exercice de formalisation des processus contractuels (B), il nous paraît nécessaire de présenter brièvement la structure générale du programme (A).

A - Structure d'un *smart contract*

417. Composition d'un *smart contract*. Un *smart contract* consiste, au stade de sa confection, en un fichier (à l'extension *.sol*) comportant des écritures dans le langage de

¹⁰⁵³ <https://litprotocol.com/>

¹⁰⁵⁴ <https://www.lighthouse.storage/>

programmation *Solidity*. Ce fichier a vocation à être compilé puis déployé dans l'EVM. Il est composé de données et de fonctions. Les données sont des informations pouvant être de tout type (chiffres, adresses cryptographiques, mots ordinaires...). Elles sont manipulées par des fonctions, qui sont des sortes de sous-programmes, effectuant n'importe quelle tâche leur étant assignée. Un *smart contract* consiste alors en une suite de fonctions où chacune d'elle effectue une opération qui vise soit à récupérer une donnée dans la *blockchain* ou à interagir avec elle et modifier son état.

418. Structure et fonctionnement du *smart contract*. Pour illustrer la structure et le fonctionnement typique d'un *smart contract*, nous présentons un exemple très simpliste d'un séquestrant des cryptomonnaies (des *ether*, la cryptomonnaie de Ethereum) et les délivrant à une personne en particulier. Il s'agit d'un transfert d'actif qui a pour seule condition que la clef publique (ou l'adresse) de la personne souhaitant les réceptionner soit celle-ci :

0x76703A497ea6c61285B43eCD89Ed97C87eD3bce1

```
pragma solidity ^0.8.7;

contract simpleExemple {

    address payable beneficiaire = 0x76703A497ea6c61285B43eCD89Ed97C87eD3bce1;

    uint montant;

    function deposer () public payable {

        //la fonction "deposer" reçoit de la crypto monnaie (ETH) et l'assigne à la
        variable "montant"

        montant = msg.value;

    }

    function retirer() public {

        //une fonction spéciale vérifie que l'adresse qui souhaite retirer la crypto-
```

```
monnaie séquestrée est bien celle du bénéficiaire...

require(msg.sender == beneficiaire);

//le montant lui est ensuite transféré

beneficiaire.transfer(montant);

}

}
```

419. Explications du *smart contract* simpliste. L'image présentée ci-dessus est le code source¹⁰⁵⁵ d'un *smart contract* tel qu'il apparaît dans le logiciel *Remix*¹⁰⁵⁶. Nous avons nommé notre *smart contract* (ou *contract* tel qu'il doit être écrit en *Solidity*) "*simpleExemple*" et il est constitué de toutes les instructions, qui s'exécutent les unes après les autres, se situant entre ses accolades « {} ». Tout en haut du fichier figure la version du langage *solidity* utilisée pour écrire le programme : *pragma solidity ^0.8.7*. Les inscriptions en vert écrites après une double accolade sont des commentaires, qui ont vocation à être lus par d'autres développeurs pour mieux comprendre le code et non être exécutées par la machine virtuelle.

Les premières lignes du *smart contract* sont consacrées à la définition et l'assignation des variables. Ces dernières peuvent être définies comme des données qu'on a nommées et auxquelles on a assigné une valeur. Pour l'exécution de notre *smart contract*, nous avons besoin d'enregistrer l'adresse du bénéficiaire qui pourra retirer les cryptomonnaies et leur montant. Il s'agit de deux données auxquelles nous donnons respectivement les noms de *bénéficiaire* et *montant* et des valeurs déterminées:

- la variable *bénéficiaire* contiendra l'adresse du bénéficiaire, avec l'attribut *payable* pour signifier qu'il s'agit d'une adresse pouvant envoyer et recevoir des *ether*;
- la variable *montant* est une donnée de type "chiffre entier" (*uint*) ; aucune valeur ne lui est encore assignée à sa création.

La première fonction *déposer* est celle qui permet de déposer les *ether*. Elle contient les attributs *public* et *payable* signifiant respectivement qu'elle est facilement lisible et capable de recevoir des

¹⁰⁵⁵ « Code source ». In Wikipédia, 29 novembre 2022.
https://fr.wikipedia.org/w/index.php?title=Code_source&oldid=199077345.

En informatique, le code source est un texte qui présente les instructions composant un programme sous une forme lisible, telles qu'elles ont été écrites dans un langage de programmation. Le code source se matérialise généralement sous la forme d'un ensemble de fichiers texte.

¹⁰⁵⁶ V., *infra*, §405

ether. Lorsqu'une personne active cette fonction, le nombre *d'ether* qu'elle envoie est automatiquement enregistrée dans une variable globale appelée *msg.value*. Cette dernière contient donc le montant *d'ether* envoyé et c'est sa valeur qui est assignée à la variable *montant*.

Pour retirer ces *ether*, il est créé une deuxième fonction *retirer*. Elle contient une mini-fonction (dont nous reviendrons sur le fonctionnement plus-bas) chargée de vérifier que l'adresse interagissant avec la fonction *retirer* est bien celle du bénéficiaire. Pour connaître l'adresse de la personne, ou du programme, interagissant avec une fonction, nous utilisons la variable globale *msg.sender*. La mini-fonction *require* vérifie alors que celle-ci correspond bien à celle enregistrée dans la variable *bénéficiaire*. Si c'est le cas, alors le montant sera transféré à ce dernier.

La structure typique d'un *smart contract* décrite, nous pouvons revenir plus en détail sur des éléments de code que peuvent mobiliser des parties dans leur développement de leur contrat intelligent.

B – Éléments de grammaire en *Solidity*

420. Éléments de grammaire *Solidity*. Nous présenterons ici plusieurs éléments de la grammaire du langage *Solidity*, afin d'expliquer comment des parties peuvent s'en servir pour exécuter des processus contractuels.

a) Les variables

421. Les variables d'adresses. Nous avons déjà abordé les variables d'adresses¹⁰⁵⁷. Comme déjà exposé, celles-ci servent à enregistrer des clefs publiques d'individus ou de *smart contract*. Les parties peuvent en avoir besoin pour s'identifier ou identifier des personnes dans leurs smart contracts, aux fins notamment de leur ouvrir l'usage de certaines fonctions ou encore de leur envoyer des jetons. Ci-dessous la manière de créer une variable d'adresse nommée « *partieA* » :

```
address partieA = 0x76703A497ea6c61285B43eCD89Ed97C87eD3bce1 ;
```

Il reste assez rare cependant que l'adresse soit assignée de la sorte dans un *smart contract*. Souvent, elle est créée (ou déclarée), puis se voit assigner plus tard une valeur. On peut ainsi prévoir qu'à l'occasion de l'interaction avec une fonction, l'adresse de la personne interagissant (enregistrée dans

¹⁰⁵⁷ V., *infra*, §418

la variable globale `msg.sender`) sera la valeur de la variable d'adresse précédemment créée :

```
address partieA;

function assignerAdressePartieA () public {

partieA = msg.sender ;

}
```

422. Les variables de nombres. Les parties peuvent évidemment avoir besoin d'utiliser des nombres dans leur smart contracts. Nous avons déjà vu que la variable à utiliser dans ce cas-là était `uint`, qui sert à assigner des nombres entiers¹⁰⁵⁸. Lorsqu'on connaît la taille du nombre qui sera la valeur d'une variable, il est possible de contrôler l'espace qu'elle prendra en spécifiant son nombre de *bits* (qui va de 8 à 256) ; et ainsi maîtriser le coût de déploiement du *smart contract*. Plus les *bits* seront nombreux, plus le nombre qu'on pourra assigner à la variable sera large¹⁰⁵⁹.

```
uint256 montant;

//il s'agit d'un montant pouvant accueillir des nombres allant de 0 jusqu'à
2^256-1

function assignerMontant () public {

montant = msg.value ;

}
```

423. Les variables de mots. S'agissant de textes, il est en général une mauvaise pratique de les enregistrer dans le code d'un *smart contract* et donc dans la *blockchain*. Ces infrastructures sont impropres à stocker ces types de données, et cause un coût de déploiement important¹⁰⁶⁰. Il est plus avisé de les conserver dans des solutions de stockage décentralisées, comme vu

¹⁰⁵⁸ V., *infra*, §419

¹⁰⁵⁹ « Solidity Data Types: Signed (Int) and Unsigned Integers (Uint) ». Consulté le 8 juillet 2023.
<https://www.alchemy.com/overviews/solidity-uint>.

An unsigned integer is a value data type declared with the uint keyword which stores a integer value equal to or greater than zero, ranging from 0 to 2to the 255th power - 1.(...) Similar to a signed integer, the keyword uint servers an abbreviation for uint256, an unsigned integer data value that can store up to a 256-bit integers or data units.

¹⁰⁶⁰ Nedelchev, Miroslav. « Answer to “What’s the proper way to store a long string (like a news article) in Solidity?” » Ethereum Stack Exchange, 24 septembre 2019. <https://ethereum.stackexchange.com/a/76195>.

Saving long strings could be very expensive operation in smart contracts. I see 2 solutions and they both require offchain actions: Save each article in IPFS and then save the returned hash into blockchain; Save each article in database and then save the record ID into blockchain.

précédemment¹⁰⁶¹. Mais des parties peuvent tout de même avoir besoin d'enregistrer de courts mots pour décrire certains éléments. Dans ce cas, elles disposent de la donnée *string* pour le faire.

Dans le développement de NFT (que nous aborderons plus-bas), il est courant de créer des variables de type *string* afin de leur assigner comme valeur un lien vers une page internet¹⁰⁶². Celle-ci renvoie elle-même vers une solution de stockage, généralement décentralisée, où réside un document contenant de plus amples informations textuelles sur l'actif non-fongible (qu'on appelle les métadonnées). Ainsi, seul le lien contenant ces informations est enregistré *on-chain*. Les parties souhaitant enregistrer des phrases ou des textes longs seront avisées de recourir à ce même type de procédé¹⁰⁶³.

Imaginons qu'un créancier souhaite représenter son contrat par un NFT :

```
//Lorsqu'on crée un NFT, on lui assigne un numéro qui fait son unicité. On peut
donc créer une variable « numero » de type chiffre entier

uint40 numeroNft = 1234;

//Les métadonnées du NFT seront enregistrées dans un site lui-même stocké sur
Filecoin. C'est le lien vers ce site qu'on enregistrera dans le smart contract, et
donc la blockchain.

string metadata = "www.lienimagenft.com";

//on donnera en paramètres à la fonction de création du NFT (c'est-à-dire entre
parenthèses), ces variables afin qu'elle puisse créer un NFT avec les données
renseignées.

function creerNftContrat (numeroNFT, metadata) public {
```

¹⁰⁶¹ V., *infra*, §415

¹⁰⁶² « ERC 721 - OpenZeppelin Docs ». Consulté le 8 juillet 2023.
<https://docs.openzeppelin.com/contracts/2.x/api/token/ERC721>.

Returns the URI for a given token ID. May return an empty string. If the token's URI is non-empty and a base URI was set (via `_setBaseURI`), it will be added to the token ID's URI as a prefix. Reverts if the token ID does not exist.

¹⁰⁶³ Patrick Collins. « How to Make an NFT and Render It on the OpenSea Marketplace ». freeCodeCamp.org, 1 avril 2021. <https://www.freecodecamp.org/news/how-to-make-an-nft-and-render-on-opensea-marketplace/>.

When smart contracts were being created, and NFTs were being created, people quickly realized that it's reaaaally expensive to deploy a lot of data to the blockchain. Images as small as one KB can easily cost over \$1M to store. This is clearly an issue for NFTs, since having creative art means you have to store this information somewhere. They also wanted a lightweight way to store attributes about an NFT – and this is where the tokenURI and metadata come into play. The tokenURI on an NFT is a unique identifier of what the token "looks" like. A URI could be an API call over HTTPS, an IPFS hash, or anything else unique.

```
//on ne le détaillera pas, mais ici la fonction en active d'autres pour créer le NFT et lui assigner des métadonnées.
```

```
. . .
```

```
}
```

424. Bytes32. Une autre technique souvent utilisée pour ne pas stocker de volumineuses données dans un *smart contract* est de recourir aux *hash* : au lieu d'écrire le texte directement dans une variable *string*, il est produit un *hash* de celui-ci, qui constitue son empreinte unique. Son stockage est bien mieux supporté dans une *blockchain*. En sus, cette démarche est potentiellement bien plus respectueuse de réglementations sur les données personnelles. En effet, le *hash* ne révèle rien du texte dont il provient, mise à part qu'il constitue un identifiant parfaitement unique à ce dernier¹⁰⁶⁴. Pour enregistrer un *hash*, il peut être recouru à la donnée dite de type *bytes32*. Imaginons qu'on doit enregistrer un élément identifiant d'une personne dans un *smart contract*, comme le nom d'un usager d'un service hospitalier.

```
//on crée une variable « nomUsager » destinée à contenir le hash du nom de l'usager
```

```
bytes32 nomUsager;
```

```
//la fonction suivante prend en paramètre le nom de l'usager qu'elle n'enregistre pas cependant dans le smart contract grâce à l'attribut "memory"
```

```
function nomToHash (string memory nom) public {
```

```
//on utilise une fonction globale appelée « keccak256 » qui prend en paramètre un mot et ressort son hash
```

```
//on assigne à la variable « nomUsager » la valeur correspondant au résultat de la fonction « keccak256 »
```

```
nomUsager = keccak256 (nom) ;
```

```
}
```

425. Bool. Les booléens sont des types de données qui peuvent également être très utiles dans le développement de *smart contract* pour des parties à un contrat intelligent. Ils consistent en des variables auxquelles peuvent être assignées une valeur : "vraie" ou "faux". Ceux-ci peuvent ainsi

¹⁰⁶⁴ « Hachage ». Consulté le 8 juillet 2023. <https://www.cnil.fr/fr/definition/hachage>.

L'utilisation d'une fonction de hachage permet de ne pas stocker les mots de passe en clair dans la base mais uniquement de stocker une empreinte de ces derniers. Il est important d'utiliser un algorithme public réputé fort afin de calculer les dites empreintes. A ce jour, MD5 ne fait plus partie des algorithmes réputés forts.

servir à renseigner si une chose est vraie ou fausse ; imaginons une simple clause de livraison déclenchant un transfert de jetons lorsque le bien est arrivé : un booléen *bienArrivé* pourra être créé. Il lui sera par défaut (c'est-à-dire dès sa création), assigné la valeur *false* qui pourra donc changer après l'activation d'une fonction.

```
//est créé une variable « bienArrive » qui a par défaut la valeur de « false »  
  
bool bienArrive;  
  
function leBienEstRecu () public {  
  
    //on imagine que cette fonction est activée par le livreur lorsqu'il fait  
    signer le PV de réception  
  
    bienArrive = true;  
  
}
```

b) Ordonnancement des données

426. Struct. Les structures sont des types de données servant à regrouper plusieurs autres données sous une seule et même variable. Nous pouvons les imaginer comme une valise pouvant contenir toutes sortes d'objets : la valise est la structure *struct* et les objets à l'intérieur sont d'autres variables. Elles peuvent donc constituer des types de données très utiles pour les organiser¹⁰⁶⁵.

Imaginons que des parties décident de coder le *smart contract* d'une société dont des processus seraient exécutés *on-chain*. A chaque associé est attaché un certain nombre d'informations : son nom, son adresse de résidence, son nombre de parts, etc. Les parties peuvent créer une structure *Associe* qui contiendrait des variables de texte, d'adresse et de nombres entiers renseignant des informations

¹⁰⁶⁵ jeeteshgavande30. « Solidity - Enums and Structs ». GeeksforGeeks (blog), 10 juillet 2020.
<https://www.geeksforgeeks.org/solidity-enums-and-structs/>.

Structs in Solidity allows you to create more complicated data types that have multiple properties. You can define your own type by creating a struct. They are useful for grouping together related data.

sur un associé. Elle prendrait la forme suivante :

```
struct Associe {  
    string nom;  
    address adresseAssocie;  
    uint nombreDeParts;  
}
```

On peut encore utiliser une structure pour représenter une créance et contenant des informations sur celle-ci : son montant, sa date d'échéance, les *wallet* du créancier et du débiteur et l'information pour savoir si elle est échue ou non.

```
struct Creance {  
    uint montant;  
    address creancier;  
    address debiteur;  
    bool echue;}
```

Pour accéder à la valeur d'une variable contenue dans une structure, on spécifie le nom ou la valeur de la structure, suivie d'un point et du nom de la variable dont on souhaite obtenir la valeur. Par exemple, pour avoir accès à la valeur du montant de la créance dans l'exemple ci-dessous :

```
Creance.montant
```

427. Les *mapping*. Similairement aux structures, les cartographies (*mapping* en anglais) sont des types de données permettant d'en organiser d'autres¹⁰⁶⁶. Celles-ci trient les données sous la forme de tableaux associatifs ayant une architecture clef-valeur comme ci-dessous :

Clef	Valeur

Ils peuvent donc servir à ordonner un nombre élevé d'informations, afin de mieux les classer et les retrouver. Imaginons le *smart contract* d'un contrat d'assurance paramétrique¹⁰⁶⁷. Dans celui-ci, l'assureur souhaite disposer d'un moyen de connaître l'état du versement des indemnisations qu'il

¹⁰⁶⁶ GeeksforGeeks. « Solidity - Mappings », 10 juillet 2020. <https://www.geeksforgeeks.org/solidity-mappings/>.

Mapping in Solidity acts like a hash table or dictionary in any other language. These are used to store the data in the form of key-value pairs, a key can be any of the built-in data types but reference types are not allowed while the value can be of any type. Mappings are mostly used to associate the unique Ethereum address with the associated value type.

¹⁰⁶⁷ V., *infra*, §111

doit aux assurés. Il peut utiliser un *mapping* qui contiendra en clef les adresses des indemnisés et en valeur un booléen indiquant si « vrai » ou « faux » une adresse a bien reçu son indemnisation.

```
//on crée un mapping en déterminant quelles seront les types de données qui
seront en clef et en valeur (puis en spécifiant son attribut et enfin son nom)
mapping(address => bool) public aEteIndemnisse;

//la fonction de versement de l'indemnisation prend en paramètre l'adresse
d'un assuré

function verserIndemnisation (address assure) public {

//en son sein, on utilise un sous-programme permettant de vérifier qu'une
condition est bien remplie

//en l'occurrence, on requête le mapping en indiquant sa clef entre ses
crochets : ici l'adresse de l'assuré

//si celle-ci renvoie « false », cela signifie que l'adresse de l'assuré
donnée en clef au tableau n'a pas reçu d'indemnisation. La fonction se
poursuit et lui verse la somme. Sinon elle s'arrête.

require (aEteIndemnisse[assure] == false);

. . .
}
```

Les parties seront avisées de faire un usage combiné des *struct* et *mapping*, afin de trier les informations de leur smart contracts et y accéder commodément. Si nous poursuivons notre exemple de société *on-chain*, on peut imaginer qu'il soit créé un *mapping* pour classer les informations sur les associés : à chaque adresse d'un d'entre eux correspondra un *struct*, qui lui-même contiendra des informations sur chaque associé comme son nombre d'actions et son nom. Autrement dit, un tableau associatif sera constitué et prendra en clef une adresse et en valeur une « valise » d'informations.

Clef	Valeur
Adresse de l'associé A	Informations sur l'associé A : {nombre d'actions de A, nom de A}
Adresse de l'associé B	Informations sur l'associé B : {nombre d'actions de B, nom de B}
Adresse de l'associé X	Informations sur l'associé X : {nombre d'actions de X, nom de X}

```
//un struct « Associe » est créé avec quelques variables

struct Associe {

string nom;

uint nombreActions;

}

//un mapping « associates » ayant pour clef une adresse (celle d'un associé) et
en valeur la struct Associe

mapping(address => Associe) public associates;
```

Lorsqu'une personne souhaite, par exemple, retirer des dividendes via une fonction *retirerDividendes*, il pourra être implémenté un mécanisme vérifiant que l'adresse interagissant avec cette fonction est bien celle d'un associé.

```
//pour vérifier que l'adresse activant la fonction « retirerDividendes » est
bien celle d'un associé, on cherche dans le mapping « associates » si l'adresse
du « msg.sender » a un nombre d'actions supérieure à 0.

function retirerDividendes () public {

//comme pour toute information contenue dans une structure, on accède à une
de ses variables en spécifiant sa valeur, un point, puis le nom de la variable
contenant la valeur qu'on cherche

//ici sa valeur est celle renvoyée par le mapping

require (associates[msg.sender].nombreActions > 0);

. . .

}
```

428. Enum. Les types de données *Enum* peuvent servir à suivre l'état d'un *smart contract*. Très souvent, ils sont utilisés pour servir de conditions à l'activation de fonctions. Des parties souhaitant que des fonctions soient actionnables qu'à la réalisation d'étapes précises peuvent utiliser un *enum* afin d'établir et suivre l'état du *smart contract*.¹⁰⁶⁸

Imaginons un contrat de construction formalisé en *smart contract*. Dans celui-ci, une fonction prévoit

¹⁰⁶⁸ jeeteshgavande30. « Solidity - Enums and Structs ». GeeksforGeeks (blog), 10 juillet 2020.
<https://www.geeksforgeeks.org/solidity-enums-and-structs/>.

Enums are the way of creating user-defined data types, it is usually used to provide names for integral constants which

que le maître d'œuvre pourra récupérer le prix de la prestation seulement lorsque le statut du *smart contract* sera sur “terminé” ; indiquant que la prestation est réalisée et que le prestataire peut récupérer son dû.

```
//on suppose que notre contrat aura 3 états : en exécution, suspendu et termine.

enum StatutContrat {

enExecution,

suspendu,

termine,

}

//l'enum est enregistré dans une variable qu'on nomme « statut »

StatutContrat public statut;

//pour activer la fonction « récupérerMontant », une fonctionnalité vérifie que le statut du contrat est sur « termine ». Si c'est le cas, alors le reste de la fonction s'exécute

function recupererMontant() public {

    require (statut == StatutContrat.termine);

    . . .

}
```

429. Event. Les types de données événements (*Event*) servent à communiquer des informations ayant eu lieu dans une *blockchain* afin qu'elles soient traitées par des programmes situés en dehors d'elle¹⁰⁶⁹. Ces programmes peuvent être des scripts de sites web, d'applications mobiles ou encore des solutions d'indexation discutées plus haut¹⁰⁷⁰. Son usage est donc recommandé pour des parties souhaitant faire déclencher des actions *off-chain* à partir d'évènements ayant eu lieu au sein

makes the contract better for maintenance and reading.

¹⁰⁶⁹ « Learn Solidity: What Are Events? » Consulté le 8 juillet 2023. <https://www.alchemy.com/overviews/solidity-events>.

In Solidity, events are dispatched signals that smart contracts can fire. When you call events, they cause the arguments to be stored in the transaction's log, which is a special data structure in the blockchain. Events notify external users, such as a listening frontend website or client application, that something has happened on the blockchain.

¹⁰⁷⁰ V., *infra*, §414

de leur smart contracts ; ou pour récupérer commodément la preuve d'opérations.

```
//on créé une variable de type évènement en indiquant son nom. Entre ses
parenthèses, on indique les types de données qu'elle devra émettre

event loyerPaye (uint40 montantLoyer);

//dans une fonction « payerLoyer », un locataire enverra un certain nombre d'ether
(d'où l'attribut « payable »).

function payerLoyer () public payable {

//on imagine que la fonction opère le transfert des ETH au bailleur

. . .
//une fois le transfert effectué, un évènement est émis indiquant que le loyer a
été payé et quel était son montant.

emit loyerPaye(msg.value);
}
```

c) Les structures conditionnelles

430. Les conditions. L'élément le plus évocateur des smart contracts dans l'imaginaire collectif est sans aucun doute la condition¹⁰⁷¹. Il s'agit d'une fonctionnalité qui suspend l'exécution de toute instruction à la satisfaction d'une opération. Il existe deux manières de poser une condition en *Solidity*.

431. Require. La première est de recourir à la fonction *require* que nous avons déjà plusieurs fois utilisé. Il s'agit d'une condition qui n'a que deux issues : soit l'opération posée est satisfaite et le reste du programme s'exécute, soit elle ne l'est pas et celui-ci affiche une erreur. Ce type de condition se retrouve généralement en tête de fonction. Dans notre contexte, elle sera souvent utilisée pour vérifier qu'une personne est autorisée à actionner une fonction.

Dans l'exemple ci-dessous, un oracle est chargé d'informer un *smart contract* qu'une marchandise a

¹⁰⁷¹ Arnaud LECOURT. « Chapitre 1 - Numérique et constitution de la société Section 3 - L'utilité de la blockchain ? §28 », Répertoire IP/IT et Communication Droit des sociétés et numérique, novembre 2020.

Un smart contract est un programme composé d'algorithmes reposant sur le principe « If ..., Then ... ». Il consiste à appliquer un résultat (then) lorsqu'il constate que les éléments nécessaires à cette application sont réunis (if). Il est un exécutant automatique : dès lors qu'il constate qu'une transaction a bien été validée par la blockchain (if), il va l'exécuter (then).

bien été envoyée. Il devra assigner au booléen *marchandiseEnvoyee*, la valeur *true*. La fonction permettant cette opération doit vérifier que l'adresse interagissant avec elle (*msg.sender*) est donc bien celle de l'oracle.

```
function informationMarchandise() public {  
  
    require(msg.sender == oracle);  
  
    bool marchandiseEnvoyee = true;  
  
}
```

432. Condition « If...else ». L'autre manière de créer des conditions est de recourir aux « if ...else ». Ces types de condition permettent de donner plus de contrôle sur le chemin que doit prendre le programme en cas de satisfaction ou non de l'opération posée. Il peut lui être indiqué quoi faire lorsqu'une condition est remplie, puis autre chose lorsqu'une autre condition est remplie, et même ce qu'il doit accomplir lorsqu'aucune des conditions est remplie. Cette structure permet donc de créer des chemins plus complexes que les *require*¹⁰⁷².

