Smart contracts

Amélie Favreau

Janvier 2023
RAPPORT N°17.40

Institut des Études
et de la Recherche
sur le Droit et la Justice

Smart contracts

La première librairie européenne et ouverte de *smart contracts* à destination des professionnels du droit et de la justice

Sous la direction de

Amélie FAVREAU

UGA Université Grenoble Alpes
CRJA Université Grenoble Alpes
RAPPORT DE RECHERCHE

Cette recherche est issue de l'appel à projet : *Droit, justice et numérique*

Recherche réalisée sous la direction de :

Amélie FAVREAU,
Maîtresse de conférences HDR, université Grenoble-Alpes

Ont également contribué à ce rapport de recherche :

Sihem AMER-YAHIA,
Directrice de recherche CNRS, université Grenoble-Alpes

Aïda BENNINI,
Maitresse de conférences, université de Caen

Alexis BOISSON,
Maître de conférences, université de Montpellier

Maître Lionel CHARBONNEL,
Avocat, docteur en droit, Marseille

Anna COMPANY,
Étudiante à la faculté de droit de l'université Grenoble-Alpes

Abdoulaye DIALLO,
Doctorant à l'université Grenoble-Alpes

Thomas FÉRAUD,
Étudiant à Centrale Supélec, université. Paris Saclay

Julien FONTRIER,
Étudiant en 2e année de DUT Carrières juridique à l'IUT2 de Grenoble

Maître Fabien GILLIOZ,
Avocat associé, Cabinet Ochsner et Associé, Genève (Suisse)

Julien GOSSA,
Maître de conférences, université de Strasbourg, ICube (Informatics)

Maître Élise GUILHAUDIS,
Avocate inscrite au barreau de Grenoble et gérante de la Legaltech NUMETIK Avocats

Benjamin JEAN,
Consultant, Inno3, Open Law Association

Nicolas JONDET,
Chercheur, université d'Édimbourg, *SCRIPT*

Nathan ROUGIER,
Étudiant en 2e année de DUT Informatique à l'IUT2 de Grenoble

Alexandre VILAY,
Doctorant à l'université Grenoble-Alpes

Nos remerciements vont au Centre de recherches juridiques de l'université Grenoble-Alpes pour leur soutien à cette recherche ; à la société Crypto4All dirigée par Rémy OZCAN pour les échanges constructifs issus de leur pratique des *smart contracts* et à la société ZELAB pour la conception de la plateforme de *smart contracts*.

Le présent document constitue le rapport scientifique d'une recherche réalisée avec le soutien du Groupement d'Intérêt Public Institut des études et de la recherche sur le droit et la justice (convention n° 17.40). Son contenu n'engage que la responsabilité de ses auteurs.

Sommaire du rapport

INTRODUCTION

Les *smart contracts* dans le paysage juridique. Les technologies émergentes ont fait apparaître de nouvelles applications comme les cryptomonnaies, les jetons non fongibles et les *smart contracts* déployés sur des *blockchains*. Chacun de ces objets techniques questionne notre droit sur sa capacité de régulation et par conséquent sur les modalités de sa réception. Les *smart contracts* sont des programmes informatiques déployés sur des *blockchains* qui s'exécutent automatiquement sans l'intervention humaine. Les *smart contracts* sont largement utilisés dans le domaine de la finance dans lequel ils assurent la circulation des flux de crypto monnaies, mais également la circulation de jetons (ou *tokens*) non fongibles. En somme, ils sont des outils au service de la circulation de la valeur sur une *blockchain*. D'autres potentialités sont envisageables pour les *smart contracts*. En effet, ils peuvent aussi accompagner l'exécution de stipulations contractuelles. Le présent rapport s'attache à ce dernier volet, à savoir l'utilisation du *smart contract* comme outil complémentaire à l'exécution d'un contrat.

Définitions. Afin de mieux comprendre les travaux de recherche explicités ici, il convient tout d'abord de dresser une définition précise de ce qu'est un *smart contract* ainsi que le support dans lequel il s'inscrit : la *blockchain*. Premièrement, la *blockchain* est une technologie de stockage et de transmission d'information décentralisée et sécurisée. Le terme « *blockchain* » désigne plus précisément une suite de transactions organisées en blocs qui relie le bloc « genèse » au bloc le plus récent formant ainsi une chaîne dont les données sont vérifiées à intervalles de temps réguliers. Ce système de vérification permet de protéger chacun des blocs de la chaîne contre toute modification. L'utilité de la *blockchain* s'observe au travers de sa capacité à maintenir en mémoire tous les échanges effectués entre les utilisateurs depuis sa création grâce à son système d'enregistrement en continu des données produites. Ces précisions sur ce qu'est la *blockchain* permettent désormais de mieux identifier ce qu'est un *smart contract*. Il s'agit d'un protocole informatique qui s'occupe de la vérification, la négociation ou l'exécution d'un contrat. Les *smart contracts* ou contrats intelligents sont régis par un code informatique enregistré dans une *blockchain*. Leur exécution est alors déclenchée par une transaction sur cette *blockchain*. Ce système permet une exécution automatique des contrats et assure ainsi leur force obligatoire. Enfin, pour effectuer les transactions qui viennent déclencher

l'exécution des *smart contracts,* il est nécessaire de disposer de jetons ou *tokens.* Il s'agit d'une représentation numérique d'un actif ou d'un droit qui peut être transférable entre deux parties sur une *blockchain*. Les jetons ou *tokens* sont inclus dans la catégorie des actifs numériques au sens de l'article L 552-2 du Code monétaire et financier qui les définit comme suit : « 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 »[1]. Ces trois définitions en tête, il convient à présent d'exposer en quoi ce projet portant sur les *smart contracts* revêt un intérêt particulier.

L'intérêt d'un projet de recherche sur les *smart contracts*. Les métiers du droit et de la justice évoluent sous l'impulsion du numérique. En effet, magistrats, avocats et notaires, entre autres, témoignent de véritables inquiétudes quant à ces nouvelles pratiques de programmation, d'automatisation ou de prédiction de certaines opérations juridiques. Le sujet est complexe. Il a été fait le choix de cibler la problématique sur l'analyse d'un objet technique en plein essor le *smart contract* ou contrat autoexécutant dans le cadre de l'exercice des professions juridiques et plus spécifiquement celle d'avocat. Il s'agissait donc de conduire un projet empirique sur les *smart contracts* en vue d'évaluer la faisabilité de traduire en langage informatique des stipulations contractuelles, les clauses ainsi traduites ayant vocation à s'insérer dans la première librairie de *smart contracts* en France. L'objectif était de déposer sur une plateforme en accès ouvert un clausier, qui répertorie sur plusieurs occurrences la traduction informatique de clauses françaises traduites en anglais pour faciliter un accès international. En d'autres termes, le projet « *Smart contracts* » visait à faire cohabiter un triple langage : un langage « juridique » et un langage « naturel » en français et en anglais avec un langage informatique « strict » et « formel ». L'exercice a mis en jeu des collaborations entre juristes franco-britanniques et informaticiens, chacun détenant un savoir-faire indispensable à la réalisation de cette nouvelle forme d'écriture du droit. Si la rédaction de stipulations contractuelles en langage informatique est connue à travers des modèles de *smart contracts,* le travail qui consiste sur plusieurs occurrences à traduire quelques clauses françaises et anglaises en *smart contracts* pour les accompagner d'explications juridiques et techniques était à notre connaissance encore mondialement inédit. L'intérêt de ce clausier est de permettre aux professionnels du droit de ne

[1] Voir l'article L 552-2 du Code monétaire et financier crée par la LOI n°2019-486, Loi PACTE, du 22 mai 2019.

pas perdre la main sur la technologie et au-delà de parvenir à se l'approprier pour pouvoir la proposer dans le cadre de leur activité.

La démarche retenue pour la conduite de cette recherche. Cette réalisation a impliqué trois étapes de recherche. La première a consisté à circonscrire les défis techniques et juridiques et à évaluer les solutions pour les dépasser. Une fois cette première étape franchie venait un travail de sélection entre les stipulations contractuelles et les différents langages informatiques. L'on sait que de nombreuses stipulations peuvent difficilement être traduites en contrat autoexécutants, comme celles impliquant des notions-cadres. Parmi les autres stipulations, les membres du projet ont effectué des choix pour sélectionner les plus utiles et pertinentes pour les professionnels du droit au sein des deux systèmes juridiques étudiés. Enfin, l'exercice a été complété par un indispensable travail d'explication sur la sélection opérée et la mise en œuvre de l'autoexécution, tant technique que juridique. Ce travail était nécessaire pour favoriser l'accessibilité et l'intelligibilité du clausier trilingue, mais également pour contribuer à l'élaboration d'un cadre en droit et en informatique pour l'essor des *smart contracts*. Pour parvenir à la réalisation de cet objectif, l'équipe de recherche mise en place se devait d'être à la fois internationale et pluridisciplinaire.

Une démarche pluridisciplinaire et internationale. La vocation pluridisciplinaire de l'équipe de recherche était indispensable tant le sujet confond la maîtrise du droit et de l'informatique. Au-delà des disciplines, c'est aussi les pratiques qui sont mises en commun. En effet, l'équipe est composée de professionnels du droit couvrant de larges champs de compétence (en droit des assurances, droit de la consommation, droit de l'immobilier, droit de la santé et des données de santé, droit de la construction, droit des sociétés et des entreprises en difficulté, droit de la propriété intellectuelle et droit de l'Internet) qui ont contribué avec les chercheurs des universités de Grenoble-Alpes, de Strasbourg, de Chambéry et de Montpellier à sélectionner les stipulations contractuelles les plus pertinentes pour réaliser le clausier. La vocation internationale de l'équipe a permis d'enrichir les perspectives de cette réalisation, et ce pour trois principales raisons. Premièrement, il était indispensable de dépasser les frontières françaises pour étudier sans naïveté la technologie du *smart contract* et anticiper le risque de voir un jour remplacer la devise contractuelle « loyauté, solidarité, fraternité »[2] par celle de « rapidité, sécurité et simplicité ». De plus, le regard porté sur les systèmes étrangers a contribué

[2] Denis Mazeaud, « Loyauté, solidarité, fraternité : la nouvelle devise contractuelle ? », dans Philippe Ardant (dir.), *L'avenir du droit. Mélanges François Terré*, *Jurisclasseur*, 1999, p. 603 s.