Dans notre contexte, les parties feront recours à ces deux types de condition en fonction de leurs besoins. Pour illustrer leur usage combiné, on peut imaginer la formalisation, *onchain*, d'une clause pénale stipulant qu'un transporteur devra s'acquitter d'une pénalité dépendant de son niveau de retard : un certain montant au-delà d'une heure, un montant supérieur au-delà de 2 heures, etc (nous partons du principe que le livreur a déjà versé au *smart contract* la somme totale qu'il est susceptible de perdre au titre de la clause pénale).

Une fonction *recupererIndemnite()* pourra être créée, permettant donc au client d'un service de livraison de réclamer une indemnité qui variera en fonction du retard du livreur.

```
function recupererIndemnite () public {  
  
    //on vérifie que le client est bien la personne interagissant avec la fonction  
    //puisque la condition est simple/binaire, un « require » suffit  
    require (msg.sender == client);  
  
    //si le client est bien le « msg.sender » alors la fonction se poursuit...  
    // dans le cas où une variable « tempsLivraison » indique une durée supérieure
```

¹⁰⁷² <https://ethereum.stackexchange.com/questions/60585/what-difference-between-if-and-require-in-solidity>

```

à une heure, les parties pourront prévoir quelque chose entre les crochets
if (tempsLivraison > 1 hours)
{
    . . .
}

// dans le cas où la variable « tempsLivraison » indique une durée supérieure
à deux heures  heure, les parties pourront prévoir autre chose entre les
crochets

else if (tempsLivraison > 2 hours)
    -   {
    -   . . .
    -   }

// et pour tous les autres cas. . .

else {
    -   . . .
}
}

```

433. Modifier. Un modificateur (*modifier*) est une fonction spéciale destinée à modifier le comportement d'une autre fonction à laquelle elle est greffée¹⁰⁷³. Comme toutes fonctions, elles peuvent contenir n'importe quelle instruction mais elles abritent souvent des conditions telles que des *require*. On utilise régulièrement les *modifier* comme des sortes de raccourcis afin d'appliquer, par exemple, la vérification d'une même condition à plusieurs fonctions.

Imaginons une clause compromissoire¹⁰⁷⁴ exécutée *on-chain*. Dans son *smart contract*, il existe plusieurs fonctions que seul l'arbitre a le droit d'actionner (comme le droit de rendre une sentence). Dans cette hypothèse, il peut être intéressant de coder un *modifier* contenant une condition (un *require*) prévoyant que l'adresse interagissant avec une fonction doit être celle de l'arbitre. Ainsi, toutes les fonctions auxquelles sera attaché ce *modifier* empêcheront quiconque que l'arbitre

¹⁰⁷³ freeCodeCamp.org. « What Are Solidity Modifiers? Explained with Examples », 6 janvier 2023.
<https://www.freecodecamp.org/news/what-are-solidity-modifiers/>.

A modifier is a special type of function that you use to modify the behavior of other functions. Modifiers allow you to add extra conditions or functionality to a function without having to rewrite the entire function.

¹⁰⁷⁴ Il s'agit d'une clause instituant un mécanisme d'arbitrage (de résolution de conflits par une personne privée au lieu d'un juge) dans un contrat.

d'interagir avec elles.

```
// on crée une fonction modifier « seulementLarbitre » comme une fonction
ordinaire, avec la particularité qu'on termine les instructions par "_" ;

//le « modifier » ici ne contiendra qu'un « require » prévoyant que le
« msg.sender » doit être celui de l'adresse enregistrée dans la variable
« Arbitre »

modifier seulementLarbitre {

    require(msg.sender == Arbitre);

    _; }

// le nom du « modifier » est ensuite placé juste après l'attribut public
d'une fonction

// ainsi, seul l'arbitre pourra utiliser la fonction « rendreSentence() » et
« recupererFraisArbitrage »

function rendreSentence() public seulementLarbitre {

    . . .

}

function recupererFraisArbitrage() public seulementLarbitre {

    . . .

}
```

d) Variables et fonctions spéciales

434. *Block.timestamp*. Il existe une variable spéciale dans *Solidity* appelée *block.timestamp* permettant de connaître le temps précis *t* au moment où elle est invoquée¹⁰⁷⁵. Plus précisément, cette variable contient le nombre de secondes écoulées depuis l'*unix epoch*¹⁰⁷⁶ jusqu'à

¹⁰⁷⁵ Morais, João Paulo. « Learn Solidity Lesson 19. Timestamp. » Coinmonks (blog), 9 août 2022. <https://medium.com/coinmonks/learn-solidity-lesson-19-timestamp-7ba91290c245>.

Let's start with the *block.timestamp* property, which returns the elapsed value, in seconds, between Unix time and the moment the block was mined (validated) on the network.

¹⁰⁷⁶ « What Is Unix Time? » Consulté le 8 juillet 2023. <https://kb.narrative.io/what-is-unix-time>.

l'instant t . *Solidity* fournit également des unités de temps comme les heures, les jours et les semaines, qui contiennent le temps, en secondes, écoulées dans une unité d'heure, de jour ou de semaine. Ces variables temporelles sont très utiles dans notre contexte car elles permettent à des parties de construire des conditions temporelles.

Imaginons un contrat de garantie autonome formalisée en *smart contract*¹⁰⁷⁷. Celui-ci permet à un bénéficiaire de récupérer, sur simple demande de sa part, le montant de la garantie. Dans notre exemple, la demande ne peut être faite que 30 jours après la création de la garantie ; ce qui correspondra au moment où la somme est déposée en séquestre dans le *smart contract*.

```
// on crée une variable « délai » de type chiffre entier

uint256 delai;

// on crée une fonction « commencerDelai » qui, dès lors qu'elle sera
// actionnée, aura pour effet d'assigner à « delai » la valeur, en secondes, du
// temps écoulé jusqu'à cet actionnement + le temps écoulé en secondes dans 30
// jours

function commencerDelai public () {

delai = block.timestamp + 30 days;

}

function depotDesFonds() public payable {

// dans la fonction servant à déposer la somme garantie, on actionne la
// fonction « commencerDelai »

commencerDelai();

...

}

function retirerFonds() public onlyBeneficiaire {

// au moment de retirer les fonds, il est vérifié que le bénéficiaire active
```

Unix time is a system for representing a point in time. It is the number of seconds that have elapsed since January 1st, 1970 00:00:00 UTC.

¹⁰⁷⁷ V., infra, §414

```

la fonction après l'écoulement du délai

//concrètement cela revient à regarder si le temps, en secondes, écoulé jusqu'à
l'activation de cette fonction est supérieur à celui enregistré dans la
variable délai1078

require(block.timestamp > delai);

...

}

```

435. **Constructor.** Un constructeur (*constructor*) est une fonction spéciale se déclenchant qu'au moment où est déployé un *smart contract* qui la contient¹⁰⁷⁹. Elle peut donc servir à établir ses paramètres initiaux avant que toute interaction avec lui soit possible. Des parties peuvent y recourir afin d'établir des paramètres de base de leurs programmes (comme des adresses, des valeurs ou autres).

Dans le morceau de code ci-dessous, on imagine le début d'un *smart contract* d'un séquestre logiciel¹⁰⁸⁰. Le programme fait intervenir le fournisseur du logiciel, le bénéficiaire du séquestre et un tiers-oracle. En utilisant la fonction *construct*, dès le déploiement, chacun des protagonistes se voit assigner des variables *adress* indiquées en paramètres de la fonction. Il ne sera ensuite plus possible

¹⁰⁷⁸ Dans une autre fonction, *retirerFonds*, on pourra alors vérifier que le bénéficiaire retire ses fonds après l'écoulement du délai en comparant le moment *t* auquel il fait sa demande et celui enregistré dans la variable *delai*. Si ce moment (mesuré encore une fois par *block.timestamp*) est inférieur à la variable *delai*, cela signifie qu'il fait sa demande trop tôt : la condition n'est pas remplie. À l'inverse, si *block.timestamp* est supérieur à *delai*, cela signifie qu'il fait sa demande après l'écoulement du délai ; la condition est remplie, il peut retirer les fonds.

¹⁰⁷⁹ jeeteshgavande30. « Solidity - Constructors ». GeeksforGeeks (blog), 10 juillet 2020.
<https://www.geeksforgeeks.org/solidity-constructors/>.

Solidity provides a constructor declaration inside the smart contract and it invokes only once when the contract is deployed and is used to initialize the contract state.

¹⁰⁸⁰ V., *infra*, §84

de les modifier.

```
address fournisseur;  
  
address beneficiaire;  
  
address tiers;  
  
constructor (address _fournisseur, address _beneficiaire, address _tiers) {  
  
    fournisseur = _fournisseur;  
  
    beneficiaire = _beneficiaire;  
  
    tiers = _tiers;  
}
```

e) Communication avec d'autres smart contracts

436. Importation. Il est recommandé, lors du développement d'un *smart contract*, de recourir à la méthode de l'importation : soit l'action d'inclure un morceau de code (stocké dans un autre fichier) dans son programme afin d'utiliser ses éléments sans avoir eu besoin de les produire soi-même et/ou de tous les faire figurer dans un seul fichier¹⁰⁸¹. Cela permet non seulement d'avoir une présentation plus limpide du *smart contract*, mais c'est également un gain de productivité et une bonne pratique en matière de sécurité.

Imaginons des membres d'une association souhaitant utiliser des smart contracts pour représenter plusieurs processus de leur organisation *onchain*. Le programme de ce contrat intelligent risque d'être complexe et verbeux. Au lieu de tout faire figurer dans un seul fichier, il peut être décidé de le modulariser et prévoir qu'un fichier contiendra le code relatif au vote des décisions, un autre celui de la gestion de la trésorerie, un autre celui de l'accueil de nouveaux membres, etc. En sus, les parties peuvent aussi choisir d'importer des morceaux de code d'organismes réputés (comme *OpenZeppelin*)

¹⁰⁸¹ Cvlr, Jean. « Solidity Tutorial: All About Imports ». Medium, 22 mars 2022.
<https://betterprogramming.pub/solidity-tutorial-all-about-imports-c65110e41f3a>.

The idea of importing files in Solidity is very similar to the concept of modules described above. It helps modularise your smart contracts by: Creating reusable pieces that other files can import. Making it easier to understand and digest the entire Solidity codebase of your project. Making it easier to work with the "Solidity modules" by focusing on smaller files (useful when debugging).

pour sécuriser leur smart contracts.

Le début du code du fichier principal ressemblerait à ceci :

```
pragma solidity 0.8.9;

// les importations se font en début de fichier

//ici on imagine que les fichiers sont dans un autre dossier que celui dans lequel on se trouve actuellement (un dossier parent)

import "./vote.sol";

import "./ajoutNouveauMembre.sol";

import "./tresorerie.sol";

//ici on importe (de OpenZeppelin) un mécanisme de sécurité contre une vulnérabilité connue des smart contracts

import
"https://github.com/OpenZeppelin/openzeppelincontracts/blob/master/contracts/security/ReentrancyGuard.sol";

//pour utiliser les fonctions et données issues des fichiers importés dans ce "contract", il faut les lui faire hériter en indiquant le nom de leur "contract" après le mot "is"

contract assosDAO is voteDAO, nouveauMembreDAO, tresorerieDAO {

. . .

}
```

Les parties peuvent ainsi utiliser les fonctions issues des smart contracts importés et même les réécrire. Ci-dessous un exemple d'une fonction qui a été importée pour être réécrite (grâce au mot "override"). A la dernière ligne de cette fonction, la fonction *transférerAdhesion* originale est ré invoquée, vierge de toute modification grâce au mot *super*.

```
function transfererAdhesion(address destinataire) public override {

. . .

super.transfererAdhesion(destinataire);

}
```

437. Interaction avec des smart contracts déjà déployés. Il est quasi-systématique qu'un

smart contract soit codé afin qu'il interagisse avec d'autres qui sont déjà déployés. Dans notre contexte, la plupart du temps les parties auront besoin de manipuler des jetons dans leurs programmes, qui sont donc eux-mêmes issus de *smart contract* déployés dans la *blockchain*.

Pour interagir avec un *smart contract*, les parties devront référencer dans le leur son *interface*¹⁰⁸², qui contient la description des fonctions du *smart contract* déployé avec lesquelles elles souhaitent interagir. L'*interface* pourra constituer alors une variable, qui se verra assigner une adresse correspondant à l'adresse de déploiement du *smart contract* dont elles référencent les fonctions.

Dans l'exemple ci-dessous, les parties utilisent un *smart contract* pour exécuter une clause de *earn-out*¹⁰⁸³. Elles souhaitent que le montant du complément de prix soit déposé dans un *smart contract*, puis transféré à nouveau dans le protocole de prêt décentralisé AAVE¹⁰⁸⁴ afin que la somme séquestrée génère des intérêts ; que les parties se partageront au moment du retrait. Les parties devront interagir avec le *smart contract* des jetons (représentant la monnaie déposée) et celui du protocole de prêt AAVE.

Elles vont donc :

- déclarer leurs interfaces ainsi que les fonctions de ces smart contracts qu'elles utiliseront,
- puis elles assigneront aux variables *interface* les adresses de déploiement de ces smart contracts,
- enfin elles pourront utiliser les fonctions des smart contracts déployés qui les intéressent dans leurs propres fonctions.

```
// on déclare l'interface du smart contract du jeton représentant la monnaie
interface EURO {

//seule la fonction de transfert de ce smart contract nous intéresse

function transferFrom(address from, address to, uint256 amount) external
```

¹⁰⁸² « What Is the Solidity Contract Interface? » Consulté le 8 juillet 2023.
<https://www.alchemy.com/overviews/solidity-interface>.

Solidity allows you to interact with other contracts without having their code by using their interface. For example, if you want to interact with another contract from your own contract, you provide your calls with an interface wrapper. By declaring an interface, you can interact with other contracts, and call functions in another contract.

¹⁰⁸³

¹⁰⁸⁴ <https://aave.com/>

```

returns (bool);

}

//on déclare l'interface du smart contract du protocole de prêt

interface PROTOCOLE {

//seule la fonction de dépôt dans ce protocole nous intéresse

function deposit(address _reserve, uint256 _amount, uint16 _referralCode)
external;

}

contract clauseEarnOut {

//on crée une variable "tokenEuro" qui représente le smart contract du jeton
EURO public TokenEuro = EURO(0xFF795577d9AC8bD7D90Ee22b6C1703490b6512FD);

//on crée une variable "aaveProto" qui représente le smart contract du
protocole

PROTOCOLE public aaveProto =
PROTOCOLE(0x580D4Fdc4BF8f9b5ae2fb9225D584fED4AD5375c);

. . .

//lorsque la fonction « déposerSommes » sera activée, elle actionnera la
fonction de transfert du jeton stable pour transférer les jetons puis celle
de dépôt du smart contract du protocole de prêt

function déposerSommes (uint256 _montant) external {

. . .

//on transfère la somme en jetons au présent smart contract...
tokenEuro.transferFrom(msg.sender, address(this), _montant);

//puis on dépose cette somme en jetons au protocole de prêt...
aaveProto.deposit(address(euro), _montant, 0);

}

}

```

Les jetons ERC20 et ERC721. Comme il vient d’être vu, les interactions les plus courantes que les

parties seront amenées à réaliser seront celles avec les smart contracts de jetons. Nous distinguons les jetons fongibles, des jetons non-fongibles. Dans l'écosystème Ethereum, les premiers respectent le standard ERC-20 et les seconds le standard ERC-721. Lorsqu'un *smart contract* suit un standard, cela signifie qu'il doit inclure un certain nombre de fonctions ayant un comportement précis¹⁰⁸⁵. Par exemple, les standards ERC-20 et ERC-721 imposent tous deux que les smart contracts les implémentant contiennent des fonctions de transfert (*transfer* et *transferFrom*), permettant d'envoyer un montant de jetons de l'adresse du détenteur vers celle du destinataire.

Imaginons un contrat de séquestre classique. Pour déposer les jetons-euros, il est fait recours à la fonction *transferFrom*.

```
function deposer (uint256 montant) public {  
    . . .  
  
    //la fonction prend en paramètres l'adresse de l'expéditeur(msg.sender), celle  
    du destinataire (le smart contract dans lequel sera séquestrée la somme, d'où  
    le « this »), et le montant envoyé (qui est une variable de type chiffre  
    entier  
  
    tokenEuro.transferFrom(msg.sender, address(this), montant);  
}
```

La fonction *transfer*, quant à elle, est utilisée afin de déclencher un transfert à partir d'un *smart contract*.

```
function retirer () public {  
  
    //cette fonction permet de transférer des jetons d'un smart contract vers une  
    personne ; c'est pour cette raison qu'elle ne prend en paramètres que le  
    destinataire (msg.sender) et le montant envoyé  
  
    tokenEuro.transfer(msg.sender, montant);  
}
```

S'agissant des jetons non-fongibles, ce ne sont pas des sommes qui sont transférées mais l'identifiant

¹⁰⁸⁵ Pratap, Zubin. « What Are Token Standards? A Complete List ». Chainlink Blog, 16 novembre 2022.
<https://blog.chain.link/token-standards/>.

Token standards are a set of agreed-upon rules that guide the design, development, behavior, and operation of cryptocurrency tokens on a given blockchain protocol. For token standards to be useful, they must be widely adopted. Without adoption, their rules cannot be elevated to the status of a “standard”—because standards are the rules that are generally followed by a wide range of people.

unique représentant l'actif.

```
function deposer (uint256 tokenId) public {  
  
    //au lieu d'indiquer le montant transféré, on fournit l'identifiant(  
    « tokenId ») du NFT  
  
    nft.transferFrom(msg.sender, address(this), tokenId);  
  
}  
  
function retirer () public {  
  
    nft.transfer(msg.sender, tokenId);  
  
}
```

438. Conclusion de la section I. Les parties pourront disposer de nombreux outils dans l'écosystème Ethereum pour créer et suivre les smart contracts exécutant leur contrat et stocker les données produites ou jointes à lui. Elles trouveront dans la grammaire du langage *solidity*, qui est le langage de programmation qu'elles seront avisées d'utiliser, plusieurs éléments qu'elles pourront mobiliser dans le cas d'usage spécifique de l'exécution de contrats *on-chain*.

Section II - Proposition de modèles de smart contract : *legalTemplate.sol*

***legalTemplate*.** Les prolégomènes établis, nous pouvons faire la présentation de nos modèles de *smart contract* spécialement codés pour exécuter des stipulations contractuelles *on-chain*¹⁰⁸⁶. Ceux-ci consistent en une suite de fichiers *solidity* (écrits en anglais afin de permettre la plus large utilisation possible) que des parties à un contrat intelligent pourront importer dans leur propres *smart contract* afin d'accélérer leur développement et/ou incorporer des fonctionnalités utiles et sécurisantes. Les modèles proposés consistent en des modules de base que nous pensons pouvoir être utilisés par des parties en toutes circonstances (§1), et des modules optionnels qu'elles pourront choisir d'ajouter ou non en fonction de leurs besoins (§2).

§ I - Modules de base

439. Modules essentiels. Nos modules de base mettent en œuvre trois fonctionnalités que nous estimons essentielles pour toute implémentation de contrat intelligent : la liaison du *smart*

¹⁰⁸⁶ Ces modèles de *smart contract* sont disponibles en *open source* sur lien *github* suivant : <https://github.com/ddyl24/legalTemplate>

contract au document *fiat* (A), la définition des rôles des intervenants dans le *smart contract* (B) et la possibilité de faire évoluer le *smart contract* (C).

A – Smart contract de liaison au contrat *fiat*

440. Implémentation de la propriété ricardienne. Notre premier module permet donc de lier le contrat *fiat* au *smart contract* l'exécutant. Nous avons vu, en effet, en quoi il est nécessaire que les smart contracts exécutant un contrat portent constamment en leur sein une référence de ce dernier¹⁰⁸⁷. La *blockchain* étant une infrastructure très impropre au stockage de données non succinctes, ce sera le *hash* de l'accord écrit en langage naturel, qui correspondra à l'empreinte unique de la version du document contractuel sur lequel les parties se seront accordées, qui sera enregistré dans le programme.

Nous ne proposons pas l'intégration de cette fonctionnalité dans le *constructor* car le *hash* enregistré doit être celui de la version complète du texte écrit en langage naturel. Or ce document n'est complet que s'il contient l'adresse de déploiement du *smart contract* censé l'exécuter¹⁰⁸⁸. Cette information n'est récupérable, par définition, qu'après le déploiement du *smart contract* ; à un moment donc où il sera impossible d'utiliser la fonction *constructor*. Le *smart contract* devra donc d'abord être déployé, puis dans un second temps, les parties pourront référencer le contrat écrit dans une fonction créée à cet effet. Le *hash* sera enregistré dans une variable public *agreementHash*, qui bénéficiera automatiquement d'un *getter*¹⁰⁸⁹ : une fonction permettant à quiconque d'avoir accès à sa valeur, et ainsi de connaître le *hash* de l'accord juridique écrit auquel se rattache un *smart contract*. Nous verrons plus tard comment faire en sorte que l'enregistrement du *hash*, qui peut être fait autant de fois que nécessaire (à chaque évolution du contrat écrit en langage naturel, un nouveau *hash* est produit et enregistré), soit ouvert qu'après avoir recueilli le consentement unanime des parties.

```
//il est créé une variable public "agreementHash" bytes32 pour stocker un hash
```

¹⁰⁸⁷ V., *infra*, §159

¹⁰⁸⁸ V., *infra*, §227

¹⁰⁸⁹ « Getter Functions for State Variables - Mastering Blockchain Programming with Solidity [Book] ». Consulté le 9 juillet 2023. <https://www.oreilly.com/library/view/mastering-blockchain-programming/9781839218262/048537c8-6e8f-4cef-860a-b923a96946e3.xhtml>.

If you have public state variables in your contract, the compiler will create getter functions for these automatically. Therefore, if you have already defined public state variables, you don't have to write getter functions explicitly for those variables. It isn't recommended to write getter functions for public state variables.

```

bytes32 public agreementHash;

//dans une fonction appelée « setAgreementHash », le hash soumis en paramètre
sera stocké dans la variable « agreementHash »

function setAgreementHash (bytes32 _agreementHash) public {

agreementHash = _agreementHash;

}

//dès qu'une personne souhaite avoir accès au hash de l'accord écrit en langage
naturel du smart contract, elle pourra librement, dans un explorateur, cliquer
sur la variable public « agreementHash »

```

B – Smart contract du contrôle d'accès des parties et des tiers au contrat

441. **Restriction à des rôles.** Nonobstant le *smart contract* implémenté par les parties, celles-ci voudront très certainement toujours restreindre l'accès de certaines de ses fonctionnalités à des personnes en particulier. Nous avons vu qu'elles pouvaient, pour cela, recourir à des structures conditionnelles comme *require(msg.sender == *adresse*)*. Mais si le *smart contract* fait intervenir un nombre assez important d'individus, elles pourront utiliser des outils plus appropriés.

442. **Multi-sig.** D'abord, les parties peuvent constituer un *smart wallet multi-sig*¹⁰⁹⁰ entre elles. Pour rappel, ces derniers sont des smart contracts mués en *wallet* qui peuvent notamment permettre de conditionner toute interaction avec une *blockchain* à l'approbation préalable d'un nombre *n* d'individus. Des parties peuvent donc décider d'être collectivement représentés par un tel *wallet* et renseigner son adresse comme la seule étant autorisée à pouvoir interagir avec certaines fonctions. La fonction *setAgreementHash* évoquée tantôt pourra être configurée afin de faire en sorte que son actionnement ne soit possible qu'après que toutes les parties aient donné leur consentement. On n'imagine mal, en effet, qu'une seule d'entre elles ait le droit unilatéral de changer la preuve de l'accord écrit en langage naturel sur lequel repose le *smart contract*. Avec un *wallet multi-sig*, une interaction réussie avec *setAgreementHash* devra donc nécessiter l'aval de chaque partie, quel que soit leur nombre.

443. **Contrôle d'accès.** De manière complémentaire au *multi-sig*, les parties seront avisées de recourir à la bibliothèque *AccessControl* de *OpenZeppelin* afin de mettre en œuvre un système

¹⁰⁹⁰ V., *infra*, §408

complexe de contrôle d'accès dans leur smart contracts¹⁰⁹¹. Il est possible, en effet, grâce à cette bibliothèque, de créer des rôles et les assigner à des adresses. Ainsi, au lieu de conditionner l'accès de certaines fonctions à des adresses, l'accès peut être soumis à la détention de rôles, qui peuvent facilement être assignés et révoqués à une ou plusieurs adresses. Par exemple, il pourrait être créé un rôle *LIVREUR*, qui serait assignée par les parties aux personnes chargées de livrer une marchandise dans leur contrat ; seules les adresses ayant ce rôle pourraient activer une fonction *bienReceptione*.

Dans notre proposition de module, nous intégrons donc cette bibliothèque et assignons un rôle d'administrateur au *multi-sig* constitué entre les parties : *DEFAULT_ADMIN_ROLE*. Ce dernier permet de créer et révoquer tous les autres rôles créés¹⁰⁹². Il est possible d'assigner une adresse à ce rôle dès le déploiement du *smart contract*. Comme le fait de créer, assigner et révoquer un rôle est un droit qui doit recueillir le consentement de toutes les parties, ce rôle d'administrateur par défaut doit être dévolu au *multi-sig* qu'auront constitué les parties entre elles.