à ouvrir l'étude sur les *smart contracts* à d'autres domaines que la sphère économique, notamment sociétaux (vie publique, éducation et santé). À ces fins, le projet de recherche a associé une équipe de recherche en droit de l'université d'Édimbourg, le Centre de recherche *SCRIPT*, représenté par son directeur, le Pr Burkhard Schafer et un de ses chercheurs, Nicolas Jondet. Outre l'apport de compétences de chercheurs ancrés dans la pratique du droit indispensables pour cette recherche, l'étude d'autres systèmes juridiques est venue enrichir la réflexion. En effet, le clausier tel que nous l'avons envisagé est trilingue avec la présentation de clauses présélectionnées françaises et britanniques et leurs traductions en langage informatique pour assurer leur autoexécution sur une *blockchain*. L'intérêt de l'ouverture internationale de la recherche est la possibilité d'implémenter de nouvelles stipulations issues de systèmes étrangers. En droit suisse, notamment, la recherche sur les *smart contracts* est particulièrement avancée.

Nous présentons dans un premier temps, les travaux préparatoires à la réalisation de la librairie (chapitre 1) puis sa réalisation (chapitre 2).

Chapitre 1. Travaux préparatoires à la réalisation de la librairie

Section 1 : Choix techniques sur la programmation des *smart contracts*

Pour la réalisation d'une librairie de *smart contracts*, il est nécessaire d'identifier l'ensemble des verrous techniques. Pour cela, une connaissance de la *blockchain* et des *smart contracts* est indispensable (§1) ainsi qu'une précision sur l'état de l'art et les préconisations techniques pour l'écriture des *smart contracts* (§2).

Paragraphe 1 : Considérations générales sur la *blockchain* et les *smart contracts*

Le défi principal du développement des *blockchains* est de répondre au trilemme décentralisation, scalabilité et sécurité. Les regards croisés du droit et de l'informatique sur les *blockchains* sont à la fois sources de richesses et parfois d'ambiguïtés tant les acceptions de ce phénomène technique s'inscrivent dans des perspectives différentes. Les développements futurs des technologies liées aux *blockchains* se retrouvent confrontés à ce que Vitalik Buterin, le fondateur d'Ethereum (ETH), a nommé pour la première fois en 2019 ce qu'il est convenu d'appeler le « trilemme de la *blockchain* » qui met en avant trois dimensions auxquelles les développeurs (et les législateurs) se voient confrontés : le degré de décentralisation, la scalabilité et enfin le degré de sécurité. L'étude de ce trilemme permet de comprendre les limites auxquelles peut se heurter le développement d'une application. Si l'on prend l'exemple d'une cryptomonnaie et qu'on la confronte à ces trois critères, Vitalik Buterin souligne qu'elle ne peut se développer sur les trois fronts à la fois. La sécurité apporte à la cryptomonnaie la réalisation de transactions non modifiables, authentiques et résistantes aux attaques. La cryptomonnaie décentralisée ne dépend pas d'un organe central et peut fonctionner en réseau sans contrôle. La scalabilité est la potentialité d'adaptation aux volumes du réseau. Ainsi, si les concepteurs d'une cryptomonnaie privilégient la sécurité et la décentralisation, celle-ci aura du mal à adapter son réseau au plus grand nombre, tandis que miser sur la sécurité et la scalabilité conduit à

développer des applications moins décentralisées, de sorte que le développement d'une cryptomonnaie décentralisée et scalable le serait au détriment de la sécurité. Cette réflexion sur le trilemme des *blockchains* est importante pour le développement de *smart contracts*, qui sont une application de la *blockchain*.

Il convient donc d'expliciter chaque élément de ce trilemme, la décentralisation (A), la scalabilité (B) et la sécurité (C) pour apprécier en conclusion lequel de ces éléments est à privilégier dans le cadre de cette recherche.

A. Centralisation, *cloud* et décentralisation

Les systèmes centralisés. La centralité d'un système, et par opposition sa décentralité, se réfère à la localité physique des calculs et données. Dans un système complètement centralisé, tout se situe en une seule et même localité : matériel et logiciel (code et données informatiques). Cela en fait un système relativement simple à mettre en œuvre et à sécuriser. Il est par exemple aisé de retrouver une information ou d'en sécuriser l'accès lorsqu'elles sont toutes au même endroit. Celui qui contrôle physiquement la localité du système contrôle par extension tout le système. Il en est aussi totalement responsable.

Les systèmes partagés : le *cloud*. L'évolution des technologies et notamment des réseaux de communication, dont le coût a drastiquement baissé pendant que les performances augmentaient, a conduit à ce que les systèmes soient de moins en moins centralisés. Cette évolution a abouti à la mise au point de *clouds*. Dans un *cloud*, les différents composants d'un système sont partagés entre les différents acteurs : un *hébergeur*, qui détient l'infrastructure matérielle et donc la localité physique du système ; un *fournisseur de service*, qui loue cette infrastructure matérielle pour y loger son logiciel (exécuter le code et stocker les données) et dispose pour cela d'un accès distant ; le *fournisseur de matériel* lui-même, qui ne peut pas nécessairement accéder à la partie logicielle. Dans ce cas de figure, le contrôle et la responsabilité sont partagés : matériel pour le fournisseur de *cloud* et logiciel pour le fournisseur de service. Ce découplage matériel/logiciel est renforcé par l'utilisation d'une technologie appelée « virtualisation », qui permet de déplacer facilement et de façon totalement transparente pour toutes les parties la partie logicielle d'un matériel à l'autre. L'infrastructure matérielle pouvant être exploitée par plusieurs fournisseurs de services indépendants, on dit ainsi qu'elle est partagée ou mutualisée. Le *cloud* tient donc son nom du fait que les clients finaux du service ne savent plus où se situent leurs services et leurs données : ils sont *dans le cloud*, *dans le nuage*.

Si cette méconnaissance est bien pratique (après tout, savoir comment sont stockées ses données est plus une charge qu'un plaisir), elle pose également un problème de perte de contrôle sur ces données, y compris celles à caractère personnel.

Les systèmes décentralisés. À l'inverse des systèmes centralisés ou partagés comme le *cloud*, qui permettent d'identifier des responsables, les systèmes totalement décentralisés reposent sur la participation de *pairs (peers)* sans partie centrale clairement identifiable. Dans ces systèmes, les pairs ont souvent un rôle équivalent. Ils apportent leurs propres matériels et leurs propres données pour constituer le système. L'exemple le plus connu de systèmes pair à pair (P2P) concerne l'échange de fichiers musicaux ou vidéos, tels que BitTorrent. Ces systèmes ont d'ailleurs été partiellement utilisés pour échapper à la surveillance des ayants droit.

Bien que plus difficiles à mettre en œuvre, moins performants et moins fiables qu'une architecture centralisée, ces systèmes ont l'avantage d'être difficilement contrôlables : en l'absence d'un point physique unique (central), ce contrôle nécessite de s'intéresser à chacun des pairs, ce qui est d'autant plus difficile qu'ils sont nombreux et peuvent rejoindre ou quitter le système à tout moment. En outre, le *cloud*, par exemple, implique des ententes commerciales, ce qui n'est pas nécessairement le cas des systèmes décentralisés qui peuvent donc permettre un réel anonymat. Par ailleurs, chaque acteur étant naturellement indépendant dans un système distribué et en l'absence structurelle d'une vue globale, il est très difficile de savoir qui fait quoi dans de tels systèmes.

En plus d'être structurellement plus lents et complexes, la contrepartie évidente de ces caractéristiques est que ces systèmes sont davantage exposés aux malveillances : il est beaucoup plus facile d'avoir un comportement malveillant dans un système décentralisé, que l'on peut rejoindre et quitter à tout moment et dans lequel on va pouvoir agir sans surveillance, que dans un système centralisé, où celui qui en a le contrôle peut imposer une authentification, savoir qui fait quoi et bannir en conséquence.

B. La scalabilité

Définition. La scalabilité d'un système désigne son aptitude à conserver des performances satisfaisantes lors d'un changement d'ordre de grandeur d'une de ses dimensions : nombre de machines, taille ou historique des données, besoins de calculs, nombre de participants... Pour rappel, une *blockchain* est un registre distribué, dont chaque transaction est validée par un habile

mécanisme de consensus. Lorsqu'on parle de sa scalabilité, on vise concrètement sa capacité à valider un nombre élevé de transactions par unité de temps (transactions par seconde, ou *débit*), ainsi que sa capacité à intégrer un nombre élevé d'acteurs (utilisateurs ou *mineurs*). Ces deux nombres sont souvent corrélés, mais il peut exister des *blockchains* avec peu d'acteurs réalisant de très nombreuses transactions, comme des *blockchains* avec beaucoup d'acteurs réalisant peu de transactions.