Nous avons donc fait hériter notre *smart contract* de la bibliothèque *AccessControl*. Dès le *constructor*, nous recourons à la fonction *grantRole*¹⁰⁹³ (issue de la bibliothèque *AccessControl*, qui prend en paramètre le rôle qu'on souhaite assigner et l'adresse à laquelle on souhaite l'assigner) pour qu'elle attribue le rôle *DEFAULT_ADMIN_ROLE* à l'adresse avec laquelle est déployé le *smart contract*, qu'on suppose être le *multi-sig* constitué entre les parties.

```
pragma solidity ^0.8.2;

//la bibliothèque « AccessControl » est importée de github

import "https://github.com/OpenZeppelin/openzeppelin-
```

¹⁰⁹¹ « Access Control - OpenZeppelin Docs ». Consulté le 9 juillet 2023.
<https://docs.openzeppelin.com/contracts/2.x/access-control>.

Access control—that is, "who is allowed to do this thing"—is incredibly important in the world of smart contracts. The access control of your contract may govern who can mint tokens, vote on proposals, freeze transfers, and many other things. It is therefore critical to understand how you implement it, lest someone else steals your whole system.

¹⁰⁹² « Access Control - OpenZeppelin Docs ». Consulté le 9 juillet 2023.
<https://docs.openzeppelin.com/contracts/2.x/access-control>.

AccessControl includes a special role, called DEFAULT_ADMIN_ROLE, which acts as the default admin role for all roles. An account with this role will be able to manage any other role, unless _setRoleAdmin is used to select a new admin role.

¹⁰⁹³ « Access Control - OpenZeppelin Docs ». Consulté le 9 juillet 2023.
<https://docs.openzeppelin.com/contracts/2.x/access-control>.

Every role has an associated admin role, which grants permission to call the grantRole and revokeRole functions. A role can be granted or revoked by using these if the calling account has the corresponding admin role.

```
contracts/blob/master/contracts/access/AccessControl.sol";

//notre smart contract « legalTemplate » hérite de celle-ci avec le mot « is »
contract legalTemplate is AccessControl {

    constructor () {

        //dès le constructor (= dès le déploiement du smart contract), le rôle
        d'administrateur est assigné au « msg.sender », qu'on suppose être l'adresse
        du multisig des parties.

        _grantRole(DEFAULT_ADMIN_ROLE, msg.sender);

    }

    . . .}

```

Pour créer d'autres rôles que celui fourni par défaut, les parties utiliseront les variables *bytes32* qui stockeront des *hash* du nom de ces rôles. Ci-dessous la création des rôles ORACLE et ARBITRE.

```
bytes32 public constant ORACLE = keccak256("ORACLE");

bytes32 public constant ARBITRE = keccak256("ARBITRE");

```

Pour restreindre l'accès des fonctions à certains rôles, il est possible de faire un usage combiné de la fonction *require* et de la fonction *hasRole*. Cette dernière, issue de la bibliothèque *AccessControl* vérifie qu'une adresse a le rôle qu'on lui a soumis en paramètre (et renvoie *true* ou *false*)¹⁰⁹⁴. Intégrée dans un *require*, cela permet de la muer en condition à remplir pour qu'on puisse interagir avec une fonction. Si le rôle de PARTIES est donnée à l'adresse du *wallet multi-sig*, il est possible de restreindre l'accès de la fonction *setAgreementHash* de la manière suivante.

```
function setAgreementHash (bytes32 _agreementHash) public {

    //le « require » contient la fonction « hasRole » qui elle-même prend en
    paramètre « PARTIES » et l'adresse interagissant. Cela signifie que si
    l'adresse interagissant n'a pas le rôle de « PARTIES » qui lui est assignée,
    alors la fonction est bloquée

```

¹⁰⁹⁴ « Access Control - OpenZeppelin Docs ». Consulté le 9 juillet 2023.
<https://docs.openzeppelin.com/contracts/2.x/access-control>.

By default, accounts with a role cannot grant it or revoke it from other accounts: all having a role does is making the hasRole check pass.


```
require(hasRole(PARTIES, msg.sender));
. . .
}
```

Comme déjà évoqué, lorsque des conditions se répètent, il est pertinent d'utiliser des *modifier*¹⁰⁹⁵. A cet égard, la bibliothèque *AccessControl* fournit clef en main un *modifier onlyRole* qui permet, adossé à chaque fonction, de restreindre l'accès au rôle qui lui est donné en paramètre¹⁰⁹⁶. Ainsi pour toutes les fonctions devant être accessibles qu'aux parties, un *modifier onlyRole(PARTIES)* leur est greffé.

```
444. function setAgreementHash (bytes32 _agreementHash) public
onlyRole (PARTIES)
```

C – Smart contract d'interruption, de modification et d'extinction du contrat

445. Vie du smart contract. L'autre fonctionnalité essentielle fournie par nos modules de base est celle d'évolution du *smart contract*. Il est entendu par-là, la possibilité pour les parties de pauser (a), mettre à jour (b) et détruire le *smart contract* (c) exécutant leur contrat.

a) Pause

446. Paralysie temporaire du smart contract. Les parties doivent bénéficier d'un mécanisme permettant de suspendre l'exécution du *smart contract* lorsque la situation et/ou une clause de l'accord écrit en langage naturel l'impose. Imaginons qu'un contrat de location soit formalisée *on-chain*, dont sa clause de dépôt de garantie qui prévoit que le locataire sera autorisé à récupérer la somme, séquestrée par le *smart contract*, à la fin du bail. Si le locataire interrompt illégitimement son paiement, il est nécessaire que le *smart contract* pause la possibilité pour le locataire de recevoir automatiquement ce dépôt de garantie.

Dans notre proposition de module, ce mécanisme prendra la forme de deux fonctions issues de la bibliothèque *Pausable* de *OpenZeppelin*¹⁰⁹⁷. Celle-ci fournit donc deux fonctions, *pause* et *unpause*,

¹⁰⁹⁵ V., *infra*, §433

¹⁰⁹⁶ « Access Control - OpenZeppelin Docs ». Consulté le 9 juillet 2023.
<https://docs.openzeppelin.com/contracts/2.x/access-control>.

An account may have, for example, 'moderator', 'minter' or 'admin' roles, which you will then check for instead of simply using onlyOwner. This check can be enforced through the onlyRole modifier.

¹⁰⁹⁷ « Lifecycle - OpenZeppelin Docs ». Consulté le 9 juillet 2023.
<https://docs.openzeppelin.com/contracts/2.x/api/lifecycle>.

Contract module which allows children to implement an emergency stop mechanism that can be triggered by an

qui dès lors qu'elles sont actionnées, modifient un booléen `_pause` qu'elles mettent sur `true` ou `false`. Des `modifier whenPaused` et `whenUnpaused` peuvent être utilisés ensuite pour altérer le comportement des fonctions auxquelles ils sont apposés en fonction de l'état du booléen `_pause`. Autrement dit, dès lors que la fonction `pause` a été actionnée, et que le booléen `_pause` est sur `true`, toutes les fonctions prévues pour fonctionner quand le *smart contract* n'est pas pausé (donc contenant le `modifier whenUnpaused`) ne pourront pas être actionnées. Inversement, des parties pourront prévoir des fonctions spéciales qui ne pourront être accessibles que lorsque la fonction `pause` aura été activée (les fonctions avec le `modifier whenPaused`).¹⁰⁹⁸

Dans notre modèle, il faudra que les parties spécifient qui aura la faculté de pauser le *smart contract* en créant, par exemple, un rôle spécifique PAUSER à l'aide la librairie *AccessControl*. A travers leur *multisig* (qui est administrateur et a donc le droit de nommer et révoquer n'importe quel rôle), elles pourront donner ce rôle à chacune d'entre elles et éventuellement à d'autres personnes comme un arbitre ou un médiateur.

```
//notre smart contract hérite de la bibliothèque « Pausable »

contract legalTemplate is Pausable, AccessControl {

    ...

    //il est créé un rôle de pauser

    bytes32 public constant PAUSER = keccak256("PAUSER");

    //on imagine que le rôle de « PAUSER » est assigné à une adresse (celle du
    multisig, d'un arbitre, etc)

    ...

}

//les personnes pouvant pauser ou « dépauser » le smart contract sont celles
```

authorized account. This module is used through inheritance. It will make available the modifiers `whenNotPaused` and `whenPaused`, which can be applied to the functions of your contract. Note that they will not be pausable by simply including this module, only once the modifiers are put in place.

¹⁰⁹⁸ « Lifecycle - OpenZeppelin Docs ». Consulté le 9 juillet 2023.

<https://docs.openzeppelin.com/contracts/2.x/api/lifecycle>.

`whenNotPaused()` Modifier to make a function callable only when the contract is not paused. `whenPaused()` Modifier to make a function callable only when the contract is paused.

```

ayant le rôle de « PAUSER »

function pause() public onlyRole(PAUSER) {

    _pause();

}

function unpause() public onlyRole(PAUSER) {

    _unpause();

}

```

On peut imaginer une fonction *recupererRemuneration* contenant un *modifier whenNotPaused*. Cela signifie que tant que la fonction *pause* n'est pas activée, un salarié pourra récupérer sa rémunération. Mais dès lors qu'elle l'est, le salarié ne pourra plus actionner *verserRemuneration*.

```

function verserRemuneration () public whenNotPaused {

    . . .

}

```

b) – Mise à jour du *smart contract*

447. Mise à jour par proxy. Les parties doivent ensuite disposer de la possibilité de mettre à jour leur *smart contracts*. Il s'agit d'une autre fonctionnalité se trouvant au cœur de notre proposition de méthodologie : le programme étant inféodé à l'écrit, il doit pouvoir être mis à jour, si nécessaire, pour refléter la volonté réelle (matérialisée par un texte en langage naturel) des parties. Or, contrairement à une croyance tenace¹⁰⁹⁹, il est possible de mettre en place un système d'évolution du *smart contract* dans l'environnement immuable et décentralisé que constitue pourtant une *blockchain*.

Pour ce faire, dans notre proposition de modèle, nous recourons de nouveau à une autre librairie *OpenZeppelin* appelée *UUPSUpgradeable*¹¹⁰⁰. Elle met en œuvre une méthode de mise à jour par

¹⁰⁹⁹ Fabien Gillioz. « Du contrat intelligent au contrat juridique intelligent », Dalloz IP/IT, 2019, 16.

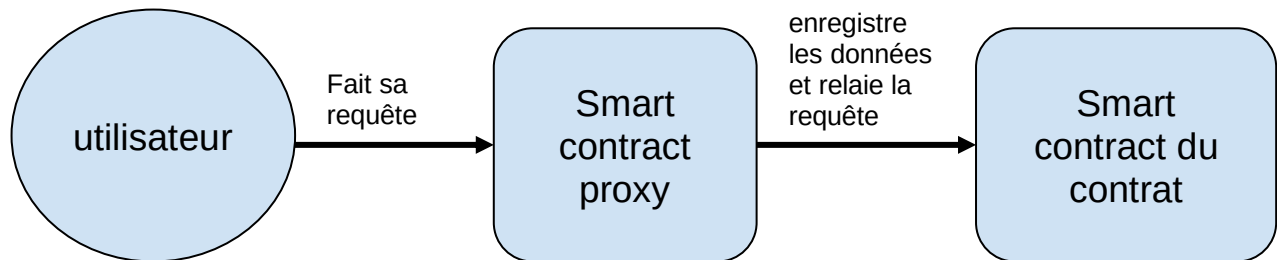
Étant donné que les smart contracts sont auto-exécutoires et immuables, une modification du contenu de la transaction n'est pas possible.

¹¹⁰⁰ « Proxies - OpenZeppelin Docs ». Consulté le 9 juillet 2023. <https://docs.openzeppelin.com/contracts/4.x/api/proxy>.

UUPSUpgradeable (...). An upgradeability mechanism designed for UUPS proxies. The functions included here can

proxy¹¹⁰¹. Celle-ci consiste à déployer un *smart contract* “mandataire” ou “intermédiaire” en même temps que le *smart contract* avec lequel il est souhaité interagir. Ce *smart contract* intermédiaire devient alors le programme par lequel passe les utilisateurs pour interagir avec le *smart contract* du contrat intelligent dans notre contexte.

Schéma explicatif de la mise à jour par proxy (1)



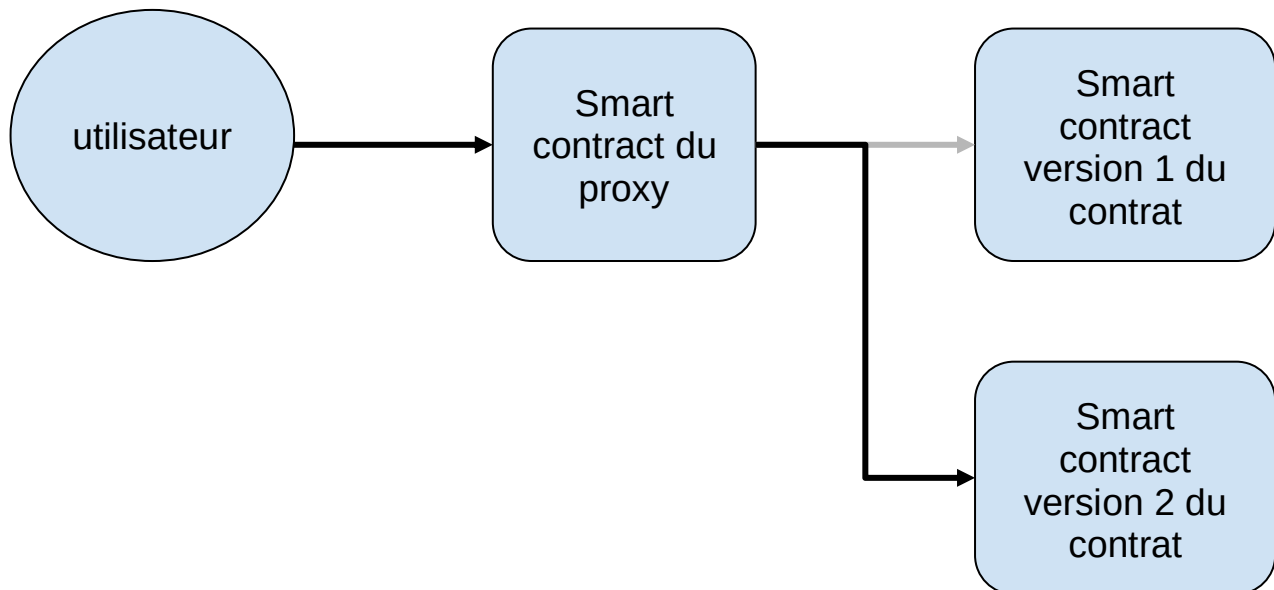
Dès lors qu’il est souhaité faire une mise à jour, un nouveau *smart contract* est créé et le programme intermédiaire pointe vers son adresse au lieu de la version précédente. Le proxy enregistre les données utilisées dans l’ancienne version, cela signifie que celles-ci pourront être réutilisées si besoin dans la nouvelle version de *smart contract* déployé.

perform an upgrade of an ERC1967Proxy, when this contract is set as the implementation behind such a proxy.

¹¹⁰¹ S, Pranesh A. « Using the UUPS Proxy Pattern to Upgrade Smart contracts ». LogRocket Blog (blog), 24 février 2022. <http://blog.logrocket.com/using-uups-proxy-pattern-upgrade-smart-contracts/>.

We all know that one of the most impressive features of the blockchain is its immutability property. But it is not advantageous in all cases. Imagine a deployed smart contract that holds user funds having a vulnerability(...). Traditional smart contract patterns don’t allow such hot fixes. Instead, the developers need to deploy a new contract every time they want to add a feature or fix a bug (...). To solve this problem, various upgradability patterns have been introduced. Among them, the proxy pattern is considered the truest form of upgradability. When we speak about upgradability, it means that the client always interacts with the same contract (proxy), but the underlying logic can be changed (upgraded) whenever needed without losing any previous data.

Schéma explicatif de la mise à jour par proxy (2)



La bibliothèque *UUPSUpgradeable* crée quelques bouleversements avec les bibliothèques précédemment évoquées, de sorte qu'il soit nécessaire d'utiliser des bibliothèques seulement compatibles avec elle. Fort heureusement, *OpenZeppelin* fournit des doubles conciliables des bibliothèques *Pausable* et *AccessControl* avec celle de mise à jour. Le changement principal occasionné par *UUPSUpgradeable* est la paralysie de la fonction *constructor*, remplacée par *initialize* ayant le même rôle et fonctionnement. Cela signifie que notre modèle utilise cette dernière fonction en tant que *constructor* et instancie les librairies susmentionnées de la sorte :

```
pragma solidity ^0.8.4;

//le smart contract importe des bibliothèques « OpenZeppelin » compatibles
avec la fonctionnalité de mise à jour (« Upgradeable »)

import"@openzeppelin/contracts-
upgradeable@4.7.3/security/PausableUpgradeable.sol";

import"@openzeppelin/contracts-
upgradeable@4.7.3/access/AccessControlUpgradeable.sol";

import"@openzeppelin/contracts-
upgradeable@4.7.3/proxy/utils/Initializable.sol";

import"@openzeppelin/contracts-
```

```

upgradeable@4.7.3/proxy/Utils/UUPSUpgradeable.sol";

contract legalTemplate is Initializable, PausableUpgradeable, UUPSUpgradeable,
AccessControlUpgradeable {
    . . .

    //la fonction « initialize » (avec son modifier « initializer ») remplace la
fonction constructor

    function initialize () initializer public {
        . . .

        _grantRole(DEFAULT_ADMIN_ROLE, msg.sender);
    }
}

```

448. Mise en œuvre de la mise à jour. Lorsque les parties souhaiteront mettre à jour leur smart contracts, elles feront recours à la fonction *upgradeTo* issue de la bibliothèque *UUPSUpgradeable*. Celle-ci prend en paramètre une adresse qui sera donc celle de la nouvelle version du *smart contract* que les parties souhaitent implémenter. Il est possible de contrôler l'accès de cette fonction avec un rôle, comme évoqué précédemment. Dans notre cas, nous avons ajouté un *modifier onlyRole* à la fonction interne *authorizeUpdate* (qui est systématiquement appelée lorsqu'est actionnée *upgradeTo*). Autrement dit, le *modifier onlyRole(PARTIES)* fera en sorte que seul le *wallet multi-sig* constitué entre les parties pourra être en mesure de mettre à jour le *smart contract*. La fonction de mise à jour contiendra aussi un *modifier whenPaused*, pour qu'elle puisse être réalisée seulement lorsque le *smart contract* aura été préalablement mis en pause.

```

function _authorizeUpgrade(address newImplementation)

internal

whenPaused

onlyRole(PARTIES)

override

{}

```

c) Destruction du *smart contract*

449. Mort du *smart contract*. Additionnellement aux mécanismes de pause et mise à jour,

notre modèle contient enfin un mécanisme de destruction du *smart contract*. Celui-ci sert à terminer le programme lorsque le contrat qu'il exécute a pris fin (résiliation, arrivée du terme ou de la condition résolutoire, dissolution de l'organisation, nullité, etc). Pour parvenir à cette fonctionnalité, nous recourons à la fonction *selfdestruct* qui supprime les *bytecodes* d'un *smart contract*, le rendant inapte à toute interaction¹¹⁰².

La fonction *selfdestruct* prend en paramètre une adresse censée recevoir les *ether* du *smart contract* supprimé. Tout comme la fonction *upgradeTo*, nous estimons que cette fonction est suffisamment importante pour ne pouvoir être invoquée que par l'ensemble des parties et après seulement que le *smart contract* ait été pausé. La fonction embarquera donc les *modifier whenPaused* et *onlyRole(PARTIES)*.

```
function terminateContract (address payable _recipient) public whenPaused
onlyRole(PARTIES) virtual {

selfdestruct(_recipient);

}
```

Il pourrait être argué que la faculté de détruire un *smart contract* puisse être accordée à chacune des parties : une partie dispose du droit unilatéral de résilier un contrat. Pour cette raison, la fonction *terminateContract* proposée dans notre modèle porte l'attribut *virtual*, cela signifie qu'elle peut être réécrite lors de son héritage. Les parties pourront ainsi prévoir que chacune d'entre elle aura droit de terminer le *smart contract* et spécifier quelles seront les conséquences dans la fonction (transfert de tous les jetons stables à un *wallet multi-sig* ou un arbitre, etc).

§ II- Modules optionnels

450. Modules éventuels. En sus des fonctionnalités de base présentées, nous proposons des modules optionnels que les parties peuvent choisir d'intégrer ou non en fonction de leurs besoins. Seront proposés des modules permettant de transférer des rôles (A), recourir à une solution d'oracle

¹¹⁰² « Learn Solidity: What Is Selfdestruct? » Consulté le 9 juillet 2023.
<https://www.alchemy.com/overviews/selfdestruct-solidity>.

Selfdestruct is a keyword that is used to terminate a contract, remove the bytecode from the Ethereum blockchain, and send any contract funds to a specified address.

décentralisée (B) et utiliser une solution de règlement des litiges *on-chain* (C).

A – *Smart contract* de cession du contrat

a) NFT

451. Rôle-NFT. Nous avons vu qu'à l'aide de la bibliothèque *AccessControl* il était possible de restreindre l'accès de certaines fonctions à la détention de rôles assignés à des adresses¹¹⁰³. Des parties peuvent créer un rôle pour chacune d'entre elles (PARTIE_A, PARTIE_B...) en plus de ceux des tiers (ARBITRE, LIVREUR, etc) et prévoir que seuls ces rôles auront la possibilité d'interagir avec certaines fonctions (pauser/suspendre/détruire le *smart contract*, renseigner une information, etc). Il est courant dans un contrat ordinaire qu'une partie prenante puisse céder sa qualité à une autre : dans un contrat de location par exemple, le bailleur comme le locataire peuvent être autorisés à céder leur positions à d'autres en cédant leur contrat. Nous proposons un moyen de reproduire ce mécanisme dans notre module en représentant les rôles de *AccessControl* par des NFT.

De cette sorte, à chaque rôle créé, correspondra un NFT qui pourra donc être cédé en tant que tel. Ce faisant, la qualité d'une partie deviendra un actif bénéficiant de la liquidité des jetons non fongibles. Imaginons un *smart contract* exécutant une simple créance : un débiteur dépose dans un smart contract des jetons-euros que seul son créancier est autorisé à retirer après un certain temps. Si ce *smart contract* hérite du module que nous allons présenter, il pourra exploiter la bibliothèque *AccessControl* afin qu'il soit défini que seule une adresse ayant le rôle de CREANCIER pourra retirer les fonds séquestrés. Mais dès lors que ce rôle sera créé, un NFT associé sera également généré, que son adresse détentrice pourra céder à qui elle le souhaite dans la *blockchain*. Il lui sera alors factuellement possible de réaliser une cession de créance *on-chain*.¹¹⁰⁴

Évidemment la possibilité de réaliser cette opération n'exonère pas du respect des dispositions du code civil en matière de cession de contrat (ou de créance). Pour poursuivre notre exemple, le créancier ne pourra céder son « NFT-Créance » qu'après avoir fait constaté la cession par écrit¹¹⁰⁵. Il

¹¹⁰³ V., *infra*, §441

¹¹⁰⁴ Ce qui est un cas d'usage qui intéresse de plus en plus.

Frederik Bussler. « Why NFTs Are The Future of Invoicing | HackerNoon », 22 août 2022. <https://hackernoon.com/why-nfts-are-the-future-of-invoicing>.

Tokenized receivables offer a more efficient and transparent way to factor invoices. Because they're stored on a blockchain, businesses can quickly and easily sell their receivables to the highest bidder. This means businesses can get the cash they need without having to worry about being taken advantage of by middlemen.

¹¹⁰⁵ Article 1216 alinéa 3 du code civil : La cession doit être constatée par écrit, à peine de nullité.

pourra s'aider, pour cela, d'une solution de signature électronique et requérir que la cession du NFT ne soit possible qu'après que le cessionnaire ait préalablement signé un document de cession de contrat¹¹⁰⁶.

452. Explications du *smart contract* de cession des rôles. Puisque notre proposition consiste à créer des NFT, il sera nécessaire que les parties recourant à notre module déploient préalablement notre *smart contract* avant le leur, et le référence dans ce dernier via une interface¹¹⁰⁷. Le fonctionnement de notre module est donc le suivant :

- à la création d'un rôle, un NFT associé lui est généré. Nous avons donc créé un *mapping* prenant comme clef une adresse et en valeur un *struct* "Person" contenant une variable nombre "*id*" et une autre *bytes32* "*role*". Comme son nom l'indique, cette structure représente une personne intervenante dans le *smart contract* et contient deux informations : l'une sur son rôle et l'autre sur l'identifiant du NFT de ce rôle.

```
struct Person{
uint40 id;
bytes32 role;
}
mapping(address => Person) public persons;
```

- nous réécrivons ensuite la fonction *grantRole* de la bibliothèque *AccessControl*, dont notre programme hérite, afin que lorsqu'elle soit actionnée, elle génère un NFT lié au rôle qu'elle assigne. Avant cela, les informations sur le rôle créé et l'identifiant du NFT seront enregistrés

¹¹⁰⁶ Très simplement elles pourront prévoir *off-chain* une page où un cessionnaire sera obligé de signer un document pré rempli de cession de créances. Cette signature déclenchera une activation dans le *smart contract* du NFT l'autorisant à pouvoir être cédé à l'adresse du signataire.