Illustration avec la *blockchain* Ethereum. Seules 15 transactions par seconde sont validées sur la *blockchain* Ethereum aujourd'hui. Cela la rend impropre à accueillir un nombre massif d'utilisateurs ou à soutenir des applications gourmandes en transactions, car à mesure que le nombre d'utilisateurs augmente, la validation des transactions prend plus de temps et ralentit les applications fonctionnant sur le réseau. C'est pourquoi la scalabilité est le sujet principal sur lequel travaillent les développeurs d'Ethereum (dits les « *core* »), puisqu'il s'agit du point sur lequel ils peuvent le plus influencer son fonctionnement afin d'accélérer l'adoption de leur technologie. La solution trouvée semble être de transiter du mécanisme de consensus du « *proof of work* », où le valideur d'un bloc de transactions est choisi après qu'il ait réussi en premier à solutionner un problème cryptographique, vers le « *proof of stake* », où les aspirants valideurs mettent en jeu une partie de leurs cryptoactifs afin d'être choisis aléatoirement comme valideurs d'un bloc (plus le montant mis en jeu est élevé, plus on a de chance d'être choisi). Ce dernier mécanisme aurait l'avantage de requérir bien moins de consommation électrique que celui du *proof of work* et permettrait une validation des transactions exponentiellement plus rapide.

Les *blockchains* privées et la scalabilité. Les *blockchains* qui utilisent un consensus par preuve d'autorité (*proof of authority)* fonctionnent de manière limitative. En effet, les blocs peuvent être validés seulement par les comptes qui ont été préalablement approuvés. En d'auteurs termes, seuls certains nœuds ont l'autorité nécessaire pour valider les transactions au sein d'un bloc de la chaîne. C'est le consensus majoritairement adopté dans les *blockchains* privées ou de consortium. Le choix de ce type de consensus réintroduit nettement les tiers de confiance dans le cadre de registres distribués caractérisés par l'absence d'autorités centrales. La liste des valideurs peut être automatisée au moyen de *smart contracts*. Il n'existe pas dans ces *blockchains* de minage ni de mineurs et la probabilité d'une attaque des 51 %[3] est réduite.

[3] Il s'agit d'une attaque qui cible la *blockchain* et dont le but est de bloquer les validations des transactions ou bien de créer une double dépense, c'est-à-dire réitérer une transaction et ainsi modifier l'historique de la *blockchain*. Voir lexique p. 215.

C. La sécurité

La sécurité dans les systèmes informatiques est un des domaines les plus complexes, qui dépasse largement le cadre de ce document. Nous survolerons donc succinctement différents éléments de ce problème pour permettre d'en comprendre simplement les enjeux.

- **Disponibilité** : un système est disponible lorsqu'il rend le service attendu dans les temps attendus. Cela implique notamment de ne pas être en panne, suite à des défaillances matérielles ou logicielles ou encore des malveillances.
- **Intégrité** : l'intégrité d'un système, qu'on regarde souvent sous l'angle de ses données, concerne le fait que ces données soient conformes à une forme de vérité interne au système. Elles ne doivent donc pas être altérées ou modifiées par des dysfonctionnements ou des malveillances.
- **Confidentialité** : la confidentialité d'un système ou d'une donnée indique le fait que son accès est strictement subordonné à un droit d'accès, assurant que seules les personnes autorisées y ont accès.
- **Traçabilité** : la traçabilité d'un système est sa capacité à permettre de savoir qui y a fait quoi et quand. C'est notamment indispensable pour avoir un historique des actions, mais également pour remonter les traces d'une action malveillante.
- **Authentification** : l'authentification est l'art de savoir qu'un acteur est bien celui qu'il prétend être. C'est notamment indispensable pour assurer la confidentialité des données.
- **Non-répudiation et imputation** : la non-répudiation et l'imputation désignent la capacité du système à associer de façon sûre les actions à un utilisateur donné, sans que celui-ci puisse les renier. C'est notamment indispensable pour assurer la traçabilité des actions.

Conclusion sur le développement de la librairie de *smart contracts* et le trilemme des *blockchains*. Les développeurs des *smart contracts* dans le cadre de la librairie se sont confrontés aux trois dimensions : son degré de décentralisation, sa scalabilité et enfin son degré de sécurité. Ne pouvant développer une application opérante sur les trois fronts, il a été nécessaire d'opérer des choix. Cette librairie à destination des professionnels du droit et de la justice ne pouvait en aucun cas concéder des failles de sécurité. L'intégrité, l'authentification ou la traçabilité des transactions que permettent les *smart contracts* sont autant de critères indispensables pour leur programmation.

Le développement de *smart contracts* sur la *blockchain* Ethereum qui permet historiquement cette programmation répond au choix de la scalabilité. Le système peut supporter une charge importante de transactions et s'ouvrir à un grand nombre d'utilisateurs. L'alternative aurait été de développer les *smart contracts* de la librairie dans une *blockchain* privée. Mais cela aurait abouti à faire fi de la transparence. Or, la validation par le plus grand nombre est un élément fondamental de la confiance dans le système et cette confiance est un élément indispensable à l'adoption de *smart contracts* dans le domaine du droit et de la justice.

Enfin, la décentralisation est l'élément sur lequel l'équipe a accepté le plus de concessions. Deux points sur lesquels nous reviendrons permettent de l'illustrer. En premier lieu, le choix des clauses qui a été réalisé impliquait de collecter des informations situées à l'extérieur de la chaîne. Ces liens avec l'extérieur de la chaîne, que les informations soient ou non récupérées par des oracles[4], sont autant de failles à la décentralisation. En second lieu, nous avons, lorsque cela était nécessaire pour garantir la sécurité des *smart contracts* et de leur exécution, introduit une fonction « contrôleur » qui peut intervenir voire interrompre le déroulement du *smart contract*. Cette fonction est codée à partir du standard ERC-1644: Controller Token Operation Standard[5] (inclus dans ERC-1400: Security Token Standards) et permet techniquement à l'émetteur d'un jeton de déclarer de manière transparente si un contrôleur peut ou non transférer unilatéralement des jetons entre plusieurs adresses. Le principal objectif de cette fonction est de permettre à l'émetteur/contrôleur de conserver la capacité de transférer de force les jetons entre les adresses. Ces transferts de contrôleur doivent être transparents (émettre des événements signalant qu'il s'agit d'un transfert forcé) et le *smart contract* lui-même doit être explicite quant à la possibilité de le faire ou non. Cela peut être indispensable par exemple lorsqu'il est nécessaire d'annuler des transactions frauduleuses, de procéder à la résolution de clés privées perdues ou de répondre à une décision de justice.

Paragraphe 2 : Considérations particulières sur la réalisation du clausier

Quatre questions ont été soulevées par le groupe en vue de la réalisation de la librairie. La première implique une étude de l'état de l'art sur les plateformes de mise à disposition de *smart contracts* pour éclairer l'originalité de la librairie proposée (A). La deuxième dresse un inventaire des langages de programmation disponibles afin de choisir le langage adapté (B). La

[4] Tiers de confiance.
[5] https://github.com/ethereum/eips/issues/1644.

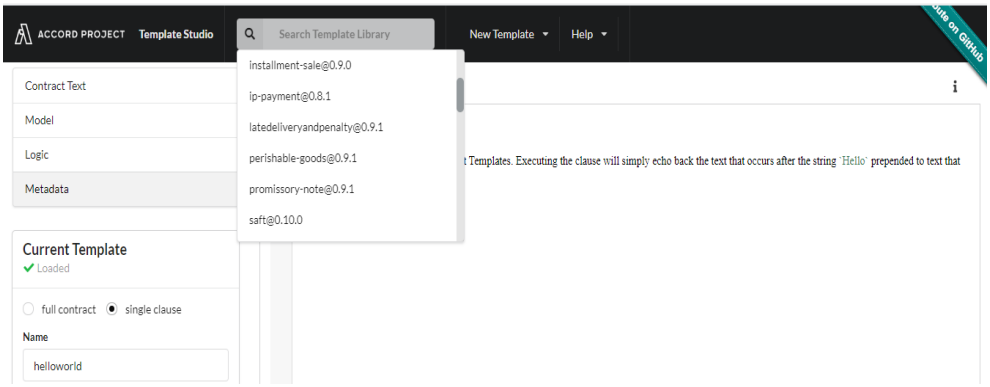
troisième question est relative à une étape supplémentaire dans le processus de traduction des *smart contracts*. En effet, il est difficile de passer du langage naturel au langage informatique, aussi nous avons utilisé une étape intermédiaire appelée « pseudo-code » (C). Enfin, la quatrième question envisage la liberté d'accès et de reproduction des langages disponibles dans les différentes librairies de *smart contracts* et soulève les questions de responsabilité (D).