¹¹⁰⁷ V., *infra*, §437

dans le mapping *persons*.

```
function _grantRole(bytes32 role, address account) internal override {  
    // vérifie que le rôle n'a pas déjà été assigné  
    require (!hasRole(role, account), "No Duplicate role");  
    //créé une structure « person » et l'enregistre dans le «mapping » persons  
    Person storage person = persons[account];  
    //enregistre le rôle dans la structure venant d'être créée  
    person.role = role;  
    //enregistre l'id du NFT dans la structure venant d'être créée  
    person.id = _tokenId;  
    // génère le nft associé au rôle  
    _mint(account, _tokenId);  
    //incrémente le compteur des identifiants  
    _tokenId++;  
    //donne le rôle spécifié à l'adresse spécifié  
    super._grantRole(role, account);  
}
```

La fonction *revokeRole* pourra se servir des informations stockées dans le *mapping persons* pour supprimer le NFT lié au rôle qu'on souhaite révoquer.

```
function _revokeRole(bytes32 role, address account) internal override {  
    //supprime le nft associé au rôle en récupérant l'information sur son id grâce  
    à « persons » et en le donnant en paramètre à la fonction de suppression d'un  
    NFT : « _burn » du standard ERC721.  
    _burn(persons[account].id);  
    super._revokeRole(role, account);  
}
```

Dans le schéma que nous proposons, le cédant d'un NFT transfère au même moment son rôle au cessionnaire. Afin de mettre en œuvre cette opération, nous avons réécrit la fonction *beforeTokenTransfer* du standard ERC-721. Celle-ci est, comme son nom l'indique, actionnée juste

avant qu'un jeton soit transféré¹¹⁰⁸. La réécriture consistera donc simplement à révoquer le rôle du cédant du NFT pour donner le même au cessionnaire. On se servira, à nouveau, des informations stockées dans le *mapping persons*.

```
function _beforeTokenTransfer(
    address from,
    address to,
    uint256 tokenId,
    uint256 batchSize
) internal virtual override {
    //on identifie le cédant dans notre mapping « persons »
    Person memory sender = persons[from];
    // on révoque son rôle (ce qui supprime le NFT associé)
    revokeRole(sender.role, from);
    // on donne ce même rôle au cessionnaire (ce qui génère un NFT associé)
    grantRole(sender.role, to);
    //le comportement par défaut de la fonction se poursuit
    super._beforeTokenTransfer(from, to, tokenId, batchSize);
}
```

Pour résumer, le module permet que :

- à chaque création d'un rôle, un NFT associé est créé et les informations sur ce rôle et ce NFT soient enregistrés dans un tableau ;
- à chaque suppression d'un rôle, le NFT associé à ce rôle est détruit (grâce aux informations fournies dans le tableau) ;
- à chaque transfert d'un NFT, le rôle qui lui est associé est révoqué au cédant et assigné au cessionnaire (grâce également aux informations fournies dans le tableau).

¹¹⁰⁸ « ERC 721 - OpenZeppelin Docs ». Consulté le 8 juillet 2023.
<https://docs.openzeppelin.com/contracts/2.x/api/token/ERC721>.

_beforeTokenTransfer(address from, address to, uint256 tokenId) internal | Hook that is called before any token transfer. This includes minting and burning.

b) Gouvernance

453. NFT-Vote. Les NFT confèrent d'autres avantages qu'une liquidité accrue aux actifs dont ils sont le support. Ils peuvent également servir pour des opérations de gouvernance. Autrement dit, il peut être prévu qu'un NFT serve à représenter une voix dans des smart contracts grâce à laquelle il est possible de voter des décisions qui déclenchent l'activation de certaines fonctions. Imaginons que soit exécuté un contrat de prêt syndiqué¹¹⁰⁹ *onchain*. Les prêteurs disposent tous de rôle-NFT « PRETEUR ». Ils ont chacun la possibilité de céder leur contrat à d'autres, ce que permet facilement notre module avec la représentation des rôles en NFT.

A certaines étapes du contrat de prêt syndiqué, les prêteurs peuvent également avoir besoin de prendre des décisions collectives. Ce processus peut opportunément être formalisé *onchain* : tous les détenteurs d'un NFT-rôle PRETEUR pourront voter afin, par exemple, de déclencher les fonctions mettant à disposition la somme désirée à l'emprunteur. Il importera peu que les prêteurs puissent changer d'identité pendant l'exécution du contrat puisque leur qualité est attachée à un NFT, qui est la seule condition nécessaire pour voter.

Dans notre module, nous avons donc importé la bibliothèque *Governor* de *OpenZeppelin*¹¹¹⁰ afin que toute adresse détenant un rôle puisse donner son aval (à la manière d'un *multisig*) ou voter, pour interagir avec les fonctions désirées par les parties. Elle pourra servir à actionner certaines fonctions nécessitant le consentement (unanime ou majoritaire) de plusieurs rôles comme *setAgreementHash* ou *upgradeTo* (précédemment étudiés). Ce *smart contract* devra être déployé après celui des NFT¹¹¹¹, et avant celui des parties. L'usage du module fourni par *OpenZeppelin* nécessitera cependant quelques adaptations. Les parties créant les rôles doivent avoir la possibilité de déterminer si ceux-ci donnent droit à participer à la gouvernance. Tout rôle créé n'a pas forcément vocation à avoir le droit de proposer au vote et/ou voter des décisions (le tiers ou l'arbitre n'ont pas vocation, par exemple, à participer à la gouvernance).

Aussi, dans la structure représentant une personne (*person*), nous avons ajouté une nouvelle variable de type *bool* indiquant si cette dernière est autorisée ou non à voter dans la *DAO*. Par défaut, la valeur

¹¹⁰⁹ Banque de France. « Fiche 412 - Référentiel des financements des entreprises - Les prêts bancaires classiques » p.5, 15 novembre 2016.

Le crédit syndiqué est un crédit fourni par une association de plusieurs établissements financiers, réunis dans un syndicat bancaire, pour financer un projet donné ou une entreprise donnée.

¹¹¹⁰ « Governance - OpenZeppelin Docs ». Consulté le 10 juillet 2023.
<https://docs.openzeppelin.com/contracts/4.x/api/governance>.

¹¹¹¹ V., *infra*, §451

de cette variable (nommée *canGovern*) est sur *false*. Cela signifie que donner un rôle ou le transférer ne donne pas automatiquement droit à la possibilité de gouverner. Pour cela, nous avons créé une fonction spéciale (différente de *grantRole*) appelée *grantGovernRole* qui donnera la valeur *true* au booléen *canGovern* de la structure *Person*. La fonction *revokeRole* est aussi modifiée pour que son appel fructueux donne la valeur *false* au booléen *canGovern* (de sorte que lorsqu'un rôle est révoqué, l'adresse assignée à ce rôle perd le droit de gouverner).

```
struct Person{
uint40 id;
bytes32 role;

// on ajoute le booléen « canGovern » à la structure « Person »
bool canGovern;
}

. . .

function grantGovernRole (bytes32 _role, address _account) public {
// on crée une condition vérifiant que la fonction n'est appelée que par la
DAO, soit les parties qui auront le NFT-Rôle GOVERNOR
require(hasRole(GOVERNOR,msg.sender), "only governor can create governance
roles");

// cette fonction crée classiquement un rôle(ce qui génère un nft et un
enregistrement dans le mapping...)..
_grantRole(_role, _account);

// ...et elle assigne la valeur de « true » au booléen « canGovern »
persons[_account].canGovern = true;
}

function _revokeRole(bytes32 role, address account) internal override {
Person memory person = persons[account];
_burn(info.id);

// dès lors qu'un rôle est révoqué, la personne détenant ce rôle ne peut plus
```

```

gouverner

person.canGovern = false;

super._revokeRole(role, account);
}

```

Enfin, dans le *smart contract* de gouvernance d'*OpenZeppelin*, nous modifions ses fonctions *propose* et *castVote* afin de faire en sorte que seules les adresses autorisées (les personnes dans le *struct person* qui ont leur booléen *canGovern* sur *true*) puissent proposer des décisions et voter.

```

function propose(address[] memory targets, uint256[] memory values, bytes[]
memory calldatas, string memory description)

public

override(Governor)

returns (uint256)

{

// ajout de la condition vérifiant que seules les membres autorisées puissent
proposer

require(token.persons(msg.sender).canGovern == true, "Only allowed members
can propose.");

super.propose(targets, values, calldatas, description);

}

function _castVote(

uint256 proposalId,

address account,

uint8 support,

string memory reason,

bytes memory params

) internal virtual override returns (uint256) {

// ajout de la condition vérifiant que seules les membres autorisées puissent
voter

require(token.persons(msg.sender).canGovern == true, "Only allowed members

```

```
can vote.");  
}
```

En résumé :

- des NFT peuvent servir à voter dans une DAO (selon les conditions voulues par ses concepteurs : unanimité, majorité, quorum minimum, etc),
- laquelle DAO peut alors être désignée comme le *smart contract* seul autorisé à actionner certaines fonctions.

B – *Smart contract* d'intervention d'oracles

454. Implémentation d'oracles machines. Nous avons déjà défini la notion d'oracle et ses différents types¹¹¹². Dans notre proposition de module, nous nous focaliserons sur les oracles “machines”. Les oracles “humains” peuvent, en effet, être implémentés de manières si différentes qu'il est inutile pour nous d'essayer d'abstraire une méthode et d'en proposer un modèle. En revanche, la requête d'une information quérable sur le web (par le biais d'une API), peut être relativement standard.

Nous mettons donc à disposition des parties un *smart contract* qu'elles pourront intégrer dans le leur afin d'obtenir une information *off-chain* récupérable par voie d'API. Pour ce faire, nous utilisons une solution d'oracle privée pour plusieurs raisons :

- bien que les parties pourraient utiliser leur propre programme requêtant une API, il nous apparaît plus raisonnable de passer par les services d'une personne tierce afin d'éviter tout risque de conflit d'intérêt ;
- autre élément, la tâche de récolter une information extérieure pour l'acheminer vers un *smart contract* est éminemment risquée. Elle introduit une faille dans le *smart contract*¹¹¹³ à laquelle les solutions privées d'oracle apportent souvent des solutions techniques élégantes et fiables ;
- enfin, les conditions matérielles pour réaliser de manière satisfaisante cette tâche peuvent être sophistiquées (garantie d'intégrité de l'information récupérée, haute disponibilité et rapidité d'exécution) et il peut être opportun à cet égard de les externaliser à des experts.

Bien qu'il existe un nombre pléthorique de solutions privées d'oracles (*API3*, *Band*, *Provable*, *Town*

¹¹¹² V., *infra*, §286

¹¹¹³ V., *infra*, §68

Crier, Witnet, Empire...)¹¹¹⁴, celle qui est, de loin, la plus dominante actuellement est *Chainlink*¹¹¹⁵. Elle met à disposition de ses utilisateurs un réseau décentralisé d'oracles qui sont économiquement incités à agir de façon optimale dans les tâches qui leur sont confiées. Le protocole permet donc de décentraliser le travail d'oracle et conserver, dans une certaine mesure, les propriétés de fiabilité du *smart contract* informé¹¹¹⁶.

Les parties peuvent choisir parmi un large panel d'oracles de son réseau, et en fonction de plusieurs critères : spécialités, réputations, prix, etc. Pour conserver les propriétés de décentralisation, plusieurs oracles pourront être sélectionnés afin de récupérer une seule information, qui elle-même pourra provenir de plusieurs sources. Ces informations seront agrégées et lorsqu'un consensus sera formé, elles seront fournies au *smart contract*. En pratique cependant, il sera courant qu'une seule source d'information et/ou qu'un seul oracle spécialisé pour l'acheminer existe pour récupérer la donnée dont les parties auront besoin pour faire fonctionner leur smart contracts. Dans ce cas-là, elles n'auront d'autre choix que de composer avec cette faiblesse dans leur programme¹¹¹⁷.

Notre proposition de module est donc destinée à réaliser une requête d'une source d'information par le biais d'un oracle du réseau *Chainlink*. Elle vise à charger celui-ci de récupérer une information accessible via une API. Cette tâche est proposée par de nombreux oracles dans le réseau *ChainLink* et peut être identifiée par un identifiant unique (*jobId*). Cet identifiant pourra être renseigné dans le *smart contract* des parties afin qu'un oracle comprenne que sa tâche est de requêter une API¹¹¹⁸.

La première fonction de notre *smart contract* consistera donc à définir les paramètres de notre oracle.

¹¹¹⁴ Garg, Priyeshu. « Chainlink, Band Protocol, API3, and Umbrella Network: Exploring the Differences Between Oracles ». Umbrella Network (blog), 1 février 2021. <https://medium.com/umbrella-network/chainlink-band-protocol-api3-and-umbrella-network-exploring-the-differences-between-oracles-9477d975e142>.

¹¹¹⁵ Weston, Georgia. « Top 10 Blockchain Oracles ». 101 Blockchains, 22 février 2023. <https://101blockchains.com/top-blockchain-oracles/>.

The foremost entry among renowned blockchain oracles is Chainlink, the largest blockchain oracle on the market. With a market capitalization crossing slightly over \$1 billion, Chainlink is a strong player in the blockchain oracle space.

¹¹¹⁶ Chainlink. « What Is Chainlink? A Beginner's Guide ». Chainlink Blog, 25 janvier 2021. <https://blog.chain.link/what-is-chainlink/>.

Chainlink, a decentralized oracle network, was developed to allow smart contracts to automate the transfer of data between blockchains and outside systems in a highly secure and reliable manner. It uses a similar model to a blockchain in that there is a decentralized network of independent entities (oracles) that collectively retrieve data from multiple sources, aggregate it, and deliver a validated, single data point to the smart contract to trigger its execution, removing any centralized point of failure.

¹¹¹⁷ V., *infra*, §71

¹¹¹⁸ Chainlink Documentation. « Make a GET Request ». Consulté le 10 juillet 2023. <https://docs.chain.link/any-api/get-request/introduction>.

Seront renseignés l'identifiant de la tâche (le *jobId*), le montant des frais facturés par l'oracle (dans l'unité du jeton LINK) et la variable qui réceptionnera la donnée récupérée.

```
contract apiOracle {

//on importe la bibliothèque chainlink relative aux requêtes de la sorte

using Chainlink for Chainlink.Request;

//les données renseignées seront le « jobId » enregistré dans un bytes32 et
le montant des frais enregistré dans un uint256

bytes32 private jobId;

uint256 private fee;

//la donnée requêtée peut être un string, un bytes32, un booléen, etc

uint256 public dataRequested;

function setChainlinkParameters(address _oracle, address _link, bytes32
_jobId, uint256 _fee) public {

//on utilise une fonction de la bibliothèque de chainlink pour indiquer
l'adresse de l'oracle qu'on souhaite utiliser...

setChainlinkOracle(_oracle);

//on utilise un même type de fonction pour indiquer l'adresse du jeton

setChainlinkToken(_link);

//on renseigne le jobId...

jobId = _jobId;

//...et les frais de l'oracle

fee = _fee;

}
```

Le travail d'un oracle *ChainLink* se fait en deux temps : il faut lui ordonner de réaliser sa requête, puis il peut lui être demandé de renseigner l'information qu'il est parvenu à récupérer. Cela prend

The jobId refers to a specific job for that node to run. Each job is unique and returns different types of data. For example, a job that returns a bytes32 variable from an API would have a different jobId than a job that retrieved the same data, but in the form of a uint256 variable.

donc la forme de deux fonctions : *requestData* et *fulfill*.

```
//cette fonction est chargée d'envoyer une requête
function requestData() public virtual returns (bytes32 requestId) {
    // on construit la requête à l'aide de « buildChainlinkRequest »
    Chainlink.Request memory req = buildChainlinkRequest(jobId, address(this),
    this.fulfill.selector);
    // et on l'envoie à l'aide de « sendChainlinkRequest »
    return sendChainlinkRequest(req, fee);
}
```

Puis il faut actionner la fonction *fulfill* qui va présenter la donnée que l'oracle a récolté après la requête. Cette donnée est enregistrée dans la variable de chiffre *dataRequested*.

```
function fulfill(bytes32 _requestId, uint40 _dataRequested) public virtual
recordChainlinkFulfillment(_requestId) {

    dataRequested = _dataRequested;

}
```

Enfin notre module comprend une fonction *withdrawLink* qui permet de retirer les jetons LINK stockés dans le *smart contract*. En effet, pour faire un appel à un oracle *ChainLink* les parties doivent le rémunérer dans cette crypto-monnaie (spécifié dans la variable *fee* précédemment évoquée). Or comme les requêtes proviennent du *smart contract* des parties, c'est celui-ci qui doit détenir cette somme nécessaire. A la fin de l'exécution du *smart contract*, il peut donc rester des LINK que la fonction *withdrawLink* permet de retirer.

```
function withdrawLink() public virtual {

    // les parties pourront prévoir un require qui conditionne l'interaction avec cette
    fonction au fait que celle-ci (msg.sender) soit l'adresse de leur multisig par
    exemple

    LinkTokenInterface link = LinkTokenInterface(chainlinkTokenAddress());

    require(link.transfer(msg.sender, link.balanceOf(address(this))), "Unable to
    transfer");
}
```

C – *Smart contract* de règlement des litiges

455. **Règlement des conflits *on-chain*.** Le dernier module optionnel que nous proposons est celui de MARD *on-chain*. Comme déjà discuté, les parties recourant à un *smart contract* doivent avoir pour principe de limiter les phénomènes d’hybridation¹¹¹⁹. Or un lieu où elles peuvent opportunément réduire ces interactions *off-chain* et *on-chain*, dans l’exécution de leur contrat, est celui de la résolution des différends. Aussi nous proposons un *smart contract* destiné à régler l’allocation de jetons en cas de dispute sur ceux-ci.

Imaginons un contrat de prestation de service formalisé *on-chain*. Les sommes devant être payées au prestataire sont séquestrées dans un *smart contract*. Si il y a un litige à la fin de la prestation, le module que nous proposons pourra être utilisé afin d’envoyer ces sommes vers un autre *smart contract*, qui les tiendra aussi en séquestre et les délivrera aux parties en fonction de la décision d’un ou plusieurs experts. Le *smart contract* que nous proposons aura, à cet égard, un fonctionnement très similaire à celui du séquestre conventionnel¹¹²⁰. Les jetons dont on pourra allouer l’allocation seront les jetons fongibles ERC-20 et les jetons non-fongibles ERC-721.

456. **Fonctionnement général du *smart contract* de résolution des conflits.** Ce *smart contract* proposé devra être déployé à part de celui exécutant le contrat des parties. Afin d’interagir plus ergonomiquement avec celui-ci, les parties seront avisées de le draper d’une application décentralisée. Dans leur smart contracts, les parties pourront prévoir, où elles le souhaitent, d’envoyer les jetons litigieux via la fonction *deposit* de notre programme. Par exemple, elles pourront rendre possible l’actionnement de cette fonction après le moment où un acheteur a payé un bien et dispose d’une certaine durée pour indiquer s’il est satisfait. Dès lors que la fonction sera activée, une nouvelle affaire sera déclarée ouverte, et les jetons ou le NFT seront séquestrés dans le *smart contract* du MARD le temps de sa résolution. Après avoir pris leur décision, les experts assigneront à chaque partie impliquée la proportion de jetons qu’ils estiment leur être dus, puis déclareront l’affaire close. Dès cet instant, chacun pourra retirer la proportion de jetons qui lui a été assignée.

¹¹¹⁹ V., *infra*, §68

¹¹²⁰ V., *infra*, §77

457. Variables du smart contract. Les premières variables que nous déclarons dans notre programme sont l'adresse de l'expert et sa commission¹¹²¹. Pour cette dernière, nous avons choisi ce mode de rémunération car il est facilement implémentable *on-chain*. Les arbitres pourront se payer sur la somme des jetons séquestrées, s'ils sont des jetons fongibles. S'il s'agit de NFT qui sont objets de la dispute et du séquestre, les experts devront être payés *off-chain*. Concernant l'adresse des experts, on postule qu'elle sera celle du *multi-sig* constitué entre eux (comme pour les parties). Chaque interaction de l'adresse *Arbiter* nécessitera donc l'aval de chacun d'entre eux.

Nous avons ensuite créé une structure afin de représenter une affaire soumise devant les arbitres. Elle renseigne notamment sur :

- l'état de l'affaire (ouverte ou non),
- l'adresse du jeton litigieux utilisé,
- l'information s'il s'agit d'un NFT ou de jetons fongibles,
- selon la donnée précédente, l'identifiant du jeton en cas de NFT ou le nombre total de jetons s'il s'agit de jetons fongibles,

Une autre structure est créée pour représenter chaque partie impliquée au litige. En son sein figurent des informations sur :

- la proportion des jetons qui lui a été assignée, l'indication que la proportion de jeton lui a bien été assignée par les arbitres,
- l'indication que la proportion de jetons assignée à une partie a été retirée par elle.

Enfin, ces structures figurent dans différents *mappings*. Le premier (*cases*) classe les affaires en fonction des *hash* des contrats écrits dont ils proviennent (eux-mêmes issus de la fonction *setAgreementHash*); schématiquement le tableau ressemble à ceci :

Hash de l'accord écrit du <i>smart contract</i>	Affaire
0x7f83b1657ff1fc53b92dc18148a1d65dfc2d4b1fa3d677284addd200126d9069	struct Case {état de l'affaire, adresse du jeton litigieux, information s'il s'agit d'un NFT ou non, id du NFT ou somme totale envoyée }
0x.....	struct Case {état de l'affaire, adresse du jeton litigieux, information s'il s'agit d'un NFT ou non, id du NFT ou somme totale envoyée }

¹¹²¹ Cour d'appel de Versailles - 3e chambre 17 novembre 2022 / n° 20/05505 (s. d.).

(...). En effet, les frais d'arbitrage sont prélevés à l'occasion d'une opération d'arbitrage, soit par pourcentage sur l'assiette du montant arbitré, soit en fonction d'un montant forfaitaire.

Le second *mapping* répertorie les structures représentant les parties (*Party*). Chaque structure *Party* est liée à l'adresse d'une partie, qui est elle-même liée à une affaire. Il en résulte un double mapping ressemblant à ceci :

Hash du contrat écrit	Adresse	Partie
0x7f83b1657ff1fc53b92dc18148a1d65dfc2d4b1fa3d677284a ddd200126d9069	0xb794f5ea0ba39494ce839613 fffba74279579268	struct Party

```
contract arbitration {

//adresse du multisig
address public arbiter;

//commission (en %) des arbitres
uint256 public rate;

//structure représentant une affaire
struct Case {

uint256 totalValueAssigned;

uint256 totalValue;

uint256 fee;

bool open;

address token;

bool isNft;

}

//structure représentant une partie
struct Party {

//cette valeur = montant de jetons assignées ou l'id du NFT
uint256 valueGiven;

bool valueIsWithdrawn;
```

```

bool valueIsGiven;

}

//mapping répertoriant les struct affaires classées en fonction des hash des
contrats écrits
mapping(bytes32 => Case) public cases;

//mapping répertoriant les parties classées en fonction de leur adresses,
elles-mêmes attachées aux hash des contrats écrits

mapping(bytes32 => mapping (address => Party)) public parties;

```

Lors du déploiement du *smart contract*, les experts assigneront l'adresse de leur *multisig*, ainsi que le pourcentage de leur commission.

```

constructor (address _arbiter, uint256 _rate) {

require(_rate > 0 && _rate < 100, "Rate must be a percentage");

arbiter = _arbiter;

rate = _rate;

}

```

Le *smart contract* contiendra quatre fonctions principales. La première *deposit* chargée de recevoir et séquestrer les jetons fongibles ou le jeton non fongible, ainsi qu'ouvrir une nouvelle affaire en renseignant toutes les informations nécessaires. Pour cela, elle prend comme paramètres :

- le *hash* du contrat écrit (afin de renseigner les *mapping cases* et *parties*),
- l'adresse du jeton,
- l'indication s'il s'agit d'un NFT ou d'un jeton fongible,
- son identifiant ou son nombre en fonction de sa nature.

```

function deposit (bool _isNft, bytes32 _agreementHash, address _token, uint256
_valueSent) public {

```

```

//vérifie qu'une affaire avec le même hash n'est pas déjà ouverte
require (cases[_agreementHash].open == false, "Case is already open");

//créé une nouvelle affaire
Case storage currentCase = cases[_agreementHash];

//récupère l'information si le jeton est un nft
currentCase.isNft = _isNft;

//récupère l'adresse du jeton
currentCase.token = _token;

//calcule la commission si le jeton n'est pas fongible
if (_isNft == false) {

currentCase.fee = _valueSent*rate/100;

currentCase.totalValue = _valueSent - currentCase.fee;

} else {

currentCase.totalValue = _valueSent;

}

currentCase.open = true;

//envoie les jetons fongibles ou le NFT en séquestre au smart contract
safeTransferFrom(_token, msg.sender, address(this), _valueSent);
}

```

La deuxième fonction principale de notre *smart contract* est *assignValue*, permettant aux experts d'assigner la proportion de jetons qu'ils estiment due à chaque partie impliquée dans une affaire. Autrement dit, après que les experts aient pris leur décision, ils exprimeront par le biais de cette fonction la quantité de jetons qui doit revenir à une partie impliquée. La fonction permet d'assigner la valeur qu'à une seule partie, il faudra donc l'utiliser autant de fois que nécessaire.

```

function assignValue (bytes32 _agreementHash, address _addressParty, uint256
_valueToAssign) public {

//il est vérifié que c'est bien le wallet des experts qui interagit avec cette
fonction et que l'affaire n'est pas close

```

```

require (msg.sender == arbiter, "Msg.sender must be arbiter");

require (cases[_agreementHash].open == true, "Case must be opened");

. . .

//la portion de jetons fongibles ou l'id du NFT est assignée à la partie
renseignée par les experts

parties[_agreementHash][_addressParty].valueGiven = _valueToAssign;

parties[_agreementHash][_addressParty].valueIsGiven = true;

}

```

La troisième fonction principale est *closeCase*. Comme son nom l'indique, elle est celle qui permet aux experts d'indiquer qu'une affaire est terminée. Cela signifie que toutes les parties impliquées dans l'affaire se sont vues assignées leur proportion de jetons. Dès cet instant, les experts pourront récupérer leur commission après que la fonction ait vérifié que la somme totale assignée soit égale à la valeur totale soumise lors de l'ouverture de l'affaire.

```

function closeCase (bytes32 _agreementHash) public {

//vérifie que l'affaire n'est pas déjà fermée et que c'est bien le wallet des
arbitres qui appelle la fonction

require(cases[_agreementHash].open == true, "Case is already closed");

require(msg.sender == arbiter, "Msg.sender must be arbiter");

//vérifie que la valeur totale assignée est celle envoyée lors de l'ouverture
de l'affaire

require(cases[_agreementHash].totalValue ==
cases[_agreementHash].totalValueAssigned, "Total value assigned doesn't match total
value sent");

//indique que l'affaire est close

cases[_agreementHash].open = false;

    458. //si les jetons envoyés dans l'affaire étaient fongibles, les
experts peuvent recevoir leur commission

```



```

if (cases[_agreementHash].isNft == false) {
    safeTransfer(cases[_agreementHash].token, msg.sender, cases[_agreementHash].fee);
}
}

```

Enfin, la dernière fonction principale du *smart contract* est *withdraw*, qui permet aux parties de retirer la portion de jetons qui leur a été assignée.

```

function withdraw (bytes32 _agreementHash) public {
    // vérifie que la valeur assignée n'a pas déjà été retirée et que l'affaire
    est close
    require(parties[_agreementHash][msg.sender].valueIsWithdrawn == false, "Value is
    already withdrawn");
    require(cases[_agreementHash].open == false, "Case is still open");
    //transfère la valeur assignée qui était séquestrée dans le smart contract et
    indique qu'elle a bien été retirée par la partie
    safeTransfer(cases[_agreementHash].token, msg.sender,
    parties[_agreementHash][msg.sender].valueGiven);
    parties[_agreementHash][msg.sender].valueGiven = 0;
    parties[_agreementHash][msg.sender].valueIsWithdrawn = true;
}

```

459. Conclusion du chapitre II. Les parties disposent donc de modules de bases et optionnels qu'elles pourront choisir d'intégrer aux smart contracts exécutant leur contrat. Nous estimons que quel que soit le contrat intelligent qu'elles implémenteront, il leur sera utile d'intégrer une fonctionnalité pour lier par *hash*, selon la méthode ricardienne, leur smart contracts avec l'accord écrit qu'il implémente. Elles pourront aussi avoir grand besoin d'un module pour organiser l'accès de certaines fonctions à différents rôles et un autre pour interrompre, mettre à jour et détruire leur smart contracts. Optionnellement, nous proposons un module leur permettant de céder leur qualité de partie dans la *blockchain*, un autre pour faire intervenir un oracle récupérant une information d'une API et un dernier pour résoudre des litiges sur l'allocation de jetons.

CONCLUSION GENERALE

460. Contextes d'utilisation du contrat intelligent. Notre travail a débuté par l'établissement des contextes d'utilisation d'un contrat intelligent les plus pertinents. Au stade de sa conception en effet, les parties doivent commencer par sélectionner les processus contractuels qui sont les plus adéquats à être exécutés dans la *blockchain*. Il apparaît que ce sont ceux dans lesquels a lieu un transfert d'actif, lequel actif est préférablement immatériel et matérialisable par un jeton¹¹²², qui se révèlent être les plus enclins à une formalisation *on-chain*. Ces processus se retrouvent surtout dans les contrats onéreux. Le fonctionnement d'un *smart contract* rappelant fortement celui de l'opération de séquestre¹¹²³, nous avons identifié que les contrats onéreux ayant cette structure figurent parmi ceux convenant particulièrement bien à une transformation en contrat intelligent.

461. Forme du contrat intelligent. Une fois le domaine matériel du contrat intelligent déterminé, il faut encore que les parties conçoivent sa forme. Le contrat intelligent peut :

- être formé uniquement par les smart contracts l'exécutant, et donc n'avoir que le code source de ces derniers pour détailler le rapport contractuel,
- être, en plus, pourvu d'un document écrit en langage naturel qui renseigne les droits et obligations des parties, concurremment avec le *smart contract*,
- enfin, être doté d'un document écrit en langage naturel qui seul renseigne la volonté des parties et dont les smart contracts ne servent qu'à exécuter ses stipulations.

Cette dernière forme de contrat intelligent est celle que les parties seront avisées de choisir. Nous l'avons nommée la forme ricardienne car elle correspond à la philosophie des contrats ricardiens¹¹²⁴ ; et par ailleurs parce que nous préconisons que l'accord écrit en langage dans ces contrats soit relié aux smart contracts de la même manière que Ian Grigg l'avait imaginé dans son article éponyme¹¹²⁵.

462. Contenu de l'accord écrit en langage naturel du contrat intelligent. Ainsi les parties rédigeront un accord écrit en langage naturel, qui contiendra des clauses relatives au recours à la *blockchain*. En particulier, elles détailleront la manière dont les smart contracts pourront être

¹¹²² V., *infra*, §59

¹¹²³ V., *infra*, §75

¹¹²⁴ V., *infra*, §217

¹¹²⁵ V., *infra*, §209

suspendus ou interrompus en cas de bogues, piratages ou tout évènement fortuit les faisant dévier de ce qu'elles avaient convenu par écrit en langage naturel¹¹²⁶. Elles prendront également le soin d'expliquer les modalités d'exécution des obligations du contrat à travers les smart contracts¹¹²⁷, et notamment la manière dont elles recourront aux *stablecoin* pour réaliser leurs obligations de paiement¹¹²⁸.

463. Choix de la *blockchain* du contrat intelligent. En ce qui concerne l'implémentation technique du contrat intelligent, les parties devront commencer par déterminer la *blockchain* sur laquelle elles déploieront leur smart contracts. L'écosystème Ethereum apparaît comme le plus fiable et pratique pour accueillir ces contrats intelligents. Si les parties cherchent une *blockchain* suffisamment expressive mais très résiliente et sécurisée pour soutenir des contrats à fort enjeu, la première couche d'Ethereum (le L1¹¹²⁹) sera le choix optimal. Pour les contrats plus ordinaires, un *rollup* connecté à Ethereum¹¹³⁰ fournira un bon équilibre entre sécurité et performance. Enfin pour les contrats nécessitant des adaptations spécifiques, les kits de développement de *blockchain* privées de *Cosmos* ou de *rollup* spécifiques de *StarkEx* constitueront, à notre humble avis, les choix les plus judicieux.

464. Développement des smart contracts du contrat intelligent. Suivant la rédaction de l'accord écrit et après avoir choisi leur *blockchain*, les parties pourront développer et déployer les smart contracts de leurs contrats intelligents. Pour diminuer le risque d'apparition de bogues et/ou de piratages dans ces programmes, elles devront largement recourir aux librairies *OpenZeppelin*¹¹³¹ ; qui fournissent des smart contracts *open-source* et audités par une vaste communauté de développeurs. A l'aide de celles-ci, nous avons développé plusieurs modèles de *smart contract* que des parties pourraient intégrer dans leur implémentation de contrat intelligent, quel que soit l'objet de ce dernier. Nous proposons :

- un module pour mettre à jour les smart contracts en cas de problèmes, permettant leur recodification et redéploiement,

¹¹²⁶ V., *infra*, §272

¹¹²⁷ V., *infra*, §263

¹¹²⁸ V., *infra*, §268

¹¹²⁹ V., *infra*, §332

¹¹³⁰ V., *infra*, §391

¹¹³¹ V., *infra*, §406

- un autre pour contrôler de façon sécurisée l'accès aux smart contracts,
- et un dernier pour lier l'accord écrit de manière "ricardienne" aux smart contracts.

En outre, nous suggérons d'autres modules pouvant être utiles aux parties pour leurs contrats intelligents :

- un module pour permettre la cession du contrat intelligent,
- un autre pour l'intégration sécurisée d'un oracle dans un *smart contract*,
- et un dernier pour la résolution de conflits *on-chain* liés à l'allocation de jetons.

465. Le risque d'une fausse impression. A travers cette méthodologie, nous espérons avoir fait la démonstration de l'opportunité du recours à la *blockchain* comme infrastructure d'exécution des contrats. Dès aujourd'hui, la technologie des smart contracts peut être mobilisée afin de créer des contrats intelligents présentant beaucoup plus d'avantages que des contrats « manuellement exécutés » ou des contrats automatisés par des programmes ordinaires. Cependant, nous reconnaissons un biais dans nos travaux dont nous souhaitons préserver nos lecteurs. En effet, nous craignons avoir pu donner l'apparence de traiter la *blockchain* comme une infrastructure tout juste meilleure qu'une autre pour exécuter les contrats. Certes les smart contracts rendent plus résiliente et commode l'exécution des contrats¹¹³², mais cela est loin de constituer leur unique atout. Cette technologie abrite un si grand potentiel disruptif que ce serait un tort de la considérer et sous-exploiter ainsi.

En effet, la *blockchain* permet des applications si novatrices en matière de contrôle d'actifs, que les juristes ne doivent surtout pas se réduire à l'utiliser comme un simple outil marginalement meilleur que d'autres pour exécuter leurs contrats. **Ils doivent aussi et surtout chercher à s'en emparer afin de révolutionner l'exécution de certaines conventions.** Par exemple, nous avons vu que les smart contracts permettent le versement en temps réel de sommes d'argent¹¹³³. Cette innovation technique peut bouleverser la manière dont les contrats de prestations de services ont toujours été exécutés. Dans un contrat de coiffure, un coiffeur reçoit la plupart du temps le prix de sa prestation à la fin de son travail. Quid si il existait une technologie lui permettant, très aisément, de recevoir un centime d'euro du compte de son client chaque seconde qu'il passe à le coiffer ? Cette prestation serait peut-être transformée pour le meilleur, puisque le coiffeur serait réellement rémunéré pour le temps passé

¹¹³² V., *infra*, §26 et §28

¹¹³³ V., *infra*, §171

à faire son travail, tandis que le client paierait exactement la durée de la prestation. De manière générale, cette fonctionnalité pourrait changer la manière dont les services sont habituellement réalisés. Un consultant pourrait facturer des micro-consultations de quelques minutes, un artisan pourrait faire rémunérer ses conseils avisés de quelques secondes (qui ne débouchent pas toujours sur l'acceptation d'un devis) et des services de *streaming* pourraient faire payer la consommation de leur contenus au temps regardé, plutôt qu'à un tarif forfaitaire mensuel.

466. Juriste-innovateur. Le juriste doit ainsi être à l'affût de ces innovations et réfléchir à leur appropriation pour bouleverser l'exécution de certains contrats. Nous arguons qu'il est le plus à même de les déceler et les appliquer dans le domaine contractuel. Ce sont des juristes qui savent à quel point le droit de suite¹¹³⁴ est peu mis en œuvre malgré son existence dans les textes, et comment il peut trouver une nouvelle vigueur grâce aux NFT¹¹³⁵. Ce sont aussi les juristes qui savent combien il peut être difficile pour les prêteurs de recouvrer leurs créances en cas de défaut de paiement de leurs débiteurs, alors que certains mécanismes de garantie basés sur les smart contracts peuvent renforcer les garanties les plus efficaces.¹¹³⁶

En conclusion, nous faisons le vœux que les juristes ne restent pas pris dans le piège de la posture d'analystes et critiques, et embrassent plutôt celle d'acteurs de l'innovation.

¹¹³⁴ Article L. 122-8 du Code de la propriété intellectuelle : *Les auteurs d'œuvres graphiques et plastiques ont, nonobstant toute cession de l'œuvre originale, un droit inaliénable de participation au produit de toute vente de cette œuvre faite aux enchères publiques ou par l'intermédiaire d'un commerçant.*

¹¹³⁵ Jean Martin et Pauline Hot. « Rapport de mission du Conseil supérieur de la propriété littéraire et artistique (CSPLA) sur les jetons non fongibles (JNF ou NFT en anglais) », p.26, Conseil supérieur de la Propriété littéraire et artistique, 12 juillet 2022. <http://www.vie-publique.fr/rapport/286012-mission-sur-les-jetons-non-fongibles-cspla>.

Les JNF doivent permettre de faciliter le versement de « royalties » aux auteurs, là où le versement du droit de suite est soumis à des conditions légales strictes.

¹¹³⁶ Mekki, Mustapha. « Les mystères de la blockchain ». Recueil Dalloz, n° 37 (2 novembre 2017): 2160.

On peut également imaginer les apports d'un smart contract en droit des sûretés. La garantie à première demande justifiée pourrait être mise en œuvre automatiquement dès lors qu'un document prédéterminé dans le programme est remis au garant.

Abdoulaye DIALLO. « Smart contract d'une garantie autonome au sens de l'article 2321 du code civil. » Village de la Justice (blog), 18 novembre 2018. <https://www.village-justice.com/articles/smart-contract-une-garantie-autonome-sens-article-2321-code-civil,29924.html>.

BIBLIOGRAPHIE

I - Sources juridiques

A - Ouvrages

B - Encyclopédies – Fascicules – Répertoires

C - Lois - Rapports – Textes officiels - Décisions

D - Articles de doctrine en français

E - Articles de doctrine en anglais

II - Sources techniques

A – Ouvrages et encyclopédies

B – Rapports

C – Articles

I – Sources juridiques

A - Ouvrages

Allen Jason et Hunn Peter. *Smart Legal Contracts: Computable Law in Theory and Practice*. Oxford University Press, 2022.

Aulagnier Jean, Aynès Laurent, et Bertrel Jean-Pierre. *Le Lamy Patrimoine*, Wolters Kluwers, 2015.

Blanc Nathalie, Boffa Romain, et Mazeaud Denis. *Dictionnaire du contrat*. LGDJ, 2018.

Bruguière Jean-Michel. *Les standards de la propriété intellectuelle*. Dalloz. Thèmes & commentaires, 2018.

Buy Frédéric, Lamoureux Marie, Mestre Jacques, et Roda Jean-Christophe. *Les principales clauses des contrats d'affaires*. Les Intégrales. LGDJ, 2018.

Chambre de commerce internationale. *Règles et usances uniformes de l'ICC relatives aux crédits documentaires : entrée en vigueur 1er juillet 2007, révision 2007*. ICC Publications, 2007.

Collectif. *Dictionnaire permanent Droit des affaires*. Editions législatives, 2022.

Debard Thierry et Guinchard Serge. *Lexique des termes juridiques 2020-2021* - 28e éd. Dalloz, 2020.

De Filippi Primavera et Wright Aaron. *Blockchain and the Law: The Rule of Code*. Harvard University Press, 2018.

Deveze Jean. *Le Lamy Droit du financement*. Lamy Expert, 2021.

Directorate-General for Financial Stability, Financial Services and Capital Markets Union (Commission Européenne). *Decentralized Finance: Information Frictions and Public Policies : Approaching the Regulation and Supervision of Decentralized Finance*. LU: Publications Office of the European Union, 2022.

Dissaux Nicolas, Chantepie Gaël, Auque Françoise, Deroussin David, Guerlin Gaëtan, Houtcieff Dimitri, Joyeux Arthur, et al. *La stylistique contractuelle*. Thèmes et commentaires Études. Dalloz, 2022.

Dross William. *Clausier ; dictionnaire des clauses ordinaires et extraordinaires des contrats de droit privé interne*. 4e éd. LexisNexis, 2020.

Guégan Elizabeth. *Blockchain et assemblées d'actionnaires*. Dalloz. Blockchain et droit des sociétés, 2019.

Hirigoyen Gérard, Couret Alain, et Devèze Jean. *Lamy Droit du financement*. Wolters Kluwers, 2022.

Hobbes Thomas. *Léviathan ou Matière, forme et puissance de l'État chrétien et civil*. Gallimard. Folio Essais, 1651.

Jamin Christophe, Billiau Marc, et Ghestin Jacques. *Les effets du contrat*. Traité de Droit civil. 3e éd. L.G.D.J, 2001.

Kenfack Hugues, Dumont-Lefrand Marie-Pierre, Astegiano-La Rizza Axelle, Colomer Patrick, Denizot Christophe, Maublanc Jean-Pierre, Reille Florence, Reygrobelle Arnaud, Schmit François. *Droit et pratique des baux commerciaux*. Dalloz Action. 6e éd., 2020.

Legeais Dominique.

Droit des sûretés et garanties du crédit. 15e éd. LGDJ, 2022.

Blockchain et actifs numériques. 2e édition. LexisNexis, 2021.

Le Tourneau Philippe. *Contrats du numérique 2022-2023 - Informatiques et électroniques*. 12e éd. Dalloz Référence, 2022.

Mousseron Pierre, Raynard Jacques, et Seube Jean-Baptiste. *Technique contractuelle*. 5e éd. Francis Lefebvre, 2017.

Sirinelli Pierre et Vivant Michel. *Formulaires ProActa Droit de l'immatériel*. Lamy Expert, 2021.

Testu François-Xavier. *Contrats d'affaires*. Dalloz référence, 2011.

Vergès Étienne. *Contrats sur la recherche et l'innovation*, Dalloz, 28 novembre 2018.

Vernières Christophe. *Guide de la rédaction des actes notariés: Actes courants - Immobilier, Famille - Patrimoine, Entreprise, Rural*. Defrenois, 2022.

Vivant Michel. *Le Lamy droit du numérique*. Editions Lamy, 2012.

B - Encyclopédies – Fascicules – Répertoires

Auckenthaler Franck. *Fasc. 2050 : Instruments financiers à terme ou contrats financiers*, JurisClasseur Sociétés, LexisNexis, 1 août 2020.

Bassilana Eva Mouial et Racine Jean-Baptiste. *Fasc. 9-1 : Contrat. – Contenu du contrat : objet du contrat*. JurisClasseur Notarial Répertoire, LexisNexis, mars 2018.

Bénédicte François. *Fiducie – Constitution de la fiducie*, Répertoire des sociétés, Dalloz, septembre 2011.

Gréau Fabrice. *Répertoire de droit civil*, Dalloz, juin 2017.

Lecourt Arnaud. *Répertoire IP/IT et Communication Droit des sociétés et numérique*, Dalloz, novembre 2020.

Legeais Dominique

Fasc. 534 : Blockchain, JurisClasseur Commercial, LexisNexis, 1 juin 2023.

Fasc. 535 : Actifs numériques et prestataires sur actifs numériques, JurisClasseur Commercial, LexisNexis, 14 octobre 2019.

Pignarre Geneviève. *Répertoire de droit civil - Prêt – Prêt de consommation - Chapitre 2*, Dalloz, janvier 2016

Salgado Maria-Beatriz. *Fasc. Q-30 : SOCIÉTÉS ANONYMES. – Pactes d'actionnaires et clauses de préemption non statutaires*. JurisClasseur Sociétés Formulaire, LexisNexis, Date NC.

Schütz Rose-Noëlle. *Répertoire de droit civil - Crédit-bail*, Dalloz, octobre 2015.

Toledo-Wolfsohn Anne-Marie. *Répertoire de droit civil - Compensation et compte-courant - Article 3 - §4*, Dalloz, avril 2017.

C - Lois - Rapports – Textes officiels - Décisions

Décisions – Avis

Autorité de la Concurrence. Avis 23-A-08 du 29 juin 2023 portant sur le fonctionnement concurrentiel de l'informatique en nuage ("cloud")

Cass civ, 6 mars 1876, D. 1876, 1, p. 193.

Cass civ, 22 juillet 1954, Bull. civ. IV, n°576.

Cass. com., 21 juin 1950, D. 1951 p. 749, note Hamel .

Cour d'appel de Versailles - 3e chambre 17 novembre 2022 / n° 20/05505.

Décision n° 2014-1102 du 30 septembre 2014 portant sur la définition des marchés pertinents de la téléphonie fixe, la désignation d'opérateurs exerçant une influence significative sur ces marchés et les obligations imposées à ce titre.

Délibération de la Commission Nationale de l'Informatique et des Libertés ; SAN-2020-003 du 28 juillet 2020

Lois :

Code civil – 2022

Code de commerce – 2022

Loi n°2016-1691 du 9 décembre 2016 relative à la transparence, à la lutte contre la corruption et à la modernisation de la vie économique.

Ordonnance n°2016-131 du 10 février 2016 portant réforme du droit des contrats, du régime général et de la preuve des obligations

Ordonnance n° 2016-520 du 28 avril 2016 relative aux bons de caisse

Ordonnance n°2017-1674 du 8 décembre 2017 relative à l'utilisation d'un dispositif d'enregistrement électronique partagé pour la représentation et la transmission de titres financiers.

Règlement n°2023/1114 du parlement Européen et du Conseil sur les marchés de cryptoactifs, et modifiant la directive (UE) 2019/1937 .

Règlement n°593/2008 du Parlement européen et du Conseil du 17 juin 2008 sur la loi applicable aux obligations contractuelles (Rome I).

Règlement du parlement européen et du conseil fixant des règles harmonisées pour l'équité de l'accès aux données et de l'utilisation des données (règlement sur les données) com/2022/68

Règlement 2016/679 du Parlement européen et du Conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE.

Règlement n° 910/2014 du Parlement européen et du Conseil du 23 juillet 2014 sur l'identification électronique et les services de confiance pour les transactions électroniques au sein du marché intérieur et abrogeant la directive 1999/93/CE,

Vocabulaire des actifs numériques (liste de termes, expressions et définitions adoptés) issu du journal officiel de la république Française n°0013 du 15 janvier 2021

Rapports :

Autorité de contrôle prudentiel et de résolution. *Finance « décentralisée » ou « désintermédiée » : quelle réponse réglementaire ?*, 3 avril 2023.

Autorité des marchés financiers. *Analyse sur la qualification juridique des produits dérivés sur crypto-monnaies*, 22 octobre 2018.

Comité européen de la protection des données. *Lignes directrices 07/2020 concernant les notions de responsable du traitement et de sous-traitant dans le RGPD*, 7 juillet 2021.

De La Raudière Laure et Mis Jean-Michel. *Mission d'information commune sur les chaînes de blocs (blockchains)*. Assemblée Nationale, 12 décembre 2018.

European Banking Authority. *Report with Advice for the European Commission on Crypto-Assets*, 9 janvier 2019.

European Central Bank. *Report on a digital euro*, octobre 2020.

Favreau Amélie. *Smart contracts - La première librairie européenne et ouverte de smart contracts à destination des professionnels du droit et de la justice*. Mission de recherche droit et justice, 11 février 2019.

Finck Michèle. *Blockchain and the General Data Protection Regulation | Can distributed ledgers be squared with European data protection law ?*, European Parliamentary Research Service, juillet 2019.

Lawtech UK. *UKJT Digital Dispute Resolution Rules & Guidance*, 8 mai 2021.

Lord Chancellor and Secretary of State for Justice. *Smart legal contracts - Advice to Government*, novembre 2021.

Martin Jean et Hot Pauline. *Rapport de mission du Conseil supérieur de la propriété littéraire et artistique (CSPLA) sur les jetons non fongibles (JNF ou NFT en anglais)*. Conseil supérieur de la Propriété littéraire et artistique, 12 juillet 2022

Toledano Joelle et Janin Lionel. *Les enjeux des blockchainss*. France Stratégie, juin 2018.

Thèses :

Barbet-Massin Alice. *Le droit de la preuve à l'aune de la blockchain*. Université de Lille, 2020.

Hvited Tom. *Contract Formalisation and Modular Implementation of Domain-Specific Language*. University of Copenhagen, 2012.

Leveneur Claire. *Les smart contractss : étude de droit des contrats à l'aune de la blockchain*. Université Paris-Panthéon-Assas, 2022.

Marjault Yvan. *Les obligations disjonctives : étude des obligations alternatives et facultatives*. Université du Mans, 2016.

Ozan Oğuz Ceran. *Enhancing letters of credit with blockchain and smart contracts*. Master Thesis, University of Tilburg, 2019.

D - Articles de revues, journaux et colloques français :

Adele Jean-François et Poracchia Didier. *Nantissement de titres financiers enregistrés sur un DLT (blockchain) : reconnaissance d'une universalité fictive*. *Entreprise et expertise Juridique, Option Finance*, n° 1494, 21 janvier 2019.

Ahoulouma Fortuné. *La « tokenisation » des valeurs mobilières dans l'espace OHADA*, *Revue Lamy droit des affaires*, n° 149, 1 juin 2019.

Ancel Bruno. *Contrats - Les smart contractss : révolution sociétale ou nouvelle boîte de Pandore ? Regard comparatiste*. *Communication Commerce électronique* n° 7-8, Juillet 2018, étude 13

Audit Mathias. *Le droit international privé confronté à la blockchain*. *Revue Critique de Droit International Privé*, n° 4, 2021, p. 669.

Barbet-Massin Alice et O'Rorke William. *Fiche pratique n° 4317, Blockchain et données personnelles*, LexisNexis360, n° 4317, 15 octobre 2019.

Barbry Éric. *Smart contracts... Aspects juridiques !* Annales des Mines - Réalités industrielles Août 2017, n° 3, p.77-80.

Barreau Catherine. *La régulation des smart contractss et les smart contractss des régulateurs*. Annales des Mines - Réalités industrielles Août 2017, n° 3, p. 74-76.

Benghozi Pierre-Jean. *Blockchain - Blockchain : objet à réguler ou outil pour réguler ?*, La Semaine Juridique Entreprise et Affaires, n° 36, 7 septembre 2017.

Berlaud Catherine. *Garanties d'un prêt : nantissement et caution*, Gazette du Palais, n° 41, 13 décembre 2022, p. 24.

Bernard-Ménoret Ronan. *Les clauses de recours aux MARC : les pièges à éviter*. Les Petites Affiches, n° 241, 3 décembre 2009, p. 20.

Boismain Corinne. *Quelques réflexions sur les contrats intelligents (smarts contracts)*, Petites affiches, n° 42, 1 mars 2021.

Bordais Pierre. *Finance décentralisée et NFT (non fungible token) : deux nouvelles innovations de la blockchain*, Revue de droit bancaire et financier, n° 6, 1 novembre 2021.

Bordet Hugo. *DAO : une nouvelle forme d'activisme pour les associations ?* Juris associations 2022, n°670, p.27

Biguenet-Maurel Cécile. *Force majeure et imprévision : des outils de droit commun pour faire face au Covid-19*, La quotidienne Francis Lefebvre - Affaires et exécution, 15 avril 2020.