A. État de l'art sur les plateformes de *smart contracts*

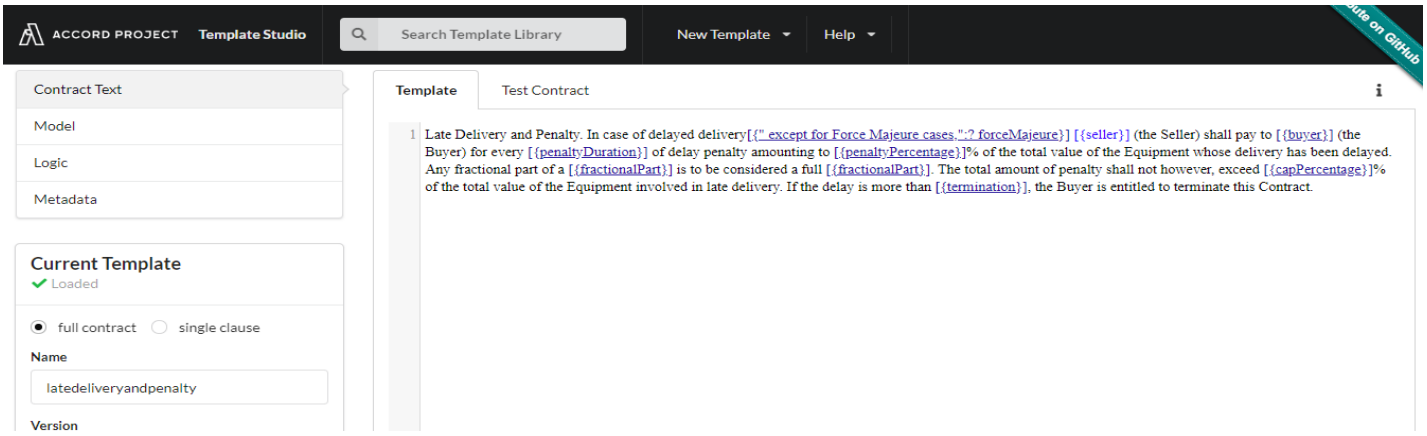
Au moment du dépôt du projet, en mai 2017, la proposition faite de créer une plateforme compilant les traductions en langage de programmation de stipulations contractuelles **était mondialement inédite**. Durant l'année 2018, aux États-Unis, se sont développées deux initiatives privées : Accord Project (1) et OpenLaw (2). Il est à noter que ces plateformes ont reçu pour leur développement des soutiens financiers de plusieurs millions de dollars de partenaires privés. Nous apprécions enfin de manière comparative ces plateformes par rapport à la librairie de *smart contracts* que proposait de réaliser l'équipe de recherche (3).

1. Accord Project

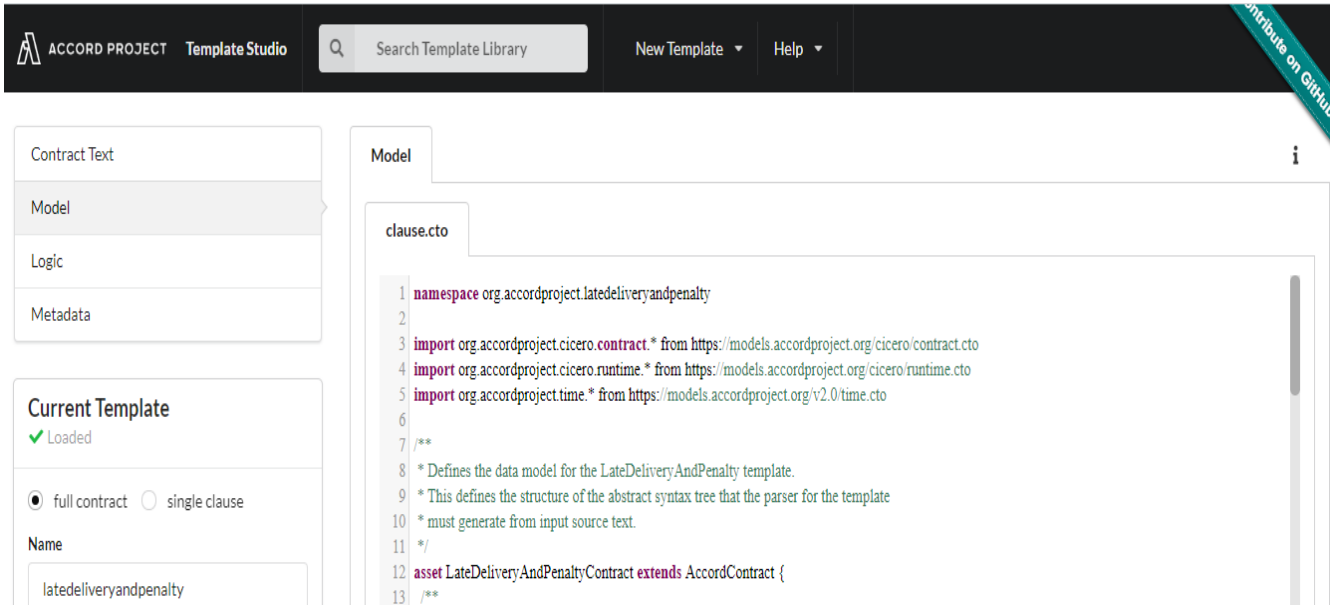
Présentation générale. Accord Project est issue de la société Clause, qui est une LegalTech américaine proposant des services '*Software-as-a-Service*' avec la génération automatique de contrats, en temps réel et en considération des éléments commerciaux des clients. Le but d'Accord Project (qui se présente comme une organisation) est d'accélérer l'adoption des *smart legal contracts* en établissant des standards et des bonnes pratiques et des librairies de clauses. Accord Projet a été initié par un consortium entre Linux Foundation's Hyperledger et l'International Association for Commercial and Contract Management (IACCM). L'IACCM est une organisation qui compte plus de 50 000 membres à travers le monde. À cet effet, ils se sont mis en partenariat avec un nombre pléthorique de gros cabinets d'avocats (PwC, Norton Rose, DLA Piper, Dentons, pour ne citer qu'eux). Ils ont notamment développé un nouveau langage appelé Ergo, servant à écrire des *smart legal contracts* s'interfaçant sur les *blockchains* Ethereum, Fabric et Corda. Le travail d'Accord Project a donc une dimension internationale. Il propose un « éditeur » permettant d'écrire le contrat ou la clause en langage naturel, puis de définir ses variables (ce qu'ils appellent le « *model* ») et enfin d'écrire une logique de développement commercial. La bibliothèque d'Accord Project est bien moins conséquente que celle d'OpenLaw. Elle est ouverte et les utilisateurs sont libres d'y contribuer.



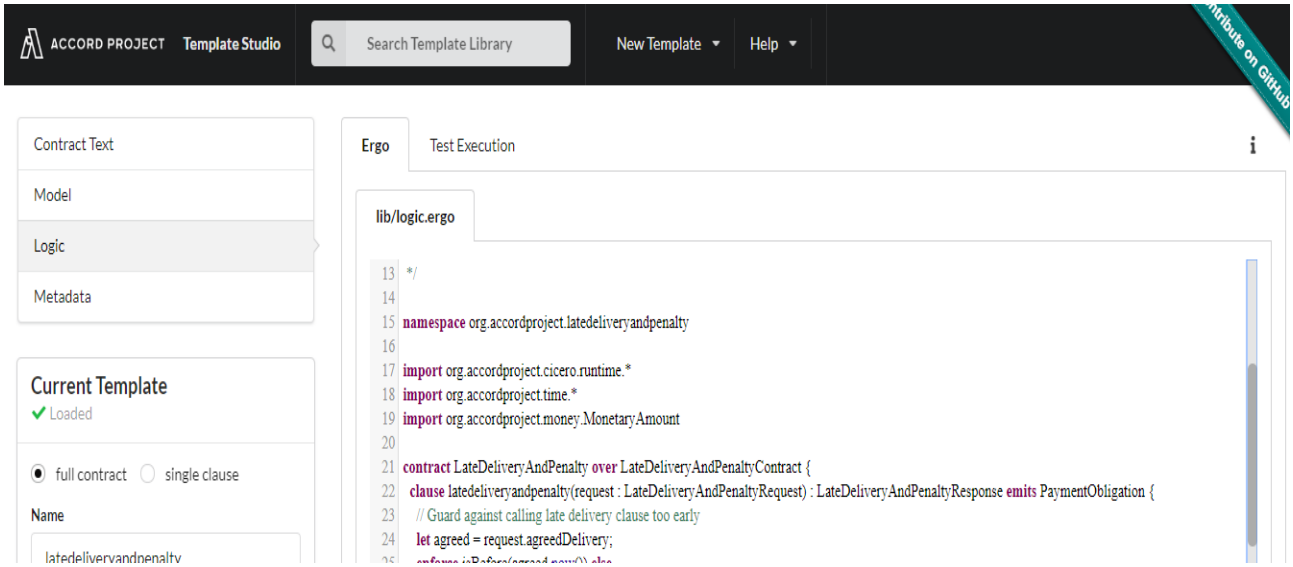
Capture d'écran 1 : Liste de *smart legal contracts* proposés



Capture d'écran 2 : Clause ou contrat écrit en langage naturel



Capture d'écran 3 : « ***Model*** » de la clause



Capture d'écran 4 : Logique de la clause exprimée dans le langage Ergo

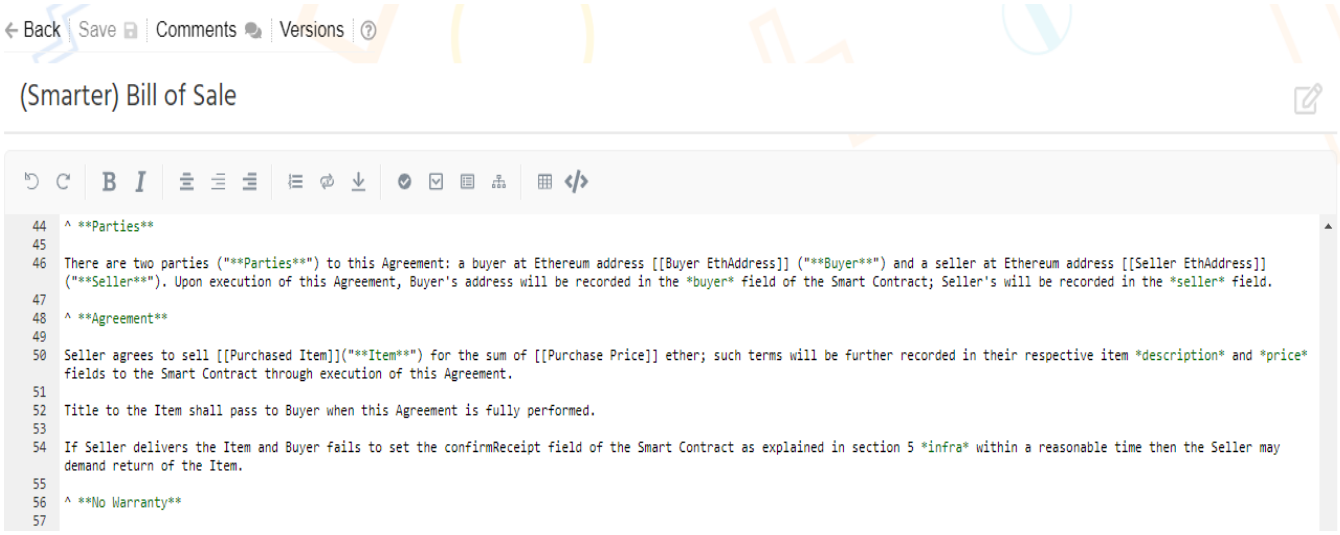
La société Clause, derrière l'initiative Accord Project, a annoncé quelques jours après l'annonce du partenariat entre RocketLawyer et OpenLaw, s'être entendue avec LegalZoom (1ère rivale de RocketLawyer), pour lancer une plateforme de création de *smart legal contracts*.

On mesure par ces différentes initiatives les marchés et les enjeux financiers qui les sous-tendent.

2. OpenLaw

Présentation générale. OpenLaw est une société cofondée par Aaron Wright, professeur de droit américain à l'École de droit Cardozo Law de l'université Yeshiva à New York. La société a levé des fonds auprès de Consensys, fondée par Joseph Lubin (cofondateur d'Ethereum), qui est une sorte de pépinière de startups utilisant la *blockchain* Ethereum. OpenLaw ambitionne de déployer un protocole facilitant la création de *smart legal contracts* – il faut entendre par là des contrats augmentés grâce aux *smart contracts* sur la *blockchain* Ethereum. Rien n'indique qu'ils souhaitent s'étendre au-delà des États-Unis, même si à terme cela doit certainement faire partie de leurs objectifs. Le site internet contient d'abord une large bibliothèque de documents juridiques librement fournis par les utilisateurs. On trouve dedans tout type de contrats, certains dans des langues étrangères. Les documents en français sont rares et l'on trouve seulement des statuts d'une association Loi 1901. Le site présente également un outil de création de contrats, qui est l'élément le plus intéressant. Concrètement il s'agit d'un éditeur avec lequel on peut

écrire tout acte juridique à automatiser pour y inclure des « *markups* » / variables afin qu'il puisse être rempli par un utilisateur via un formulaire intelligent. La génération du *smart contract* et son remplissage sont automatisés.



Capture d'écran 5 : un exemple de l'éditeur de contrat pour un contrat de vente américain

Capture d'écran 6 : *Le formulaire permettant de remplir le modèle*

Leur sérieux dans l'écosystème et leur partenariat avec Consensys les ont conduits fin 2018 à s'entendre avec Rocklawyer (la première LegalTech au monde en termes de clients – spécialisée dans la génération de documents juridiques) pour lancer un produit nommé Rocketwallet : **https://www.rocketlawyer.com/rocketwallet** qui doit consister en un outil permettant la création de documents juridiques sur Rocketlawyer augmentés de *smart contracts*.

3. Comparaison des plateformes Accord Project et OpenLaw à la librairie de smart contracts

Les différences entre ces plateformes. Nous notons trois différences principalement.

Une première différence importante avec le présent projet tient à l'existence de partenariats et aux niveaux de financement, qui nous ont fait prendre conscience de son caractère ambitieux. C'est une dimension que nous n'avions pas imaginée au moment de la rédaction du projet (soumission en mai 2017), car ces plateformes américaines n'existaient pas encore.

La deuxième différence a trait à la vision américaine du *smart contract*, qui s'inscrit dans une démarche juridique libérale. En effet, l'efficacité juridique est ici recherchée, au détriment d'une réflexion d'ensemble sur la place que doit tenir l'exécution automatique en droit et sur les modalités de sa mise en œuvre. En somme, dans ces plateformes tout semble « smart-contractualisable ». Notre entreprise de sélection, effectuée par des chercheurs français de haut niveau, est un avantage considérable et marque notre volonté affirmée de produire une librairie de *smart contracts* qualitative.

La troisième différence est liée au mode de génération du *smart contract*. Sur la plateforme OpenLaw, comme lors de la plupart des générations automatiques de *smart contracts*, les utilisateurs entrent leurs informations personnelles et commerciales dans un logiciel, qui produit automatiquement à partir d'un document standard un contrat personnalisé. L'étape supplémentaire proposée par OpenLaw est la traduction de l'intégralité de ce contrat personnalisé en *smart contract*. Nous n'envisagions pas dans ce projet l'étape consistant à générer l'intégralité d'un contrat automatiquement à partir de variables entrées par les utilisateurs. Seules certaines clauses – et non, l'intégralité du contrat – préalablement sélectionnées pour leur pertinence sont automatisées et traduites en langages informatiques. Accord Project fonctionne plus comme une librairie de clauses traduites en *smart contracts*, de même qu'envisagé dans le cadre de ce projet.

Les conséquences pour la réalisation de la librairie de *smart contracts*. Face à ces différents constats, il a été convenu de poursuivre la réalisation du projet en accentuant la dimension recherche qui la sous-tend. L'éclairage de ces exemples américains doit nous conduire à mesurer combien l'ambition est grande. **Il s'agit de créer la première plateforme de *smart contracts* en France et sur le continent européen.** Cette dimension recherche nous permet de conserver notre liberté de sélection des stipulations contractuelles qui présentent un intérêt pour les professionnels du droit et de choisir également en toute liberté le langage de programmation des *smart contracts* le plus efficace.

B. Inventaire des langages de programmation de *smart contracts*

La nécessité de choisir le langage de programmation. Cet inventaire permet à l'équipe de recherche d'apprécier la diversité des langages de programmation existante dans la programmation des applications de *blockchain* et d'en mesurer les qualités et les limites. La technologie évolue très rapidement et l'une des inquiétudes de l'équipe était d'inscrire la plateforme réalisée dans le temps. Un double souci d'efficacité et de longévité anime donc les membres de l'équipe. Cet inventaire a été réalisé par M. Abdoulaye Diallo, dans le cadre de son doctorat, encadré par Mmes Sihem Amer-Yahia et Amélie Favreau. Trois éléments ont été pris en compte : l'expressivité du langage, la sécurité du langage et la performance de la *blockchain*. Notons que dans le choix du langage l'on retrouve les impératifs de sécurité et de scalabilité qui ont été déterminants pour l'équipe dans le trilemme de la *blockchain*. Sur l'expressivité d'un langage, il s'agit d'apprécier la capacité créative que nous permet le langage informatique pour traduire des engagements contractuels. Une expressivité limitée laisse peu de champ à la traduction de clauses issues d'un contrat et à la finesse du raisonnement juridique qui les accompagnent. S'agissant de la sécurité, l'objectif est de limiter les bogues et les vulnérabilités que peuvent avoir certains langages de programmation. Même si les mises à jour de ces langages sont fréquentes et collaboratives, nous devons veiller à utiliser un langage de programmation qui ne soit pas faillible au risque de perdre la confiance des professionnels du droit dans l'outil. Enfin sur les performances de la *blockchain,* elles sont directement liées à la nécessité de scalabilité. Le tableau qui suit permet de représenter de manière synthétique ces impératifs.

Tableau 1. Inventaire des langages de programmation des *smart contracts* sur la *blockchain*

Plateformes	Langage(s)	Expressivité du langage	Sécurité du langage	Performance de la BC
Bitcoin	**Script - Ivy Lang** (ce dernier étant en développement)	**Script** : langage déclaratif non ***turing complete***[6] permettant la création de *smart contracts* très basiques (ex : transaction à l'occurrence d'une certaine signature) et ayant une syntaxe peu malléable. **Ivy Lang** : langage de programmation **déclaratif** qui se compile en script et permet de le manipuler plus aisément.	Les transactions étant limitées, elles limitent le champ d'apparition de bogues et de vulnérabilités. Script n'étant en fait qu'une suite d'instructions, le langage pour le peu de transactions qu'il permet, est très sécurisé. Même chose pour Ivy Lang qui consiste juste en une couche supérieure à Script pour écrire plus facilement du Script.	**Mécanisme de consensus** : *proof of work*[7]. 7 **transactions** par seconde aujourd'hui. Pourrait augmenter exponentiellement grâce au développement du *lightning network*[8] (solution permettant de créer des canaux d'échanges *off-chain*, qui ne nécessiteraient une inscription sur la *blockchain*

[6] Voir lexique p. 215.
[7] Un algorithme de consensus utilisé pour sécuriser les transactions sur la *blockchain* et éviter les doubles transactions. Voir lexique p. 215.
[8] Réseau décentralisé construit comme une application de deuxième couche adossée à la *blockchain* Bitcoin. Voir lexique p. 215.

				qu'au début et à la fin).
Ethereum	**Solidity - Vyper** (également LLL et Serpent, mais ils sont tombés en désuétude)	Les deux langages impératifs sont *turing complete* et permettent donc la traduction d'opérations complexes assez facilement.	**Solidity** : langage inspiré de C++ et JavaScript, connu pour être **vulnérable**. Comme il est orienté objet, il peut ne pas être très lisible et déclaratif, ce qui facilite l'apparition de bogues qui peuvent être aussi facilement exploités **Vyper** : inspiré de python, n'a pas encore passé l'épreuve du temps, mais le langage a été créé en réaction aux problèmes de Solidity. Sa syntaxe est épurée, des fonctionnalités orientées objet sont supprimées comme l'héritage pour permettre une	Mécanisme de consensus : PoW[9]. 15 transactions par seconde. Travail sur des protocoles comme Casper, Sharding qui sont d'autres noms pour des mécanismes similaires au *lightning network* et au PoS[10].