Bouthinon-Dumas Hugues. *Les contrats relationnels et la théorie de l'imprévision*. Revue internationale de droit économique, n° 3, 2001, p. 339.

Capdeville Lasserre Jérôme. *[Jurisprudence] De quelques précisions intéressant le bitcoin et le prêt de bitcoins*. La lettre juridique, mars 2020.

Caprioli Eric. *Mythes et légendes de la blockchain face à la pratique*, Dalloz IP/IT, 2019, p. 429

Cattalano Garance. *Smart contracts et droit des contrats*. AJ Contrats d'affaires - Concurrence - Distribution, 1 juillet 2019, p. 321.

Chafiol Florence et Barbet-Massin Alice. *La blockchain à l'heure de l'entrée en application du règlement général sur la protection des données*, Dalloz IP/IT, 2017, p. 637.

Charpiat Victor et De Vauplane Hubert. [Textes] *La réglementation des initial coin offerings (ICO) en France par la loi «PACTE»*. La lettre juridique, mai 2019.

Corbet-Picard Virginie. *Fiducie sur titres ou sur les actifs sociaux*. Option Finance, La lettre des fusions-acquisitions et du private equity, 26 mars 2021.

Corbi Bertrand. *Les risques des cryptoactifs pour la stabilité financière*, Dalloz Actualité, 19 novembre 2022.

Courtecuisse Matthieu et Davit Ronan. *Les assurances paramétriques au cœur des smart contractss : une révolution pour l'assurance*. Revue d'économie financière 135, n° 3, 2019, p.145-162.

Delpech Xavier.

La délicate appréhension de la Blockchain par le droit. AJ Contrat, n° 244, 2017, p.244

Le projet de loi PACTE, c'est aussi (un peu) du droit des contrats. AJ contrat n°7, 2018, p.300

De Vauplane Hubert. *La nature juridique des stablecoin*. Chronique Digitalisation et droit financier, RTDF, 4 septembre 2019.

Dorchies Olivier. *La clause pénale dans les contrats informatiques et télécoms*, Communication commerce électronique, 2014, n°14.

Favreau Amélie. « *Justice distribuée par une blockchain* » et procédure civile. Dalloz IP/IT, n° 01, 26 janvier 2022, p. 24.

Féraud-Courtin Muriel. *Legaltech : derrière la technologie, l'Homme !*, La lettre des juristes d'Affaires, 24 janvier 2020.

Finck Nathalie et Seroc Samuel. *Notion de faute d'une gravité suffisante justifiant le licenciement d'un salarié protégé*, Gazette du Palais, n° 01, 11 janvier 2022, p. 31.

Gautier Pierre-Yves. *Passions et raison du droit en matière de jeux d'argent*. Pouvoirs n° 139, n° 4, 2011, p. 91 à 101.

Gillioz Fabien. *Du contrat intelligent au contrat juridique intelligent*, Dalloz IP/IT, 2019, p. 16.

Gossa Julien. *Les blockchainss et smart contracts pour les juristes*. Dalloz IP/IT, 2018, p. 393 à 397.

Granier Thierry.

Fiducie sûreté et fiducie gestion, les premiers pas... RTDF, n° 4 2010, p. 98 à 102.

Règlement MiCA : les marchés de crypto-actifs appréhendés par le droit européen Bulletin Joly Bourse, n°05, 30/09/2023, p. 30

Guerlin Gaetan. *Considérations sur les smart contracts*, Dalloz IP/IT, 2017, p. 512.

Guilhaudis Elise. *Comprendre la blockchain à travers l'étude d'un cas pratique : Le covoiturage "BlockCar"*, Lamy Droit de l'immatériel, n° 143, 1 décembre 2017.

Guillaume Florence et Sven Riva. *Libres propos - DAO, code et loi : le régime technologique et juridique de la decentralized autonomous organization*, Revue de droit international d'Assas, n° 4, 13 décembre 2022.

Houtcieff Dimitri.

La réactivité en droit contemporain des contrats : des réactions unilatérales au smart contract. Gazette du Palais, n°3, 19 juin 2019, p. 9 .

Les dispositions de l'article 1195 sont-elles supplétives ?, Gazette du Palais, n° 16, 10 mai 2022, p. 7.

Huet Jérôme. *Le bitcoin, dont la légalité paraît admise, est une sorte de monnaie contractuelle*, Revue des contrats, n° 1, 1 mars 2017, p. 54.

Jarrosson Charles. *Les modes alternatifs de règlement des conflits. Présentation générale*. Revue internationale de droit comparé, n° 2, 1997, p. 325 à 345.

Juhan Jean-Luc. *Pratique du droit du contrat : les clauses limitatives de responsabilité*, Revue Le Lamy Droit civil, n° 99, 1 décembre 2012.

Khadiri Yanis-Said. *Comment faire un apport en nature de crypto-actif au capital d'une société ?* Village-Justice, 17 février 2022.

Laithier Yves-Marie. *A propos de la réception du contrat relationnel en droit français*. Recueil Dalloz, n° 1003, 2006.

Lanskoy S. *La nature juridique de la monnaie électronique*, Bulletin de la Banque de France, n° 70, octobre 1999.

Lapousterle Jean. *Les NFT artistiques à l'épreuve des droits d'auteur*, Dalloz IP/IT, n°2, 2023, p. 84

Lavayssière Xavier. *Blockchain et titres financiers : décret minimaliste pour réforme ambitieuse*. Revue Lamy droit des affaires, n° 144, 1 janvier 2019.

Legeais Dominique.

Blockchain - Blockchain et droit des sociétés - Quelles perspectives ? Quelle incidence véritable de la technologie ?, Droit des sociétés, n° 2, 1 février 2022.

Blockchain et crypto-actifs : état des lieux, RTDcom., n° 754, 2018

Marché financier - 3 QUESTIONS - De nouveaux développements pour la finance décentralisée - Veille par Dominique Legeais, La Semaine Juridique - Entreprise et affaires, n° 52, 26 décembre 2019.

Lucchesi Matthieu. *Stablecoins privés et secteur des paiements Une innovation conditionnée par la réglementation*, Cahiers de droit de l'entreprise, n° 6, novembre 2021.

Lucchesi Matthieu et Raisse Bastien. *Règlement Régime Pilote - Le règlement européen sur le régime pilote : l'innovation réglementaire pour les infrastructures de marché en blockchain face au défi de sa mise en œuvre*, Revue de Droit bancaire et financier, n° 5, octobre 2022.

Marchand Alexis. *Décrypter les enjeux d'une cession ou d'une reprise pour le cédant ou le repreneur afin de donner des conseils pratiques*, Lamy Droit des affaires, n° 116, 1 juin 2016.

Marin Gaetan. *Le bitcoin à l'épreuve de la monnaie*, AJ Contrat, décembre 2017, p. 522.

Marly Pierre-Grégoire et Sorel Arnaud. *Assurance et nouvelles technologies - Les promesses de l'assurance paramétrique*, Responsabilité civile et assurances, n° 3, 1 mars 2023.

Martin Mathieu. *Contrat de l'informatique - Pratique contractuelle. Contrats de l'informatique. Les clauses de convention de preuve*, Communication commerce électronique, n° 3, 1 mars 2021.

Mathis Bruno. *Quel décret d'application pour les ordonnances blockchains ?*, Lamy Droit de l'immatériel, n° 149, 1 juin 2018.

Mekki Mustapha.

Le contrat, objet des smart contractss (Partie 1). Dalloz IP/IT, 1^{er} juillet 2018, n°7, p. 409

Le smart contract, objet du droit (Partie 2). Dalloz IP/IT, 1^{er} janvier 2019, n°1, p. 27

Les mystères de la blockchain. Recueil Dalloz, n° 37, 2 novembre 2017, p. 2160.

If code is law, then code is justice ? Droits et algorithmes. Gazette du Palais, 27 juin 2017, n° 297k2, p. 10

Mis Jean-Michel. *Les technologies de rupture à l'aune du droit*, Dalloz IP/IT, n° 425, 2019.

Mousseron J.-M. *La gestion de risques par le contrat*, RTDciv., n° 481, 1988.

Poullet Yves. *Blockchain : une révolution pour le droit ?* Journal des tribunaux, n° 6748, 10 novembre 2018, p. 815.

Pouliquen Elodie. *Le cautionnement relatif à un bail d'habitation n'est pas soumis aux dispositions du Code de la consommation !*, La revue des Loyers, n° 1027, 1 mai 2022.

Reinhard Dammann et Rotaru Vasile. *La fiducie et le trust : une concurrence inégale*, Recueil Dalloz, 2018, p. 1763.

Roda Jean-Christophe. *Smart contracts, dumb contracts*. Dalloz IP/IT, 4 juillet 2018, p. 397

Rontchevsky Nicolas, Storck Michel et de Ravel d'Esclapon Thibault. *Loi PACTE : innovations et modifications en matière de droit financier*. RTD com. 2019. 713

Schlumberger Edmond. *Réflexions sur la liberté contractuelle dans la SAS*. Mélanges offerts à Michel Germain, 2015, p. 767.

Sommelet Cécile. *Les modalités optimales de l'earn-out*. Option Finance, La lettre des fusions-acquisitions et du private equity, 1 octobre 2019.

Théocharidi Eva. *La conclusion des smart contractss : révolution ou simple adaptation ?*, Revue Lamy Droit civil, n° 161, 1 juillet 2018.

Thibierge-Guelfucci Catherine. *Libres propos sur la transformation du droit des contrats*, RTD Civ., 1997, p.357.

Van Eeckhout Arnould. *La maîtrise des codes sources par le client utilisateur d'un logiciel*. Revue des contrats, n° 4, 1 octobre 2007, p.1335.

Zolynski Célia. *Blockchain et smart contracts : premiers regards sur une technologie disruptive*. Revue de Droit Bancaire et financier étude n°4, n° 1, 1 janvier 2017, p. 84 à 85.

E - Articles de revues, journaux et colloques anglais :

Allen J.G. *Wrapped and Stacked: 'Smart contracts' and the Interaction of Natural and Formal Language*. European Review of Contract Law 14, n° 4, 19 décembre 2018, p. 307 à 343.

Allen Layman E. *Symbolic Logic: A Razor-Edged Tool for Drafting and Interpreting Legal Documents*. The Yale Law Journal 66, n° 6, mai 1957, p. 833.

Allen Layman et Charles Saxon. *Controlling Inadvertent Ambiguity in the Logical Structure of Legal Drafting by means of the Prescribed Definitions of the A-Hohfeld Structural Language*. Articles, University of Michigan Law School, 1 janvier 1994.

Becker Katrin. *Blockchain Matters—Lex Cryptographia and the Displacement of Legal Symbolics and Imaginaries*. Law and Critique 33, n° 2, 1 juillet 2022, p. 113 à 130.

De Filippi Primavera et Samer Hassan. *Blockchain Technology as a Regulatory Technology: From Code is Law to Law is Code*. First Monday, décembre 2016.

Hunn Peter. *Smart contracts as Techno-Legal Regulation*. Journal of ICT Standardization 7, n° 3, 30 septembre 2019, p. 269 à 286.

ISDA.

ISDA Legal Guidelines For Smart Derivative Contracts : Introduction, janvier 2019.

Smart contracts and Distributed Ledger – A Legal Perspective – International Swaps and Derivatives Association, août 2017.

Jones SL Peyton, Eber J.-M., J. Seward et Simon Peyton Jones. *Composing Contracts: An Adventure in Financial Engineering*, 1 septembre 2000.

Larson Dakota A. *Mitigating Risky Business: Modernizing Letters of Credit with Blockchain, Smart contracts, and the Internet of Things*. Law Review 2018, n° 4, 1 janvier 2019.

Sklaroff Jeremy. *Smart contracts and the Cost of Inflexibility*. University of Pennsylvania Law, Review 166, n° 1, 1 janvier 2017, p. 263.

Sophie Goossens et Nick Breen. *NFTs: Ownership in the Metaverse – the Birth of a New Concept*. Reed Smith Guide to the Metaverse, 1 août 2022.

Sreehari P, M Nandakishore, Goutham Krishna, Joshin Jacob, et V. S. Shibu. *Smart will converting the legal testament into a smart contract*. 2017 International Conference on Networks & Advances in Computational Technologies (NetACT), p. 203 à 207, 2017.

Szabo Nick. *Formalizing and Securing Relationships on Public Networks*. First Monday 2, n° 9, 1 septembre 1997.

Tjong Tjin Tai, Eric. *Formalizing Contract Law for Smart contracts*. SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 18 septembre 2017.

Wright Aaron et De Filippi Primavera. *Decentralized Blockchain Technology and the Rise of Lex Cryptographia*. SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 10 mars 2015.

II - Sources non juridiques

A - Ouvrages – Encyclopédies

Barbe-Dandon, Odile, Didelot Laurent et Siegwart Jean-Luc. *Comptabilité approfondie - 2011/2012 - DCG - Épreuve 10 - Corrigés des Applications*. NATHAN - Business & Economics. 2012

Ben Lauwens et Downey Allen B. *Think Julia: How to Think Like a Computer Scientist*. O'Reilly Media, Inc., 2019.

Diry Jean-Charles, Lechat Christine, Pintiaux Serge, Guirao Fabien, Teyssier Monique, Sainty

Catherine, Mousset Bruno, et al. *BLOC 1 - Gérer la relation avec les clients et les fournisseurs de la PME*. Foucher, 2018.

Prudence Jérémy. *NFT pour les Débutants: Achetez, Vendez et Créez Vos Propres NFT Étape Par Étape. Gagnez de l'argent avec l'art numérique, devenez un expert en NFT, en objets de collection cryptographiques*. Jérémy Prudence, 2022.

Séché Alain. *La morale de la machine*. E. Malfère., 1929.

Tanveer Hasnain et Nadeem Javaid. *Using Ethereum Blockchain Technology for Road Toll Collection on Highways*, 2019.

Vigna Paul et Casey Michael J.. *The Truth Machine: The Blockchain and the Future of Everything*. St. Martin's Publishing Group, 2018.

B - Rapports

Agence nationale de la sécurité des systèmes d'information. *Guide de sélection du niveau des signatures et des cachets électroniques*, décembre 2021.

Banque de France. *Fiche 412 - Référentiel des financements des entreprises - Les prêts bancaires classiques*, 15 novembre 2016.

Lawtech UK. *UKJT Digital Dispute Resolution Rules & Guidance*, 8 mai 2021.

L'Organisation de coopération et de développement économiques.

The Tokenisation of Assets and Potential Implications for Financial Markets. OECD Blockchain Policy Series, 17 janvier 2020. www.oecd.org/finance/The-Tokenisation-of-Assets-and-PotentialImplications-for-Financial-Markets.htm.

Why Decentralised Finance (DeFi) Matters and the Policy Implications. OECD Blockchain Policy Series, 19 janvier 2022. <https://www.oecd.org/finance/why-decentralised-finance-defi-matters-and-the-policy-implications.htm>

Faure-Muntian Valéria, De Ganay Claude, et Le Gleut Ronan. *Comprendre les blockchainss : fonctionnement et enjeux de ces nouvelles technologies*, 20 juin 2018. https://www.senat.fr/rap/r17-584/r17-584_mono.html.

C - Articles

Alharby Maher et Aad van Moorsel. *Blockchain-based Smart contracts: A Systematic Mapping Study*. Computer Science & Information Technology, 2017, p. 125 à 140.

Baygin Mehmet, Orhan Yaman, Nursena Baygin et Mehmet Karakose. *A Blockchain-Based Approach to Smart Cargo Transportation Using UHF RFID*. Expert Systems with Applications 188, 1 février 2022

Chu Hanting, Pengcheng Zhang, Hai Dong, Yan Xiao, Shunhui Ji, et Wenrui Li. *A Survey on Smart contract Vulnerabilities: Data Sources, Detection and Repair*. Information and Software Technology n°159, 1 juillet 2023.

Clack Christopher D., Vikram A. Bakshi et Lee Braine. *Smart contract Templates: foundations, design landscape and research directions*. arXiv:1608.00771 [cs], 15 mars 2017.

Cohn Alan, West Travis, et Parker Chelsea. *Smart After All: Blockchain, Smart contracts, Parametric Insurance, And Smart Energy Grids*, Geo. L. Tech. Rev. 273, avril 2017.

De Filippi Primavera, Deffains Bruno et Poux Philémon. « *Maximal Extractable Value* » ou la Tragédie des blockchainss en tant que Communs. Technologie de l'information, culture & société, n° 136, 4 avril 2023.

Drew Mailen. *Why Ethereum Is More Decentralized After the Merge*. Blockworks, 7 novembre 2022.

Egberts Alexander. *The Oracle Problem - An Analysis of How Blockchain Oracles Undermine the Advantages of Decentralized Ledger Systems*. SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 12 décembre 2017.

Enescu Florentina, Fernando Birleanu, Maria Raboaca, Nicu Bizon et Phatiphat Thounthong. A

Review of the Public Transport Services Based on the Blockchain Technology. Sustainability n°14, 12 octobre 2022, p. 13027.

Eskandari Shayan, Salehi Mehdi, Wanyun Catherine Gu, et Clark Jeremy. *SoK: Oracles from the Ground Truth to Market Manipulation*. Proceedings of the 3rd ACM Conference on Advances in Financial Technologies, 2021, p. 127 à 141.

Hoffmann Christian Hugo. *A double design-science perspective of entrepreneurship – the example of smart contracts in the insurance market*. Journal of Work-Applied Management 13, n° 1, 1 janvier 2020, p. 69 à 87.

Kiffer Lucianna, Levin Dave et Mislove Alan. *Analyzing Ethereum's Contract Topology*. Proceedings of the Internet Measurement Conference 2018. Association for Computing Machinery, p. 494 à 499.

Kirli Desen, Couraud Benoit, Robu Valentin, Salgado-Bravo Marcelo, Norbu Sonam, Merlinda Andoni, Antonopoulos Ioannis, Negrete-Pincetic Matias, Flynn David et Kiprakis Aristides. *Smart contracts in Energy Systems: A Systematic Review of Fundamental Approaches and Implementations*. Renewable and Sustainable Energy Reviews n°158, 1 avril 2022, p. 112013.

Kumar, S., A. Murugan, Muruganantham B., et Sriman B. *IoT-smart contracts in data trusted exchange supplied chain based on block chain*. International Journal of Electrical and Computer Engineering n°10, 1 février 2020, p. 438.

Neidhardt, Nils, Köhler Carsten et Nüttgens Markus. *Cloud Service Billing and Service Level Agreement Monitoring based on Blockchain*. Entwicklungsmethoden für Informationssysteme und deren Anwendung: Fachtagung, 2018.

Neuburger Jeffrey D., Wai L. Choy et Milewski Kevin P. *Smart contracts: Best Practices*, Thomson Reuters - Practical Law, 2019, p.11 à 19.

Norton Rose Fulbright. *Smart contracts: coding the fine print*, mars 2016.

Oliveira, Ludmila, Simoyama Felipe, Grigg Ian, et Luiz Bueno Ricardo. *Triple entry ledgers with blockchain for auditing*. International Journal of Auditing Technology n°3, 1 janvier 2017, p. 163.

Renu, Sathya A., et Barnali Gupta Banik. *Implementation of a Secure Ride-Sharing DApp Using Smart contracts on Ethereum Blockchain*. International Journal of Safety and Security Engineering 11, n° 2, 30 avril 2021, p. 167 à 173.

Reshi Iraq, Khan Muneeb, Shafi Sadaf, Sholla Sahil, Assad Assif, et Shafi Huzaib. *AI-Powered Smart contracts: The Dawn of Web 4.0*, 6 mars 2023.

Scheid J., Rodrigues B. B., Granville L. Z. et Stiller B., *Enabling Dynamic SLA Compensation Using Blockchain-based Smart contracts*, 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2019, p. 53-61.

Six Nicolas, Negri Ribalta Claudia, Herbaut Nicolas et Salinesi Camille. *A Blockchain-Based Pattern for Confidential and Pseudo-Anonymous Contract Enforcement*. 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2020, p. 1965 à 1970

Small Dave. *Un langage de programmation est censé être une façon conventionnelle de donner des instructions à un ordinateur, et doit pouvoir être écrit et relu par des personnes différentes*. ST Magazine, novembre 1992.

Uriarte, Brundo Rafael, Zhou Huan, Kritikos Kyriakos, Shi Zeshun, Zhao Zhiming, et De Nicola Rocco. *Distributed service-level agreement management with smart contracts and blockchain*. Concurrency and Computation: Practice and Experience 33, 28 avril 2020.

WEBOGRAPHIE

I – Sources juridiques

A - Billets de blog

B – Sites web

II – Sources techniques

A - Billets de blog

B – Sites web

I – Sources juridiques

A – Billets de blog

ArtificialLawyer. *Machine-Readable Contracts – A New Paradigm For Legal Documentation*. Artificial Lawyer (blog), 28 août 2019. <https://www.artificiallawyer.com/2019/08/28/machine-readable-contracts-a-new-paradigm-for-legal-documentation/>.

Bem Anthony. *Responsabilité des concepteurs et développeurs de sites internet : l'importance du "PV de recette"*. Legavox (blog), 30 octobre 2013. <http://www.legavox.fr/blog/maitre-anthony-bem/responsabilite-concepteurs-developpeurs-sites-internet-12867.htm>.

Cambournac Emilie. *Clause pénale et vente immobilière, un gage de loyauté dans la conduite des pourparlers*. Village de la Justice (blog), 20 février 2015. <https://www.village-justice.com/articles/clause-penale-matiere-vente-immobiliere-gage-loyaute-dans-conduite-des,33814.html>.

Collet Michel et Bordenave Alexandre. *Prêteurs et emprunteurs : la fiducie, c'est maintenant !* CMS Francis LEFEBVRE (blog), 22 octobre 2020.

Clause. *REALLY Smart (and Legal!) Contracts*. Clause (blog), 28 mars 2018. <https://medium.com/clause-blog/really-smart-and-legal-contracts-a77fcd1d0d10>.

Devigny Emmanuelle. *Médiation, Conciliation, Arbitrage, Négociation : C'est quoi les MARC ?* Officiel de la Médiation Professionnelle et de la Profession de Médiateur (blog), 16 janvier 2017. <https://www.officieldelamediation.fr/2017/01/16/mediation-conciliation-arbitrage-negociation-a-vos-marcs/>.

Diallo Abdoulaye.

Smart contract d'une clause pénale. Village de la Justice (blog), 19 mars 2021. <https://www.village-justice.com/articles/smart-contract-une-clause-penale,30899.html>.

Smart contract d'une garantie autonome au sens de l'article 2321 du code civil. Village de la Justice (blog), 18 novembre 2015. <https://www.village-justice.com/articles/smart-contract->

[une-garantie-autonome-sens-article-2321-code-civil,29924.html](https://www.village-justice.com/articles/smart-sas-une-societe-par-actions-simplifiee-dans-blockchain,33192.html).

Smart SAS : une Société par Actions Simplifiée dans la Blockchain. Village de la Justice (blog), 13 décembre 2019. <https://www.village-justice.com/articles/smart-sas-une-societe-par-actions-simplifiee-dans-blockchain,33192.html>.

TokenIP : Une Proposition de Tokenisation Des Droits d’auteurs En Droit Français. Medium (blog), 30 décembre 2020. <https://abdoulaye77124.medium.com/tokenip-une-proposition-de-tokenisation-des-droits-dauteurs-en-droit-fran%C3%A7ais-43624b402ecc>.

Fusions-Acquisitions et Smart contracts. Medium (blog), 9 mars 2022. <https://abdoulaye77124.medium.com/fusions-acquisitions-et-smart-contract-71e7985cf84b>.

Fèvre Aliénor. *La détermination de la loi applicable dans les contrats*. CMS Francis LEFEBVRE (blog). Consulté le 12 juin 2023. <https://cms.law/fr/fra/news-information/la-determination-de-la-loi-applicable-dans-les-contrats>.

Legalis | L’actualité du droit des nouvelles technologies. *Logiciel spécifique : manquement à l’obligation de résultat mais pas de résolution du contrat*, 2 février 2021. <https://www.legalis.net/actualite/logiciel-specifique-manquement-a-lobligation-de-resultat-mais-pas-de-resolution-du-contrat/>.

Livia Chartrain. *Quelle blockchain a choisi MonJuridique ?* Consulté le 5 juillet 2023. <https://monjuridique.infogreffe.fr/blog/blockchain-monjuridique>.

Netter Emmanuel. *II - L’idéal de rigueur dans l’exécution des contrats - Droit et numérique*. Consulté le 28 mars 2023. <https://enetter.fr/le-contrat/section-1-droit-commun-des-contrats/ii-lideal-de-rigueur-dans-lexecution-des-contrats/>.

Ortlepp Clemens. *Molecule’s Biopharma IP-NFTs - A Technical Description*, 6 août 2021. <https://www.molecule.to/blog/molecules-biopharma-ip-nfts-a-technical-description>.

Quinn Emanuel Urquhart. *Client Alert: Code is Law*. Consulté le 25 avril 2023. <https://www.quinnemanuel.com/the-firm/publications/code-is-law/>.

Rouah Arnold. *La fiducie au service du M&A*. Village de la Justice (blog), 21 novembre 2015. <https://www.village-justice.com/articles/fiducie-service,40749.html>.

Verbiest Thibault. *Le Service Level Agreement dans les contrats informatiques*. Droit & Technologies (blog), 11 novembre 2003. <https://www.droit-technologie.org/actualites/le-service-level-agreement-dans-les-contrats-informatiques/>.

B - Sites web

AMF. *Le mandat de gestion*. Consulté le 11 avril 2023. <https://www.amf-france.org/fr/espace-epargnants/comprendre-les-produits-financiers/supports-dinvestissement/mandat-de-gestion>.

Banque centrale européenne. *Un euro numérique*, 8 novembre 2022. https://www.ecb.europa.eu/paym/digital_euro/html/index.fr.html.

Banque populaire. *Modèle de convention de compte courant la Banque Populaire Méditerranée*, 22 juin 2021. https://web.archive.org/web/20210622101835/https://www.mediterranee.banquepopulaire.fr/portailinternet/Editorial/Lists/DocEditoList/Convention_de_compte_courant-BPMED-01-12-2020.pdf.

Bissegger Mark. *Smart contract Applications in M&A: Earn-Outs*. Deal Law Wire, 22 novembre 2017. <https://www.deallawwire.com/2017/11/22/smart-contract-applications-in-ma-earn-outs/>.

Bofip. *BOFiP BOI-BIC-CHAMP-70-20-60 - 12/09/2012*, 12 septembre 2012. <https://bofip.impots.gouv.fr/bofip/3606-PGP.html/identifiant=BOI-BIC-CHAMP-70-20-60-20120912>.

CNIL.

Finalité d'un traitement. Consulté le 5 juin 2023. <https://www.cnil.fr/fr/definition/finalite-dun-traitement>.

Force brute (attaque informatique). Consulté le 30 septembre 2021. <https://www.cnil.fr/fr/definition/force-brute-attaque-informatique>.

Sous-traitants : la réutilisation de données confiées par un responsable de traitement | CNIL.
Consulté le 31 mai 2023. <https://www.cnil.fr/fr/sous-traitants-la-reutilisation-de-donnees-confiees-par-un-responsable-de-traitement>.

Dalloz.

Garantie autonome - Fiches d'orientation - juillet 2022. Consulté le 28 mars 2023.
<https://www.dalloz.fr/documentation/Document?id=DZ%2FOASIS%2F001645>

Clause d'agrément - Fiches d'orientation - juillet 2022. Consulté le 14 avril 2023.
<https://www.dalloz.fr/documentation/Document?id=DZ%2FOASIS%2F000186>

Clause de garantie de passif - Fiches d'orientation - juillet 2022. Consulté le 16 avril 2023.
<https://www.dalloz.fr/documentation/Document?id=DZ%2FOASIS%2F001762>

Statuts de société et actes annexes - Fiches d'orientation - août 2022. Consulté le 13 avril 2023.
<https://www.dalloz.fr/documentation/Document?id=DZ%2FOASIS%2F000954>

Grigg Ian.

The Ricardian Contract, 2004.
https://iang.org/papers/ricardian_contract.html#ref_11.

On the intersection of Ricardian and Smart contracts, février 2015.
https://iang.org/papers/intersection_ricardian_smart.html.

International Chamber of Commerce (ICC). *When a Non-Bank Issues a Letter of Credit*.
Consulté le 28 mars 2023. <https://2go.iccwbo.org/when-a-non-bank-issues-a-letter-of-credit.html>.

Law Insider. *Force Majeure | Sample Clauses*. Consulté le 13 juin 2023.
<https://www.lawinsider.com/fr/clause/force-majeure>.

Lessig Lawrence. *Code Is Law*. Harvard Magazine, 1 janvier 2000.
<https://www.harvardmagazine.com/2000/01/code-is-law-html>.

Mekki Mustapha. *Blockchain : l'exemple des smart contractss*. Consulté le 20 mai 2023.
<https://mustaphamekki.openum.ca/publications/1314/>.

Open Source Initiative. *The MIT License*, 31 octobre 2006.
<https://opensource.org/license/mit/>.

OpenLaw. *Introducing OpenLaw*. Medium, 22 avril 2020.
<https://media.consensys.net/introducing-openlaw-7a2ea410138b>.

Poloni Flore et Roujou de Boubée Thibaud. *[Avis d'expert] Quatre raisons de privilégier l'arbitrage dans les litiges sur de nouvelles technologies*, 6 juin 2021.
<https://www.usinenouvelle.com/editorial/avis-d-expert-quatre-raisons-de-privilegier-l-arbitrage-dans-les-litiges-sur-de-nouvelles-technologies.N1099754>.

Sutour Jérôme. *Etat des lieux sur la réglementation des « nouveaux actifs »*. Consulté le 17 février 2023. <https://cms.law/fr/fra/news-information/etat-des-lieux-sur-la-reglementation-des-nouveaux-actifs>.

Szabo Nick. *A Formal Language for Analyzing Contracts* | Satoshi Nakamoto Institute, 2002.
<https://nakamotoinstitute.org/contract-language/>.

Twitter.

_gabrielShapir0 sur Twitter, 19 juillet 2018.
https://twitter.com/lex_node/status/1019819161298456577.

_gabrielShapir0 sur Twitter, 12 août 2018.
https://twitter.com/lex_node/status/1028727472311873537.

michael rice, legal engineer sur Twitter, 2 novembre 2018.

i) <https://twitter.com/michaelriceLE/status/1058477183872581637>.

II – Sources techniques

A – Billets de blog

Adejumo Oluwapelumi. *Crypto Investments Fund Founder Says Polygon Is “Highly Insecure & Centralized”*. CryptoSlate (blog), 16 août 2022. <https://cryptoslate.com/crypto-investments-fund-founder-says-polygon-is-highly-insecure-centralized/>.

American Trust Escrow. *What’s the Difference Between an Escrow Account and a Trust Account ?* American Trust Escrow (blog), 20 avril 2018. <https://americantrustescrow.com/2018/04/20/whats-the-difference-between-an-escrow-account-and-a-trust-account/>.

APP - Agence pour la Protection des Programmes. *Entiercement - Gérer ses entiercements avec l’APP*. Consulté le 27 mars 2023. <https://www.app.asso.fr/nos-solutions/escrow-agreement>.

Arluck Jacob. *Amending Tezos*. Tezos (blog), 13 mai 2020. <https://medium.com/tezos/amending-tezos-b77949d97e1e>.

Assas Legal Innovation. *Le petit guide de la Blockchain*. Assas Legal Innovation (blog), 16 juillet 2018. <https://assaslegalinnovation.com/2018/07/16/le-petit-guide-de-la-blockchain/>.

Bellanca Chloe. *Qu’est-ce que le Proof of Authority ?* Coinhouse (blog), 28 juin 2019. <https://www.coinhouse.com/fr/academie/blockchain/proof-of-authority/>.

Bispo Nuno. *The Benefits of Decentralized Storage vs Cloud Storage*. Geek Culture (blog), 6 janvier 2022. <https://medium.com/geekculture/the-benefits-of-decentralized-storage-vs-cloud-storage-f5f01592ed9d>.

Born Cody. *Tokenized Reputation*. The Capital (blog), 26 janvier 2019. <https://medium.com/the-capital/tokenized-reputation-dee463fbc631>.

Brady Dale. *Bitcoin Will Change Money Like the Internet Changed Video*. Observer (blog), 17 janvier 2017. <https://observer.com/2017/01/bitcoin-lightning-network-andreas-antonopoulos/>.

Buterin Vitalik.

Visions, Part 1: The Value of Blockchain Technology. Ethereum Foundation Blog (blog), 13 avril 2015. <https://blog.ethereum.org/2015/04/13/visions-part-1-the-value-of-blockchain-technology>.

Why sharding is great: demystifying the technical properties, 7 avril 2021. <https://vitalik.ca/general/2021/04/07/sharding.html>.

EDIpourtous. *Ce Qu'est EDI (Échange de Données Informatisé)?*, 20 septembre 2019. <https://www.edipourtous.fr/ce-qu-est-l-edi/>.

Chainlink.

What Are Smart contracts and How Can They Revolutionize the Future, 7 juin 2019. <https://blog.chain.link/the-power-of-smart-contracts-what-they-are-and-how-they-can-revolutionize-the-future/>.

Trust to Derivatives Using Chainlink DeFi Smart contracts, 11 octobre 2019. <https://blog.chain.link/solving-deep-seated-trust-problems-in-derivatives-using-chainlink-enabled-smart-contracts/>.

Challapalli Kiran. *Trade Finance Workflow Automation Using AI*. IBM Digital Transformation Blog (blog), 5 mai 2021. <https://www.ibm.com/blogs/digital-transformation/in-en/blog/trade-finance-workflow-automation-using-ai/>.

Clause. *Clause Joins Hyperledger*. Clause (blog), 22 mai 2017. <https://medium.com/clause-blog/clause-joins-hyperledger-38a10f8f3ea5>.

Clemens Ortlepp. *Molecule's Biopharma IP-NFTs - A Technical Description*, 6 août 2021. <https://www.molecule.to/blog/molecules-biopharma-ip-nfts-a-technical-description>.

Cryptonio.tez. *Rollups on Tezos [Part I]*. Medium, 3 juin 2023. <https://news.tezoscommons.org/rollups-on-tezos-part-i-cdd7b70d53da>.

Cvllr Jean. *Solidity Tutorial: All About Imports*. Medium, 22 mars 2022. <https://betterprogramming.pub/solidity-tutorial-all-about-imports-c65110e41f3a>.

Fábio José. *Building a Smart contract to Sell Goods*. Coinmonks (blog), 26 juin 2022. <https://medium.com/coinmonks/build-a-smart-contract-to-sell-goods-6cf73609d25>.

Falkon Samuel. *The Story of the DAO — Its History and Consequences*. The Startup (blog), 12 août 2018. <https://medium.com/swlh/the-story-of-the-dao-its-history-and-consequences-71e6a8a551ee>.

Garg Priyeshu. *Chainlink, Band Protocol, API3, and Umbrella Network: Exploring the Differences Between Oracles*. Umbrella Network (blog), 1 février 2021. <https://medium.com/umbrella-network/chainlink-band-protocol-api3-and-umbrella-network-exploring-the-differences-between-oracles-9477d975e142>.

Gluchowski Alex. *Optimistic vs. ZK Rollup: Deep Dive*. Medium, 7 avril 2021. <https://blog.matter-labs.io/optimistic-vs-zk-rollup-deep-dive-ea141e71e075>.

Guitard Grégory. *Tezos (XTZ) : le mirage du fork DUNE*. Journal du Coin, 20 juin 2019. <https://journalducoin.com/altcoins/actualites-altcoins/tezos-mirage-fork-dune/>.

Hosking Ben « The Hosk ». *You Cannot Create Software Without Bugs, Problems and Mistakes*. Geek Culture (blog), 18 septembre 2021. <https://medium.com/geekculture/you-cannot-create-software-without-bugs-problems-and-mistakes-615b6540bc3f>.

Hugh. B. *Hive vs Steem - Le fork rebelle dépasse de maître vendu*. CryptoActu (blog), 4 juin 2020. <https://cryptoactu.com/hive-vs-steem-fork-rebelle-depasse-maitre-vendu/>.

Offchain Labs. *Hello, Stylus*. Medium (blog), 7 février 2023. <https://offchain.medium.com/hello-stylus-6b18fecc3a22>.

Leifke Robert. *Perpetual Options for DeFi*. Numoen (blog), 20 septembre 2022. <https://medium.com/numoen/perpetual-options-for-defi-821351c0a24f>.

Lipton Alex et Stuart Levi. *An Introduction to Smart contracts and Their Potential and Inherent Limitations*. The Harvard Law School Forum on Corporate Governance (blog), 26 mai 2018.

<https://corpgov.law.harvard.edu/2018/05/26/an-introduction-to-smart-contracts-and-their-potential-and-inherent-limitations/>.

Livia Chartrain. *Quelle blockchain a choisi MonJuridique ?*. Consulté le 5 juillet 2023. <https://monjuridique.infogreffe.fr/blog/blockchain-monjuridique>.

Mark Simon. *MetaMask and Infura: The Centralised Infrastructure Behind Crypto*. HelloCrypto (blog), 19 septembre 2022. <https://hellocrypto.com/article/metamask-and-infura-the-centralised-infrastructure-behind-crypto/>.

Michael @ CryptoEQ. *Is Avalanche (AVAX) Governed More by On-Chain or Off-Chain?*, 1 juin 2022. <https://www.publish0x.com/cryptoeq/is-avalanche-avax-governed-more-by-on-chain-or-off-chain-xnnedee>.

Moore Galen. *The Social Layer Is Ironically Key to Bitcoin's Security*. TechCrunch (blog), 19 janvier 2019. <https://techcrunch.com/2019/01/19/bitcoin-social-layer/>.

Morais João Paulo. *Learn Solidity Lesson 19. Timestamp*. Coinmonks (blog), 9 août 2022. <https://medium.com/coinmonks/learn-solidity-lesson-19-timestamp-7ba91290c245>.

OpenLaw. *The Smart contract Stack*. Medium (blog), 24 septembre 2019. <https://medium.com/@OpenLawOfficial/the-smart-contract-stack-5566ea368a74>.

Papacharissiou Harry. *Build a Parametric Insurance Smart contract With Chainlink*. Chainlink Blog (blog), 15 décembre 2020. <https://blog.chain.link/parametric-insurance-smart-contract/>.

Quinn Emanuel Urquhart. *Client Alert: Code is Law*. Consulté le 25 avril 2023. <https://www.quinnemanuel.com/the-firm/publications/code-is-law/>.

S. Pranesh. *Using the UUPS Proxy Pattern to Upgrade Smart contracts*. LogRocket Blog (blog), 24 février 2022. <http://blog.logrocket.com/using-uups-proxy-pattern-upgrade-smart-contracts/>.

Seq. *What Sets Avalanche Apart From Other Blockchains ?* Medium (blog), 17 octobre 2021. <https://cryptoseq.medium.com/what-sets-avalanche-apart-from-other-blockchains-3c5f4a4c0889>.

Sharer Happy. *Exploring How Many Ethereum Nodes Are There: An Analysis of the Ethereum Network - The Enlightened Mindset*, 18 janvier 2023. <https://www.lihpao.com/how-many-ethereum-nodes-are-there/>.

Staff Block Telegraph. *General-Purpose vs App-Specific L1s - Block Telegraph*, 7 septembre 2022. <https://blocktelegraph.io/general-purpose-vs-app-specific-l1s/>.

Stankovic Stefan. *Who Is Nick Szabo, The Mysterious Blockchain Titan*. Unblock.Net (blog), 11 janvier 2018. <https://unblock.net/nick-szabo/>.

Starkware. *Tech Stack*. Consulté le 3 juillet 2023. <https://starkware.co/tech-stack/>.

Syscoin Corey Crypto. *Solana: The 65,000 TPS Blockchain: Warp Speed*. Medium (blog), 9 juillet 2020. <https://coreycrypto.medium.com/solana-the-65-000-tps-blockchain-warp-speed-b34d3ebb85c>.

Tan Cindy. *NFT Landlords Are Making Big Bucks from Renting Out Blockchain Game NFTs*. NFTgators (blog), 25 janvier 2022. <https://www.nftgators.com/nft-landlords-are-making-big-bucks-from-renting-out-blockchain-game-nfts/>.

Throuvalas Andrew. *Is Solana Really Decentralized? A Validator Health Report*. CryptoPotato (blog), 16 août 2022. <https://cryptopotato.com/is-solana-really-decentralized-a-validator-health-report/>.

TokenInsight. *Optimism vs. Arbitrum — A Complete Comparison*. Medium (blog), 4 juillet 2022. <https://tokeninsight.medium.com/optimism-vs-arbitrum-a-complete-comparison-f504f727e4df>.

Toon Mark. *Polygon Hit by 157-Block 'Reorg' despite Hard-Fork to Reduce Reorgs*. Protos (blog), 24 février 2023. <https://protos.com/polygon-hit-by-157-block-reorg-despite-hard-fork-to-reduce-reorgs/>.

Tsankov Alexander. *The "Oracle Problem" Isn't a Problem, and Why Smart contracts Makes Insurance Better for Everyone*. Medium (blog), 21 juin 2018. <https://antsankov.medium.com/the-oracle-problem-isnt-a-problem-and-why-smart-contracts-makes-insurance-better-for-everyone-8c979f09851c>.

Wilkof Neil. *Tokenization of intellectual property for IP rights management*. The IPKat (blog). Consulté le 23 février 2023. <https://ipkitten.blogspot.com/2022/01/tokenization-of-intellectual-property.html>.

Young Julie. *Understanding Fully Funded Documentary Letters of Credit (FFDLC)*. Investopedia (blog), 27 janvier 2023. <https://www.investopedia.com/terms/f/ffdlc.asp>.

Yousif Ahmed. *Blockchain: The Protocol of Value*. BSV Blockchain (blog), 1 juin 2022. <https://bsvblockchain.org/news/blockchain-the-protocol-of-value/>.

Zamfir Vlad. *Against Szabo's Law, For A New Crypto Legal System*. Crypto Law Review (blog), 29 janvier 2019. <https://medium.com/cryptolawreview/against-szabos-law-for-a-new-crypto-legal-system-d00d0f3d3827>.

B - Logiciels et sites web

Academy. *Maker : Le Pionnier de La Finance Décentralisée*, 15 avril 2021. <https://academy.youngplatform.com/fr/cryptomonnaies/maker-le-pionnier-de-la-finance-decentralisee/>.

Accord Project.

Documentation. Consulté le 27 avril 2023. <https://docs.accordproject.org/index.html>.

Example: Late Delivery Clause. Consulté le 16 avril 2023. <https://docs.accordproject.org/index.html>.

Alchemy.

Learn Solidity: What Are Events ?. Consulté le 8 juillet 2023. <https://www.alchemy.com/overviews/solidity-events>.

Learn Solidity: What Is Selfdestruct ?. Consulté le 9 juillet 2023. <https://www.alchemy.com/overviews/selfdestruct-solidity>.

Solidity Data Types: Signed (Int) and Unsigned Integers (Uint). Consulté le 8 juillet 2023.
<https://www.alchemy.com/overviews/solidity-uint>.

Solidity vs. Rust: Everything You Need to Know. Consulté le 2 juillet 2023.
<https://www.alchemy.com/overviews/solidity-vs-rust>.

What Is the Solidity Contract Interface ?. Consulté le 8 juillet 2023.
<https://www.alchemy.com/overviews/solidity-interface>.

AMF. *Qu'est-ce qu'une Initial Coin Offering (ICO) ?* Consulté le 17 février 2023.
<https://www.amf-france.org/fr/quest-ce-quune-initial-coin-offering-ico>.

Aragon. *Terms and Conditions*. Consulté le 18 mai 2023. <https://aragon.org/terms-and-conditions>.

AXA. *AXA se lance sur la Blockchain avec fizzy*. Consulté le 28 février 2023.
<https://www.axa.com/fr/actualites/axa-se-lance-sur-la-blockchain-avec-fizzy>.

Avalanche.

Avalanche Consensus. Consulté le 30 juin 2023.
<https://docs.avax.network/learn/avalanche/avalanche-consensus>.

What Are the Differences between the X, P, and C-Chains ?. Consulté le 30 juin 2023.
<https://support.avax.network/en/articles/6077308-what-are-the-differences-between-the-x-p-and-c-chains>.

What Is a Subnet ?. Consulté le 30 juin 2023.
<https://docs.avax.network/learn/avalanche/subnets-overview>.

What Is Transactional Finality ?. Consulté le 16 mai 2023.
<https://support.avax.network/en/articles/5325234-what-is-transactional-finality>.

Beams Andrew, Gu Catherine, Raghuraman Srini, Mohsen Minaei, et Ranjit Kumaresan. *Visa Crypto Thought Leadership – Auto Payments*. Consulté le 2 avril 2023. <https://visa-signature.com/solutions/crypto/auto-payments-for-self-custodial-wallets.html>.

Become Ethereum Blockchain Developer. *Remix vs Truffle vs Hardhat vs Foundry*. Consulté le 6 juillet 2023. <https://ethereum-blockchain-developer.com/124-remix-vs-truffle-vs-hardhat-vs-foundry/00-overview/>.

Bennett Mark. *PhD Research in Computing, IT & Computer Science*. www.FindAPhD.com, 2 février 2021. <https://www.findaphd.com/guides/computing-phds>.

Bitcoin.com. *How Does Governance Work in Ethereum*. Consulté le 29 juin 2023. <https://www.bitcoin.com/get-started/how-does-governance-work-in-ethereum/>.

Bird & Bird. *Private Blockchains*. Briefing Note, <https://www.twobirds.com/~media/pdfs/in-focus/blockchain/private-blockchain-briefing-note.pdf>.

BitPay Blog. *Custodial vs Non-Custodial Wallet - What's the Difference ?*, 24 mai 2023. <https://bitpay.com/blog/non-custodial-wallets-vs-custodial-wallets/>.

BlockBar. *Glenfiddich Sera Le Premier Partenaire à Commercialiser Un Whisky Rare via NFT Avec BlockBar, La Première Plateforme NFT de Vente Directe Aux Consommateurs Pour Les Vins et Spiritueux*. Consulté le 22 février 2023. <https://www.prnewswire.com/news-releases/glenfiddich-sera-le-premier-partenaire-a-commercialiser-un-whisky-rare-via-nft-avec-blockbar-la-premiere-plateforme-nft-de-vente-directe-aux-consommateurs-pour-les-vins-et-spiritueux-841290934.html>.

Blockwatch. *Bakers Tezos sur TzStats*. Consulté le 30 juin 2023. <https://tzstats.com/bakers>.

Brenda Mary. *Polygon Became the Most Widely Used Blockchain in 2023*, 18 mai 2023. <https://crypto.news/polygon-became-the-most-widely-used-blockchain-in-2023/>.

Bussler Frederik. *Why NFTs Are The Future of Invoicing*, 22 août 2022. <https://hackernoon.com/why-nfts-are-the-future-of-invoicing>.

Buterin Vitalik.

Ethereum Whitepaper. ethereum.org. Consulté le 12 octobre 2021. <https://ethereum.org>.

Why we need wide adoption of social recovery wallets, 11 janvier 2021.

<https://vitalik.ca/general/2021/01/11/recovery.html>.

Why sharding is great: demystifying the technical properties. Consulté le 8 décembre 2021.

<https://vitalik.ca/general/2021/04/07/sharding.html>.

Capot Christopher et Lammerding Eric. *LexisNexis Launches Flight Status Data Tracking Using Chainlink Node, Enabling Smart contract-Based Parametric Insurance Products*, 29 juin 2022.

<https://www.lexisnexis.com/community/pressroom/b/news/posts/lexisnexis-launches-flight-status-data-tracking-using-chainlink-node-enabling-smart-contract-based-parametric-insurance-products>.

ChainLink.

Blockchain Scalability Approaches. Consulté le 28 juin 2023. <https://chain.link/education-hub/blockchain-scalability>.

Solidity vs. Vyper: Which Smart contract Language Is Right for Me ?, Chainlink Blog, 17 octobre 2022. <https://blog.chain.link/solidity-vs-vyper/>.

What Is Chainlink ? A Beginner's Guide. Chainlink Blog, 25 janvier 2021. <https://blog.chain.link/what-is-chainlink/>.

What Is the Blockchain Oracle Problem? Chainlink. Chainlink Blog, 27 août 2020. <https://blog.chain.link/what-is-the-blockchain-oracle-problem/>.

Make a GET Request. Consulté le 10 juillet 2023. <https://docs.chain.link/any-api/get-request/introduction>.

What Is an Oracle in Blockchain ?. Consulté le 29 mai 2023. <https://chain.link/education/blockchain-oracles>.

Chawla Vishal. *NFT Gamers Are Clogging Up Polygon*. Crypto Briefing, 5 janvier 2022. <https://cryptobriefing.com/nft-gamers-are-clogging-up-polygon/>.

CNIL. Interface de programmation d'application (API). Consulté le 27 février 2023.

<https://www.cnil.fr/fr/definition/interface-de-programmation-dapplication-api>

C. May, Timothy. *The Crypto Anarchist Manifesto*, 22 novembre 1992.

<https://www.activism.net/cypherpunk/crypto-anarchy.html>.

Codekeeper. *Software Escrow: Ready for Quick Recovery*. Consulté le 19 avril 2023.

<https://codekeeper.co/software-escrow.html>.

Coinbase Help. *What Is a Transaction Hash/Hash ID ?*, Consulté le 18 mai 2023.

<https://help.coinbase.com/en/coinbase/getting-started/crypto-education/what-is-a-transaction-hash-hash-id>.

Cointelegraph. *What Is the Network Effect ?*, 19 février 2023. <https://cointelegraph.com/news/what-is-the-network-effect>.

Collins Patrick. *How to Make an NFT and Render It on the OpenSea Marketplace*. freeCodeCamp.org, 1 avril 2021. <https://www.freecodecamp.org/news/how-to-make-an-nft-and-render-on-opensea-marketplace/>.

ComputerHope. *What Is Compilation ?*. Consulté le 12 juillet 2023.

<https://www.computerhope.com/jargon/c/compilat.htm>.

Cosmos.

Cosmos Network - Internet of Blockchains. Consulté le 12 juillet 2023.

<https://cosmos.network>.

Application-Specific Blockchains. Consulté le 3 juillet 2023.

<https://docs.cosmos.network/main/intro/why-app-specific>.

Cryptoast. *Aave (AAVE), le protocole de prêt de cryptomonnaies non-custodial*, 7 novembre 2020.

<https://cryptoast.fr/aave-protocole-pret-cryptomonnaies/>.

Cuny Delphine. *Retard d'avion : Axa lance une assurance automatique sur la Blockchain*. La Tribune, 14 septembre 2017. <https://www.latribune.fr/entreprises-finance/banques-finance/retard-d-avion-axa-lance-une-assurance-automatique-sur-la-blockchain-750202.html>.

Curry Benjamin et Napoletano E. *What Is Proof of Stake? How Does It Work?* – Forbes Advisor. Forbes, 16 février 2023. <https://www.forbes.com/advisor/investing/cryptocurrency/proof-of-stake/>.

Démarches.Intérieur. *Qu'est-ce qu'un marché public ?*,
<https://www.demarches.interieur.gouv.fr/professionnels/qu-est-ce-qu-un-marche-public>.

Consulté le 17 septembre 2023.