[9] Voir lexique p. 215.
[10] Voir lexique p. 215.

			meilleure **auditabilité** et déclarativité du programme.	
Hyperledger	Go - Java	Les deux langages sont haut niveau et *turing complete*, donc ils peuvent retranscrire facilement des opérations complexes. Cependant ce sont des langages tournés vers la programmation de grosses applications et donc plus bas niveau que la moyenne.	Ce sont des langages **orientés objet**, qui présentent donc, de manière inhérente, des défauts en termes de lisibilité/auditabilité et vulnérabilité. Mais ces langages sont très typés et bas niveau (**procéduraux**). Ils présentent donc un niveau de sécurité bien au-delà de la moyenne pour des langages orientés objet.	Pas de métriques sur la performance, mais le fonctionnement de la *blockchain*, notamment son architecture modulaire, permet l'optimisation de la scalabilité. On peut choisir ses valideurs et la manière dont ils peuvent valider les transactions. Donc le nombre de transactions par secondes est très variable.
EOS	Dérivé de C - C++	Les langages sont ***turing complete*** et très bas niveau, ils	C et C++ sont deux langages très bas niveau, très populaires	Utilise le DPoS (*Delegate Proof of Stake*)[11].

[11] Voir lexique p. 215.

		permettent de retranscrire toutes opérations complexes, mais assez difficilement en raison de leur syntaxe peu accessible.	pour programmer des applications nécessitant une attention accrue à la performance et la sécurité. Le code est plus explicite, mais très verbeux et impropre à une **auditabilité** aisée.	Actuellement 3000 transactions par seconde, mais à noter que cette scalabilité accrue par rapport à la concurrence se fait au détriment d'une grande centralisation (21 valideurs/ décideurs de la *blockchain*.
NEO	Langages actuels : C#, VB.Net, F#, Java, Kotlin, Python Langages à venir : C, C++, Golang, JavaScript	Tous ces langages sont ***turing complete*** donc permettent de retranscrire des transactions complexes, certains plus facilement que d'autres.	Les langages actuellement supportés par NEO pour écrire des *smart contracts* sont **soit bas niveau** et présentent les qualités et défauts déjà évoqués (performance, sécurité) **soit haut niveau** comme Python et présentent des avantages en termes de **lisibilité**/ **auditabilité**.	Mécanisme de consensus : *Delegated Byzantine Fault Tolerance* qui est une sorte Dpos de EOS (donc souci de centralisation). 1000 **transactions** par seconde.

Cardano	**Plutus**	***Turing complete***, très typé, purement **fonctionnel** permet de retranscrire des opérations complexes. Le fait que ce soit un langage fonctionnel le rend cependant moins malléable que des langages orientés objet.	Le langage Plutus est un langage inspiré de Haskell, qui est un langage fonctionnel par excellence. Il a été créé spécialement pour permettre une plus grande maîtrise et sécurité dans le développement des *smart contracts* et une invulnérabilité aux attaques.	Utilise le mécanisme PoS Ourobos. Potentiel 1000 **transactions** par seconde. Dernières évaluations : 250 transactions par seconde.
Tezos	**Michelson**	***Turing complete***, très typé, *domaine* spécifique[12], qui permet de retranscrire des transactions complexes. C'est un langage dit *domain* spécifique, c'est-à-dire fait exclusivement pour la plateforme qui le dessert, il ne ressemble	Le langage a été désigné exclusivement pour les *smart contracts* implémentant de la logique opérationnelle, il est fait pour permettre une meilleure lisibilité, auditabilité, et canaliser l'apparition de bogues et vulnérabilités.	Mécanisme de consensus : *Liquid proof of stake*, (ressemble au DPoS sauf que la délégation est optionnelle et non requise pour le fonctionnement de la *blockchain*) Apparemment traiterait plus de

[12] Voir lexique p. 215.

		donc à aucun autre langage et peutêtre assez difficile à prendre en main.		45 **transactions** actuellement, mais devrait en théorie pouvoir atteindre le millier.
Stellar	Les *smart contracts* sur **Stellar** peuvent être interrogés via une API[13] en utilisant le langage JavaScript.	Les *smart contracts* ne sont pas ***turing complete*** et le système ne permet que des transactions relativement basiques, mais courantes (double signature, séquestre, etc.).	Étant donné la limitation des *smart contracts* sur **Stellar**, la marge d'erreur et d'exploitation d'erreurs est moindre.	Mécanisme de **consensus** : *Stellar consensus protocol*. 1000 **transactions** par seconde

Choix du langage de programmation de la librairie. Nous avons fait le choix de travailler avec le langage Solidity. Ce langage impératif est *turing complete* et permet donc la traduction d'opérations complexes assez facilement**.** Il est inspiré de C++ et JavaScript et présente donc une facilité d'appropriation pour les développeurs de l'équipe. Son seul défaut est sa vulnérabilité aux bogues. Il sera donc nécessaire lors de la réalisation de la librairie de mettre en place des outils de vérification de la programmation pour assurer la sécurité des écritures informatiques. Son déploiement sur la *blockchain* Ethereum est aussi une garantie de la scalabilité.

C. Le passage par l'écriture en pseudo-code.

Justifications sur l'utilisation du pseudo-code. Lors de la réunion du 11 février 2019, les membres de l'équipe de recherche ont émis le souhait de voir la plateforme s'inscrire dans le

[13] *Application Programming Interface*. Voir lexique p. 215.

temps. La pérennité des langages de programmation présentés n'est pas assurée. En effet, ils dépendent des structures qui les portent. Sihem Amer-Yahia a alors proposé une étape supplémentaire dans l'écriture de la plateforme, qui consisterait à traduire les clauses en pseudo-code. En programmation, le pseudo-code, également appelé LDA (pour langage de description d'algorithmes) est une façon de décrire un algorithme en langage *presque naturel*, sans référence à un langage de programmation particulier. L'écriture en pseudo-code permet souvent de bien prendre toute la mesure de la difficulté de la mise en œuvre de l'algorithme et de développer une démarche structurée dans la construction de celui-ci. En effet, son aspect descriptif permet de décrire avec plus ou moins de détail l'algorithme, permettant de ce fait de commencer par une vision très large et de passer outre temporairement certains aspects complexes, ce que n'offre pas la programmation directe. Il n'existe pas de réelle convention pour le pseudo-code[14].

Explications sur l'écriture en pseudo-code. Le pseudo-code doit être complet. Il doit décrire l'intégralité de la logique de l'algorithme afin que la mise en œuvre devienne une tâche mécanique de traduction ligne par ligne en code source. Le pseudo-code est un récit pour quelqu'un qui connaît les exigences (domaine du problème) et qui essaie d'apprendre comment la solution est organisée. Chaque concepteur individuel peut avoir son propre style de pseudo-code. L'exercice n'est pas unifié. Le pseudo-code n'est pas une notation rigoureuse, puisqu'il est lu par d'autres personnes, et non par l'ordinateur. Il n'existe pas de « norme » universelle dans le secteur, il est le plus souvent utilisé à des fins pédagogiques[15].

Le plus souvent, un pseudo-code reprend des conditions structurées, notamment :

ALGORITHME	**PROCÉDURE**	**CONSTANTES**	**VARIABLES**	**DÉBUT**	**FIN**	**FONCTION**	**SI**	**ALORS**
SINON	**POUR**	**TANT_QUE**	**JUSQU'À**	**RÉPÉTER**	**SELON**	**AUTREMENT**		

Chacune de ces constructions peut être intégrée dans n'importe quelle autre construction. Ces constructions représentent la logique, ou le flux de contrôle dans un algorithme.

Les *smart contracts* de la librairie sont donc préalablement écrits en pseudo-code avant leur traduction en langage informatique.

[14] Source : https://fr.wikipedia.org/wiki/Pseudo-code.
[15] Source : https://users.csc.calpoly.edu/~jdalbey/SWE/pdl_std.html.

D. Les licences sur les langages de programmation de *smart contracts*

L'importance des licences sur les langages de programmation. Le choix d'avoir une plateforme en *open source* nous invite à être prudents sur les questions relatives aux licences. Même s'il existe peu de licences propriétaires dans le domaine des *smart contracts*, nous remarquons que les langages vers lesquels nous nous orientons sont sous licences GNU/GPL[16]. Il est donc important de vérifier les conditions de réutilisation de ces langages et fonctionnalités.

Licence propriétaire. Il n'existe pas de langage de programmation de *smart contracts* publié sous licence propriétaire, mais pour mettre en relief les aspects des différentes autres licences libres et *open source* sous lesquelles sont publiés les langages, nous nous prêterons à l'exercice d'analyser une licence propriétaire. Dans une licence propriétaire, il n'y a pas accès au code source d'un programme et l'utilisation, la modification et la redistribution sont limitées et strictement encadrées par un contrat.
Adapté à un langage de programmation, cela signifie qu'on ne peut l'utiliser qu'à des desseins limités (pour ne coder qu'un certain type d'applications), qu'on ne peut pas le partager pour l'améliorer ou le modifier par une communauté (ce qui est très préjudiciable pour faire évoluer un langage), ni revendre des programmes codés dans ce langage sans y être autorisé. Enfin, le régime de responsabilité est déterminé dans le contrat.

Licence MIT. La Licence MIT provient du Massachusetts Institute of Technology (MIT). C'est une licence *open source*, très courante dans le milieu et extrêmement permissive. Ce qui fait d'elle la licence quasiment par défaut sur GitHub. Elle autorise toute utilisation, modification et redistribution ; la seule obligation est de faire mention de la licence dans le logiciel. Elle prohibe l'usage du nom du logiciel pour promouvoir le produit créé à partir du contenu sous cette licence. Dans les langages de programmation de *smart contracts*, elle est la plus prisée puisqu'elle permet le mieux la mise en œuvre de la volonté des concepteurs de pousser à l'utilisation massive de leur langage.
En termes de responsabilité, elle exonère le créateur du logiciel ou du langage de toute faute. Le logiciel est dit livré "*as is*" : tel quel.

Licence BSD. La licence BSD (Berkeley Software Distribution License) est comme la licence MIT, une licence *open source* permissive imposant simplement la mention de la licence dans

[16] Licence publique générale qui fixe les conditions légales de distribution d'un logiciel libre du projet GNU. Voir lexique p. 215.

les logiciels et la clause d'exclusion de garantie. Elle prohibe également l'usage des éditeurs du logiciel pour promouvoir le produit créé à partir du contenu sous cette licence. Autrement dit, elle autorise l'utilisation, la modification et la redistribution libre du logiciel.

Licence publique générale (GNU/GPL). Elle est la licence libre par excellence. Elle a été créée par la Free Software Foundation et Richard Stallman. Les utilisateurs qui adhèrent aux termes et aux conditions de cette licence ont la permission de modifier, étudier et redistribuer le logiciel ou un logiciel dérivé. En somme, elle permet une utilisation très permissive quant à l'utilisation et la modification. Mais le droit de redistribuer est garanti seulement si l'utilisateur fournit le code source de la version modifiée. En outre, les copies distribuées, incluant les modifications, doivent être aussi sous les termes de la GPL. C'est l'effet « *copyleft* » abordé précédemment. La volonté derrière cette licence est de contraindre les utilisateurs à garantir la même liberté dont ils ont joui en utilisant le logiciel.
Dans le contexte d'un langage de programmation, elle force les utilisateurs à partager toute modification qu'ils feraient du langage, pas nécessairement le code de leur programme, mais si une entreprise souhaitait faire une nouvelle version du langage pour l'améliorer elle serait contrainte de partager tous ses éléments.

Tableaux 2 : Récapitulatif des licences portant sur les langages de programmation

	Licence propriétaire	Licence GNU/GPL	Licence BSD	Licence MIT
Langages de développement des *smart contracts* utilisant cette licence	Aucun	Solidity	Go	Script - Ivy (Bitcoin) Vyper (Solidity) VB. NET (NEO) Python (NEO)

				Plutus (Cardano) Michelson (Tezos)
Accès au code source	Non, seulement au code objet	Oui	Oui	Oui
Restriction d'utilisation	Dépend du contrat	Non	Non	Non
Droit de modification	Non	Oui	Oui	Oui
Droit de distribution	Non	Obligation de fournir le code source et d'informer des modifications et de leur date Distribution aux conditions de la licence GNU/GPL	Oui	Oui
Responsabilité	Dépend du contrat	Non, mais sous réserve de la loi applicable	Exclusion. Langage livré "*as is*"	Exclusion. Langage livré "*as is*"

Section 2 : Choix juridiques pour la programmation des *smart contracts*

Sélection de clauses juridiques traduites en *smart contracts*. Toutes les clauses ne sont pas susceptibles d'être traduites en *smart contracts*. Après quelques remarques générales sur les clauses que nous ne pouvons pas traduire en *smart contracts* (§1), l'équipe a procédé à une sélection (§2).

Paragraphe 1 : Observations générales

Plan. Après avoir circonscrit la notion de *smart contract* (A), l'équipe s'est attachée à fixer des limites juridiques aux *smart contracts* (B) avant d'expliquer la méthode retenue (C).

A. Précisions terminologiques

Rappel sur la notion de *smart contract*. Le *smart contract* n'est pas un contrat, mais une simple modalité d'exécution du contrat. Nick Szabo lui-même a fait ce constat[17]. Derrière le *smart contract* qui s'apparente à une assistance technique, il y a un « vrai contrat ». En effet, le *smart contract* n'est ni un contrat, ni intelligent, cela a été rappelé par Mustapha Mekki, lors du séminaire introductif de la recherche[18]. Dit autrement, le contrat précède le *smart contract* qui n'est qu'un outil informatique intéressant son exécution, un contrat à proprement parler en est le support, « c'est la réalisation technique d'un contrat déjà formé »[19]. Par analogie, le contrat n'est pas tant le document papier : *instrumentum*, que l'acte juridique : *negotium*. Il en va de même pour le *smart contract* qui ne se confond pas entièrement avec le contrat. Il serait un *instrumentum* technique.

[17] N. SZABO, « Smart contracts : Formalizing and Securing Public Networks », *First Monday*, sept. 1997, n° 9.
[18] M. MEKKI, « Le contrat, objet des smart contracts », *Dalloz IP/IT*, 2018, p. 409.
[19] A. TOUATI, « Tous les contrats ne peuvent pas être des smart contracts », *RLDC*, 2017, p. 6302 ; v. également : Enguerrand Marique, « Les smart contracts en Belgique : une destruction utopique du besoin de confiance », *Dalloz IP/IT*, 2019, p. 22.

***Smart contract* et exécution du contrat.** Fabien Gillioz indique dans sa contribution lors du séminaire introductif : « en l'état actuel de la législation et de la technologie, un *smart contract* est plus approprié comme mécanisme d'exécution pour un ensemble d'obligations déterminées, plutôt qu'en tant que contrat en soi »[20]. Les *smart contracts* permettent d'automatiser l'exécution de certaines clauses ou contrats. Ils peuvent aussi servir à automatiser les sanctions de l'inexécution d'obligations contractuelles. La phase de conclusion du contrat peut sembler en apparence quelque peu délaissée par cette étude. Mais ce n'est pas tant la conclusion du contrat en soi (le *smart contract* suppose tout de même un contrat) que certaines étapes de la phase de conclusion qui sont en jeu : la remise de documents, des délais de réflexion, la constitution de garantie, les étapes d'achèvement d'une création intellectuelle, etc. Sur un thème voisin, le *smart contract* associé à la *blockchain* est aussi un moyen de preuve que ces étapes de formation ont été réalisées dans les temps.

B. Limites juridiques aux *smart contracts*

Exclusion des règles d'ordre public. L'équipe a fait le choix d'exclure de la traduction en *smart contracts* les règles issues de l'ordre public de protection et de l'ordre public de direction. Elles sont plus ou moins spéciales et sectorielles ce qui peut exclure :

- soit seulement certaines clauses,
- soit certaines catégories de contrat, notamment en raison de la situation des parties contractantes (salarié, consommateur…).

Ces points sont à l'esprit de chacun, et participent aux choix des clauses au cas par cas. Maître Charbonnel particulièrement sensible à cette question propose d'écarter le bail d'habitation.

L'application du droit commun des contrats. Rappelons que le droit commun des contrats s'applique aux *smart contracts*. Il comporte en son sein des dispositions supplétives et d'autres impératives. Ce point est assez peu développé en doctrine, mais traité par Fabien Gillioz[21]. On pense à la mise en jeu des règles de consentement et de capacité. Le groupe s'est notamment interrogé sur l'incapacité d'une personne partie à un *smart contract* ou sur l'hypothèse d'un manque de pouvoir ou d'habilitation (le contrat conclu par un interdit de commerce, par un dirigeant sur les prérogatives propres aux associés, etc.). C'est peut-être là une contrainte plus

[20] F. GILLIOZ, « Du contrat intelligent au contrat juridique intelligent », *Dalloz IP/IT*, 2019, p. 16.
[21] F. GILLIOZ, « Du contrat intelligent au contrat juridique intelligent », *Dalloz IP/IT*, *op. cit.*

délicate à apprécier encore que l'incapacité des personnes mineures. Cette réserve levée, il demeure que si le *smart contract* est affecté d'une telle limite, c'est que le contrat principal, dont il n'est que la traduction informatique, l'est aussi. Le droit possède les ressorts pour le traitement juridique de ces situations. Il faut donc considérer le *smart contract* comme **un accessoire**, le contrat traditionnel étant **le principal**, le droit commun des contrats affectant l'existence du *smart contract* qui lui est attaché.

Perspective pour l'encadrement juridique du *smart contract.* Le groupe insiste sur la nécessité d'approfondir trois principales réflexions.

- Le risque lié à la disparition du tiers de confiance, et à l'intermédiation de l'oracle qui fait le lien entre la *blockchain* et le monde réel. Que se passe-t-il si l'information décisive n'est pas ou mal transmise, etc. ?
- La question soulevée par Fabien Gillioz de l'immuabilité du contrat. Elle est contraire au droit, français du moins. Ce que les parties ont convenu, elles peuvent le rompre ou le modifier. Or, la *blockchain* fonctionne comme un registre, il ne s'agit pas tant de modifier ou supprimer un élément (ce qui serait illusoire), mais d'ajouter des correctifs. Il a donc été souligné la nécessité d'intégrer dans la programmation informatique du *smart contract* la possibilité d'un « retour », or cette possibilité technique peut parfaitement être mise en œuvre avec la fonction de Controller abordée précédemment.
- Les dysfonctionnements**,** bogues, erreurs de codage et de conception appellent une mise en garde sur la fidélité du code informatique du *smart contract* aux engagements contractuels souscrits dans le monde réel. On pourrait dégager un principe de « fidélité » du *smart contract* au contrat juridique. Le *smart contract* devrait être « homothétique » au contrat juridique.