DEV Community. *Indexing Smart contract Data Using The Graph Protocol*, 27 juillet 2022.
<https://dev.to/jamiescript/indexing-smart-contract-data-using-the-graph-protocol-4h95>.

Distributed.

How Tokenization Is Putting Real-World Assets on Blockchains. Consulté le 23 février 2023.
<https://www.nasdaq.com/articles/how-tokenization-putting-real-world-assets-blockchains-2017-03-30>.

How Tokenization Is Putting Real-World Assets on Blockchains. Consulté le 23 février 2023.
<https://www.nasdaq.com/articles/how-tokenization-putting-real-world-assets-blockchains-2017-03-30>.

dYdX. *Announcing dYdX Chain*, 22 juin 2022. <https://dydx.exchange/blog/dydx-chain>.

Equisafe. *CGU Equisafe*. Consulté le 18 mai 2023. <https://www.equisafe.io/cgu-equisafe>.

Ethereum Improvement Proposals.

EIP-20: Token Standard. Consulté le 14 avril 2022. <https://eips.ethereum.org/EIPS/eip-20>.

EIP-721: Non-Fungible Token Standard. Consulté le 14 avril 2022.
<https://eips.ethereum.org/EIPS/eip-721>.

EIP-4844: Shard Blob Transactions. Consulté le 12 juillet 2023.
<https://eips.ethereum.org/EIPS/eip-4844>.

Ethereum.org.

Gas and Fees. Consulté le 29 juin 2023. <https://ethereum.org>.

Introduction to Dapps. Consulté le 8 décembre 2021. <https://ethereum.org>.

JavaScript API Libraries. Consulté le 7 juillet 2023. <https://ethereum.org>.

Nodes and Clients. Consulté le 7 juillet 2023. <https://ethereum.org>.

Optimistic Rollups. Consulté le 3 juillet 2023. <https://ethereum.org>.

Proof-of-Stake (PoS). Consulté le 28 juin 2023. <https://ethereum.org>.

Scaling. Consulté le 3 juillet 2023. <https://ethereum.org>.

Etherscan Beacon Chain (Phase 0) Ethereum 2.0 Explorer. *Statistics - Validators | Mainnet Beacon Chain (Phase 0) Ethereum 2.0 Explorer*. Consulté le 28 juin 2023. <https://beaconscan.com//stat/validator>.

Etherscan.io. *Verify & Publish Contract Source Code*. Ethereum (ETH) Blockchain Explorer. Consulté le 7 juillet 2023. <http://etherscan.io/verifyContract>.

Evans Steve. *Parsyl Sensor Driven Parametric Insurance Pays-out in Less than 8 Hours - Artemis.Bm. - The Catastrophe Bond, Insurance Linked Securities & Investment, Reinsurance Capital, Alternative Risk Transfer and Weather Risk Management site*, 29 juin 2021. <https://www.artemis.bm/news/parsyl-sensor-driven-parametric-insurance-pays-out-in-less-than-8-hours/>.

Exaion. *Exaion, EDF Group Subsidiary, Becomes a Tezos Baker*, 21 juin 2022. <https://exaion.edf.fr/en/exaion/our-news/exaion-edf-group-subsidiary-becomes-a-tezos-baker>.

freeCodeCamp.org. *What Are Solidity Modifiers ? Explained with Examples*, 6 janvier 2023. <https://www.freecodecamp.org/news/what-are-solidity-modifiers/>.

Froment Etienne. *Quelle différence entre un token et un coin pour les cryptomonnaies ?* 20Minutes, 14 avril 2022. <https://www.20minutes.fr/high-tech/3179779-20211123-token-et-coin-comprendre-les-differences>.

G, Thomas. *Comment fonctionnent les phrases mnémoniques (seed) ?*, Journal du Coin, 13 avril 2018. <https://journalducoin.com/actualites/phrases-mnemoniques/>.

_g4brielShapir0. « SCoDA – Simple Code Deference Agreement », 25 avril 2023. <https://github.com/lex-node/SCoDA-Simple-Code-Deference-Agreement->.

GeeksforGeeks.

Solidity - Mappings, 10 juillet 2020. <https://www.geeksforgeeks.org/solidity-mappings/>.

Solidity - Constructors, 10 juillet 2020. <https://www.geeksforgeeks.org/solidity-constructors/>.

Solidity - Enums and Structs, 10 juillet 2020. <https://www.geeksforgeeks.org/solidity-enums-and-structs/>.

Gemini.

Crypto Wallets: Hot vs. Cold Wallets. Consulté le 6 juillet 2023. <https://www.gemini.com/cryptopedia/crypto-wallets-hot-cold>.

What Is a Block Explorer? BTC Block Explorers, Etc. Consulté le 7 juillet 2023. <https://www.gemini.com/cryptopedia/what-is-a-block-explorer-btc-bch-eth-ltc>.

Gilbert Aleksandar. *Solana To Focus On Stability In 2023 After Repeated Outages*. The Defiant, 2 mars 2023. <https://thedefiant.io/solana-focus-stability-2023>.

Gnosis.

Who Can Govern Gnosis Chain, 18 mars 2022. <http://forum.gnosis.io/t/who-can-govern-gnosis-chain/4133>.

Gnosis Chain. Consulté le 30 juin 2023. <https://www.gnosis.io/>.

H. Renaud. *Ethereum : le hard fork Istanbul, qu'est-ce que ça change ?*, Journal du Coin, 7 décembre 2019. <https://journalducoin.com/ethereum/ethereum-hard-fork-istanbul-definition/>.

HackMD. *Decentralized zk-Rollup*. Consulté le 12 juillet 2023. <https://hackmd.io/@yezhang/SkmyXzWMY#Decentralized-zk-Rollup>.

Haig Samuel. *Gnosis Executes Its Own Merge in Shift to PoS in Boost for Staking*. The Defiant, 8 décembre 2022. <https://thedefiant.io/gnosis-to-execute-its-own-merge-in-shift-to-pos>.

Harsh Kumar. *Demystified: The Difference Between Crypto Coins And Crypto Tokens*, 21 mai 2022. <https://www.outlookindia.com/business/demystified-the-difference-between-crypto-coins-and-crypto-tokens-read-here-for-details-news-197683>.

Hayes Frank. *The Story So Far*. Computerworld, 17 juin 2002. <https://www.computerworld.com/article/2576616/the-story-so-far.html>.

Hayward Andrew. *MetaMask, Ethereum Apps Down as Infura Suffers Outage*. Decrypt, 22 avril 2022. <https://decrypt.co/98457/metamask-ethereum-apps-down-infura-outage/>.

Heal Jordan. *Hard forks: Contentious or not ?*, Coin Rivet, 16 janvier 2019. <https://coinrivet.com/hard-forks-contentious-or-not/>.

Hillard Frank. *Tezos - Vérification formelle*. OCTO Talks !, 4 septembre 2020. <https://blog.octo.com/tezos-verification-formelle/>.

Ho How Albert. *How Does Tokenization Work, Anyway ?* freeCodeCamp.org, 20 octobre 2018. <https://www.freecodecamp.org/news/how-does-tokenization-work-anyway-afb5fed1ac47/>.

Hughes Eric. *A Cypherpunk's Manifesto*, 9 septembre 1993. <https://www.activism.net/cypherpunk/manifesto.html>.

Hussey Matt. *Private vs Public Blockchains*. Decrypt, 21 janvier 2019. <https://decrypt.co/resources/private-blockchains/>.

Infura Blog. *The Benefits and Tradeoffs of Application-Specific Blockchains*, 17 janvier 2023.
<https://blog.infura.io/post/the-benefits-and-tradeoffs-of-application-specific-blockchains>.

Investopedia. *Hot Wallet vs. Cold Wallet*. Consulté le 6 juillet 2023.
<https://www.investopedia.com/hot-wallet-vs-cold-wallet-7098461>.

Jaspart Audrey. *Qu'est-ce qu'une tech stack ? Définition et exemples*. Consulté le 5 juillet 2023.
<https://blog.hubspot.fr/marketing/tech-stack>.

Journal du Net. *Framework ou infrastructure logicielle : définition et traduction*, 20 janvier 2019.
<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203355-framework/>.

Hathi Kamal. *Taking the Next Step in Our Smart Agreement Journey*, 27 mai 2021.
<https://www.docuSign.com/blog/clause-docuSign-smart-agreement-journey>.

Kar, Joon Ian Wong, Ian. *Everything You Need to Know about the Ethereum “Hard Fork”*. Quartz. Consulté le 9 décembre 2021. <https://qz.com/730004/everything-you-need-to-know-about-the-ethereum-hard-fork/>.

Kelly Liam J. *Making Ethereum Wallets Smarter Is the Next Challenge—and Visa Is Among Those Working on It*. Decrypt, 22 mars 2023. <https://decrypt.co/124100/making-crypto-wallets-smarter-ethereum>.

Kessler Sam. *Ethereum’s Layer 2 Rollups Reduce Costs, but the Risks Are Underappreciated*, 26 octobre 2022. <https://www.coindesk.com/tech/2022/10/26/ethereums-layer-2-rollups-speed-things-up-but-the-risks-are-underappreciated/>.

Larousse Éditions. *Définitions : automatisation*. Consulté le 15 août 2023.
<https://www.larousse.fr/dictionnaires/francais/automatisation/6753>.

Lars Ludovic.

Qu'est-Ce Qu'une Monnaie Fiat ?, Cryptoast, 20 décembre 2021.
<https://cryptoast.fr/monnaie-fiat-definition-explications/>.

Suivre ses cryptomonnaies avec un explorateur de blocs. Cryptoast, 10 octobre 2019.
<https://cryptoast.fr/suivre-cryptomonnaies-explorateur-blocs/>.

Ledger. Vesting, 21 février 2023. <https://www.ledger.com/academy/glossary/vesting>.

Lee Alexander. *What Is Programmable Money?*, FEDS Notes, 23 juin 2021.
<https://www.federalreserve.gov/econres/notes/feds-notes/what-is-programmable-money-20210623.html>.

Liesl Eichholz. *Beyond Monolithic: The Modular Blockchain Paradigm*. Fuel, 5 octobre 2022.
<https://fuel-labs.ghost.io/beyond-monolithic-the-modular-blockchain-paradigm/>.

Maire Vincent. *Qu'est-Ce Qu'un Consensus et Quels Sont Les Principaux En Crypto-Monnaie ?* Cryptoast, 11 janvier 2021. <https://cryptoast.fr/liste-differents-consensus-crypto-monnaies-blockchain/>.

Maldonado José. *What Is Timestamp on Blockchain ? - Bit2Me Academy*. Consulté le 28 février 2023.
<https://academy.bit2me.com/en/blockchain-timestamp/>.

Marin Gautier. *Why Application-Specific Blockchains Make Sense*. Medium, 8 février 2019.
<https://blog.cosmos.network/why-application-specific-blockchains-make-sense-32f2073bfb37>.

MetaMask. *MetaMask Learn | What Is a Crypto Wallet ?*. Consulté le 6 juillet 2023.
<https://learn.metamask.io/lessons/what-is-a-crypto-wallet>.

Moralis. *What Is OpenZeppelin? The Ultimate Guide*. Enterprise-Grade Web3 APIs, 13 novembre 2021. <https://moralis.io/what-is-openzeppelin-the-ultimate-guide/>.

Nakamoto Satoshi. *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008. bitcoin.org.

Nambiapurath Rahul. *What Are ZK-Rollups ? The Defiant*, 1 décembre 2022.
<https://thedefiant.io/what-are-zk-rollups>.

Narrative. *What Is Unix Time ?*. Consulté le 8 juillet 2023. <https://kb.narrative.io/what-is-unix-time>.

Nedelchev Miroslav. *Answer to “What’s the proper way to store a long string (like a news article) in Solidity?”* Ethereum Stack Exchange, 24 septembre 2019.

<https://ethereum.stackexchange.com/a/76195>.

Nomadic Labs. *The Road to a Million TPS (and beyond): Smart Rollups Are Coming*. Consulté le 3 juillet 2023. <https://research-development.nomadic-labs.com/smart-rollups-are-coming.html>.

O’Leary Rachel-Rose. *Blockchain Bloat: How Ethereum Is Tackling Storage Issues*, 18 janvier 2018. <https://www.coindesk.com/markets/2018/01/18/blockchain-bloat-how-ethereum-is-tackling-storage-issues/>.

OpenLaw. *Introducing OpenLaw*. Medium, 22 avril 2020.

<https://media.consensys.net/introducing-openlaw-7a2ea410138b>.

OpenZeppelin.

Governance - OpenZeppelin Docs. Consulté le 10 juillet 2023.

<https://docs.openzeppelin.com/contracts/4.x/api/governance>.

Proxies. Consulté le 9 juillet 2023. <https://docs.openzeppelin.com/contracts/4.x/api/proxy>.

O’reilly. *Getter Functions for State Variables - Mastering Blockchain Programming with Solidity [Book]*. Consulté le 9 juillet 2023. <https://www.oreilly.com/library/view/mastering-blockchain-programming/9781839218262/048537c8-6e8f-4cef-860a-b923a96946e3.xhtml>.

Pawan Nahar. *51% attack: What are 51% attacks in cryptocurrencies ?* - The Economic Times. Consulté le 28 septembre 2021.

<https://economictimes.indiatimes.com/markets/cryptocurrency/what-are-51-attacks-in-cryptocurrencies/articleshow/85802504.cms>.

Polak Kamil. *Hack Solidity: Reentrancy Attack*, 17 janvier 2022. <https://hackernoon.com/hack-solidity-reentrancy-attack>.

Polygon. *Build Your Own Blockchain, without the Complexity*. Consulté le 12 juillet 2023.

<https://polygon.technology/polygon-supernets>.

Pratap Zubin. *What Are Token Standards? A Complete List*. Chainlink Blog, 16 novembre 2022. <https://blog.chain.link/token-standards/>.

Prysmatic Labs. *What Happens After Finality in ETH2?* HackMD. Consulté le 16 mai 2023. <https://hackmd.io/@prysmaticlabs/finality>.

Rajeev Gopalakrishna. *Account Abstraction (EIP-2938): Why & What*. Our Status, 19 novembre 2020. <https://our.status.im/account-abstraction-eip-2938/>.

Roh Jooyun. *Parametric Drought Insurance: Case Studies From Latin America & Europe*. Descartes. Consulté le 4 avril 2023. <https://www.descartesunderwriting.com/whitepaper/parametric-drought-insurance-case-studies-latin-america-europe/>.

Sablier. *Sablier*. Consulté le 18 avril 2023. <https://www.sablier.finance>.

Solidity 0.8.13 documentation. *Solidity by Example*. Consulté le 19 mai 2023. <https://docs.soliditylang.org/en/v0.8.13/solidity-by-example.html#safe-remote-purchase>.

Stevens Robert. *What Is Arbitrum? Speeding Up Ethereum Using Optimistic Rollups*. Decrypt, 20 mars 2023. <https://decrypt.co/resources/what-is-arbitrum-speeding-up-ethereum-using-optimistic-rollups/>.

Strudwick James et Kugler Dylan. *A Comprehensive Guide to Rollups and Mina's Place in the Landscape*. Mina Protocol, 4 novembre 2023. <https://minaprotocol.com/blog/guide-to-rollups-and-minas-place-in-the-landscape>.

Sutour Jérôme. *Etat des lieux sur la réglementation des « nouveaux actifs »*. Consulté le 17 février 2023. <https://cms.law/fr/fra/news-information/etat-des-lieux-sur-la-reglementation-des-nouveaux-actifs>.

Takyar Akash. *Uber Blockchain Platform*. LeewayHertz - AI Development Company, 20 août 2018. <https://www.leewayhertz.com/blockchain-disrupting-uber-platform/>.

Techno-Science.net. *Programme informatique : définition et explications*. Techno-Science.net. Consulté le 5 juillet 2023. <https://www.techno-science.net/definition/5403.html>.

Tezos. *SmartPy*. Consulté le 30 juin 2023. <https://tezos.com/developers/smartpy/tezos.com/developers/smartpy>.

Twitter.

xTotem :: *web3sec* sur Twitter, 22 décembre 2022.
<https://twitter.com/OxTotem/status/1606269251727151105>.

michael rice, legal engineer sur Twitter, 2 novembre 2018.
<https://twitter.com/michaelriceLE/status/1058477183872581637>.

vitalik.eth sur Twitter, 11 octobre 2018.
<https://twitter.com/VitalikButerin/status/1051161357104635906>.

Uniswap Protocol. *Terms of Service*. Consulté le 18 mai 2023. <https://uniswap.org/terms-of-service>.

user610620. *Is Polygon (Matic) a layer-2 or a sidechain ?* Forum post. Ethereum Stack Exchange, 29 mars 2022. <https://ethereum.stackexchange.com/q/125024>.

UtorG. *What Is xDai On The Gnosis Chain ?* Consulté le 30 juin 2023.
<https://utorg.pro/blogs/what-is-xdai-on-the-gnosis-chain/>.

Vergara Ingrid. *Tezos, la blockchain aux racines françaises qui veut concurrencer Ethereum*. Le Figaro, 9 décembre 2020, Tech & Web. <https://www.lefigaro.fr/secteur/high-tech/tezos-la-blockchain-aux-racines-francaises-qui-veut-concurrencer-ethereum-20201209>.

Wikipedia. *Gestion des droits numériques*. Wikipédia, 12 mars 2021.
https://fr.wikipedia.org/w/index.php?title=Gestion_des_droits_num%C3%A9riques&oldid=180791087.

Yaga Dylan J., Peter M. Mell, Roby Nik, et Scarfone Karen. *Blockchain Technology Overview*. NIST, 3 octobre 2018. <https://www.nist.gov/publications/blockchain-technology-overview>

TABLE DES MATIERES

ECRITURE DE CONTRATS INTELLIGENTS – ESSAI DE METHODOLOGIE EN DROIT ET EN INFORMATIQUE	1
INTRODUCTION	3
I – Définition du sujet de recherche	3
II – Cadres de la recherche	10
III – Intérêt de la recherche	24
PARTIE I – DELIMITATION DU CONTRAT INTELLIGENT	31
Titre I – Le domaine substantiel du contrat intelligent	32
Chapitre I - Les processus sujets à une exécution dans la <i>blockchain</i>	33
Section I - Les transferts d’actifs matérialisés par des jetons	33
§ I - Les transferts d’actifs	33
A - Les processus de transferts	33
B - Les actifs	36
§ II - Les jetons	38
A - Définition d’un jeton	38
B - L’opération de « tokenisation »	42
Section II - Les conditions relatives au déclenchement des transferts	52
§ I – Une condition objective	52
A – La nécessité d’un évènement objectif	52
B - L’inadéquation des conditions subjectives	55
§ II – Une condition automatique	56
§ III – Une condition provenant de la <i>blockchain</i>	58
A - L’inopportunité de l’hybridation pour les évènements déclencheurs de transferts d’actifs	58
B - Une condition provenant de la <i>blockchain</i>	61
Chapitre II - Les contrats sujets à une exécution dans la <i>blockchain</i>	63
Section I - Les contrats essentiellement constitués de transferts d’actifs	63
§ I - Les contrats aux opérations de séquestre	63
A – Les contrats aux opérations de dépôts	65
B – Les contrats aux opérations de garanties	73
§ III - Les contrats modélisables en séquestre	81
A – Les contrats encadrant des opérations financières	82
B – Les contrats encadrant d’autres opérations simples	91
Section II - Les contrats partiellement constitués de transferts d’actifs	99
§ I - Les clauses des contrats <i>corporate</i>	99
A - Les clauses statutaires	100
B - Les clauses des pactes d’actionnaires	105
§ II - Les clauses relatives au paiement de sommes d’argent	109
A – Les clauses de paiement sous forme de séquestre	109

B - Les clauses de paiement modélisables en séquestre.....	112
Titre II – Le domaine formel du contrat intelligent	117
Chapitre I – Les différents <i>instrumentum</i> des contrats intelligents	117
Section I – Le code informatique comme seul acte instrumentaire.....	117
§1 – Le choix le plus répandu.....	118
A – Un choix dans l’esprit du milieu de la <i>blockchain</i>	118
B – Exemples de contrats intelligents sans texte écrit en langage naturel.....	120
§ II – Un choix pratique, mais risqué juridiquement	123
A – La praticité du code comme seul <i>instrumentum</i>	123
B – L’insécurité juridique lié au code comme seul <i>instrumentum</i>	124
Section II – Le texte écrit en langage naturel comme acte instrumentaire.....	128
§ I – Le texte en tant qu’ <i>instrumentum</i> prioritaire sur le code	128
A – Un choix minoritaire dans le monde de la <i>blockchain</i>	129
B – Un choix sécurisé mais porteur d’une certaine lourdeur	133
§ II - Le texte en tant qu’ <i>instrumentum</i> complémentaire avec le code	135
A – Genèse de l’idée	136
B – Un compromis entre praticité et sécurité	139
Chapitre II – Sélection de l’ <i>instrumentum</i> du contrat intelligent	142
Section I – Le choix de la méthode ricardienne.....	142
§ I - Définition du contrat ricardien	142
A – Origine du concept.....	142
B – Le contrat ricardien aujourd’hui	146
§ II – Recours à la méthode ricardienne	147
A – Un recours justifié par la sécurité juridique.....	147
B - Implémentation.....	149
Section II – Le corpus ricardien du contrat intelligent	152
§ I – Composition du corpus.....	152
A – Le contrat <i>fiat</i>	152
B – Les spécifications fonctionnelles et le code source	153
§ II – Les modalités de production du corpus.....	155
A – Organisation de la production.....	155
B – Charge de la production	157
PARTIE II – ÉLABORATION DU CONTRAT INTELLIGENT.....	160
Titre I – Rédaction des clauses indispensables du contrat intelligent	161
Chapitre I - Stipulations initiales du contrat.....	162
Section I - Définitions et identités.....	162
§ I – Clause relative à l’identité des parties	162
A – L’identité réelle des parties	163
B – L’adresse cryptographique	164
§ II – Clause de définition des termes	165
A – Définitions relatives à la blockchain	166
B – Définitions relatives aux smart contracts.....	170
Section II - L’opération d’ensemble du contrat	175
§ I – Clause de préambule	175

A – Contexte de conclusion du contrat.....	176
B – Exposition du projet.....	177
§ II – Clause d’objet du contrat.....	178
A – Choix d’une qualification juridique précise	178
B – Exclusion d’une qualification juridique	179
Chapitre II - Stipulations relatives à l’exécution du contrat	182
Section I – Les clauses intéressant les parties au contrat.....	182
§ I - Clauses relatives aux obligations.....	182
A – Exécution des obligations par <i>smart contract</i>	183
B – Obligation de paiement monétaire	189
§ II - Clause d'imprévision	193
A – L’évènement imprévisible déclencheur.....	194
B – La révision du contrat.....	198
§ III - Clause relative à la responsabilité	199
A - Faits générateurs de responsabilités.....	200
B – Indemnisation	203
Section II – Les clauses intéressant les tiers au contrat	205
§ I – Clause relative aux oracles	205
A – Typologie d’oracles	205
B – Défaillance et responsabilité.....	208
§ II - Clause de traitement de données personnelles.....	209
A – Gestion des données personnelles.....	211
B – Transfert de données personnelles	214
Chapitre III - Stipulations terminales du contrat.....	218
Section I - Interruption du contrat.....	218
§ I - Clause de force majeure	218
A – Qualification d’un évènement de force majeure	219
B – Régime juridique de la force majeure à suivre.....	220
§ II - Clauses de suspension et résiliation.....	221
A - Clause de suspension.....	221
B – Clause de résiliation.....	223
Section II – Clauses relatives à la résolution des conflits	224
§ I - Clauses de modes alternatifs de règlement des conflits.....	225
A – Intérêt des MARC pour un contrat intelligent	225
B – Contenu de la clause	227
§ II - Clauses attributive de juridiction et loi applicable.....	228
A - Clause attributive de juridiction	229
B - Clause de loi applicable	230
Titre II – Développement technique du contrat intelligent.....	233
Chapitre I – La détermination de la <i>blockchain</i> comme infrastructure d’exécution	234
Section I – Les différentes infrastructures	234
§ I – Les <i>infrastructures</i> généralistes	237
A – Les premières couches (L1)	238
B - Les deuxièmes couches (L2).....	249

§ II - Les <i>infrastructures</i> spécifiques	256
A - Les blockchains spécifiques.....	256
B - Les <i>rollup</i> spécifiques.....	259
Section II – La sélection des infrastructures	262
§ I - Les choix par principe.....	263
A – Ethereum.....	263
B - Un <i>rollup</i> généraliste sur Ethereum	265
§ II - Les choix par exception	266
A – Une <i>blockchain</i> privée	267
B – Un <i>rollup</i> spécifique sur Ethereum	270
Chapitre II – L’écriture des smart contracts	273
Section I – Prolégomènes.....	273
§ I - Présentation du <i>stack technique</i>	273
A - Nécessaire de création et d’interaction avec des smart contracts	273
B – Nécessaire de suivi et de stockage	281
§ II – Développement des <i>programmes</i>	284
A - Structure d’un <i>smart contract</i>	284
B – Eléments de grammaire en <i>Solidity</i>	287
Section II - Proposition de modèles de smart contract : legalTemplate.sol	307
§ I - Modules de base	307
A – <i>Smart contract</i> de liaison au contrat <i>fiat</i>	308
B – <i>Smart contract</i> du contrôle d’accès des parties et des tiers au contrat.....	309
C – <i>Smart contract</i> d’interruption, de modification et d’extinction du contrat	312
§ II- Modules optionnels	318
A – <i>Smart contract</i> de cession du contrat	319
B – <i>Smart contract</i> d’intervention d’oracles.....	326
C – <i>Smart contract</i> de règlement des litiges	330
CONCLUSION GENERALE.....	337
BIBLIOGRAPHIE	341
WEBOGRAPHIE.....	361
TABLE DES MATIERES.....	386