Il est enfin important de souligner que le contrat est un outil d'anticipation, ce qui n'est pas pour l'heure le cas du *smart contract*. L'anticipation ne se résume pas à une clause d'indexation du prix. On peut ajouter dans cet esprit que si une clause de « revoyure », renégociation, imprévision pourrait être déclenchée automatiquement en fonction de certains seuils et paramètres, il semble aller de soi que la négociation (notamment à travers une conduite de bonne foi) qui en résulte échappe à l'automatisation.

C. Méthode proposée : les clauses comme point de départ

Pour parvenir à la réalisation de la librairie, le groupe a fait le choix d'une méthode pour la présentation des clauses sur la plateforme (1) et de critères de sélection des clauses (2).

1. Méthode pour la présentation des clauses sur la plateforme

Du contrat à la clause. Le groupe s'est interrogé en premier lieu sur la méthode de sélection des clauses. Comment opérer un classement face à la diversité des clauses ? Faut-il raisonner et présenter la réflexion par *clause* ou par *contrat* ? Quel est le point de départ : la clause ou le contrat ?

Pour une partie de l'équipe de recherche, il est pertinent de raisonner *par contrat*. En effet, la question ne paraît pas forcément, *a priori*, se poser simplement à l'égard de clauses ; mais plus généralement eu égard à certains contrats, pour lesquels le *smart contract* peut être un outil utile, ou au contraire peu pertinent. Pour une autre partie de l'équipe de recherche, évaluer l'intérêt du *smart contract* invite à raisonner au final *par clause*. En effet, il semble essentiel de ne pas généraliser et de raisonner systématiquement au cas par cas, en fonction de la nature du contrat, de son objet et surtout des clauses elles-mêmes qui comportent souvent des réserves et exceptions. Ce seront donc les clauses qui seront appréciées au cas par cas.

Travail à partir de l'ouvrage de technique contractuelle de Jean-Marc Mousseron[22]. L'ouvrage reprend « dans l'ordre » quasiment chronologique les clauses d'un contrat d'affaires « typique » et donne des exemples de rédaction et des encarts thématiques. Il s'agit d'un plan de travail pertinent et pédagogique qui a été adopté par les autres membres du groupe de recherche.

Titre I. Dispositions communes initiales
I. Titre
II. Préambule
III. Définitions
IV. Système juridique applicable au contrat

[22] Selon l'ouvrage J.-M. Mousseron *et al.*, *Technique contractuelle*, 5e éd., Paris, Francis Lefebvre, 2017. Le plan de l'ouvrage est davantage détaillé.

Titre II. Dispositions relatives à la naissance des relations contractuelles

I. Les partenaires

A – Les tiers

B – Les parties

II. L'opération

A – Le principal de l'opération

B – Les modalités (rétractation – conditions)

III. La Chose ou le service

IV. Le prix

A – Fixation initiale

B – Correctifs (indexation – révision)

V. Le territoire

VI. La durée

A – Définition de la durée initiale (termes – types de durées)

B – La prolongation (prorogation – reconduction)

Titre III. Dispositions relatives à l'existence des relations contractuelles

I. Effets du contrat

A – Effets personnels (dont obligation de paiement)

B – Effets réels (conditions suspensives – transfert des risques)

II. Circulation du contrat (cession de contrat – agrément – sanction)

Titre IV. Dispositions relatives à l'extinction des relations contractuelles

I. Les causes de l'extinction du contrat (cause – le terme extinctif – l'anéantissement rétroactif)

II. Les effets de l'extinction du contrat (effet rétroactif ou non – autres obligations : positives ou négatives)

Titre V. Dispositions communes terminales

I. Les clauses d'adaptation

II. Les clauses d'interprétation

III. Les clauses d'avenant

IV. Les clauses de différend

V. Les clauses de preuve

VI. Les clauses de conservation et de mise en œuvre

VII. Les signatures

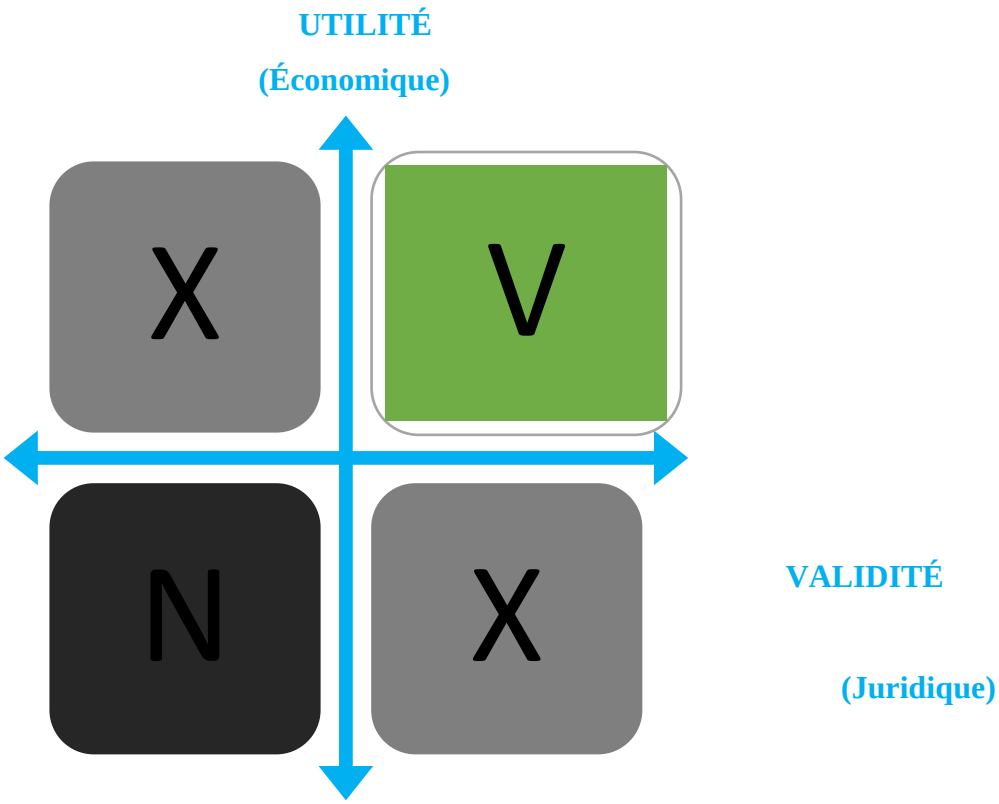
Titre VI. AUTRES

2. Critères pour la sélection des clauses sur la plateforme

À la recherche de critères discriminants. L'intérêt du recours au *smart contract* est lié à l'opportunité et l'efficacité de cet outil. Quelques pistes sont proposées :

- mettre en œuvre le *smart contract* seulement dans des situations d'une grande simplicité, dans laquelle la question de l'exécution des obligations des parties, ou au moins de celles du débiteur ne laisse place à aucune subjectivité dans l'appréciation.
- traduire les clauses opérationnelles, c'est-à-dire celles qui requièrent une action déterminée en fonction de l'avènement d'une condition spécifique, ou d'une période de temps, et laisser en suspens les clauses non opérationnelles, à savoir celles qui se réfèrent à des obligations qui ne sont pas déterminées ou qui n'ont pas de logique conditionnelle, telle que les clauses de for, clauses de juridiction, clauses de confidentialité, clauses d'intégralité du contrat.

La méthode adoptée. Il a été proposé de classer les clauses en fonction de deux critères : l'utilité et la validité. L'utilité correspond à l'intérêt économique, pratique. Cela présente l'avantage d'être universel. La validité correspond à la faisabilité juridique, la sécurité juridique. L'appréciation est différenciée selon les régimes juridiques. Cette présentation est schématique : on aura parfois du mal à faire la part des choses entre le manque de pertinence et la licéité d'un *smart contract*. Quant à la zone verte, elle n'offre pas un blanc-seing, le passage en zone grise peut être brutal !



Paragraphe 2 : Sélection des clauses

Le travail de sélection des clauses est particulièrement délicat. Les choix opérés ont une incidence directe sur la réalisation de la librairie et les clauses déposées sur la plateforme.

A. Clauses non éligibles au *smart contract* (zone noire)

Selon la présentation schématique précédente, il est possible de rassembler les clauses non éligibles aux *smart contracts*. Elles sont utiles mais non-valides, non-utiles mais valides, non-utiles et non-valides.

1. Les clauses non utiles, écartées de la « smart contractualisation »

Distinction « condition contractuelle » ou « termes contractuels ». Les notions de « condition contractuelle » ou « termes contractuels » sont ambiguës. En effet, « Conditions » et « Termes » peuvent désigner deux réalités différentes. Il peut s'agit d'une stipulation contractuelle, d'une simple clause d'un contrat, cela peut renvoyer à un mécanisme conditionnel

au sens d'obligation conditionnelle ou à une obligation à terme. Le *smart contract* nous apparaît comme le domaine de prédilection des « obligations conditionnelles » ou « à terme » et ne pas concerner toutes les conditions ou tous les termes d'un contrat. D'autres clauses n'ont qu'un but informatif, descriptif d'un cadre juridique, comme le rappel du droit commun applicable ou des dispositions impératives (respect du droit moral de l'auteur, désignation du juge conforme au droit applicable, ou de nombreuses clauses des statuts de société…). Leur place dans un contrat est parfois d'ordre purement pédagogique. Certaines accordent un droit, une faculté, à l'une des parties et ne sont pas pour autant pertinentes pour le projet. En effet, elles autorisent un créancier, mais ne supposent pas de prestation, d'action ou d'abstention du débiteur. Cette catégorie (clauses non utiles) semble avoir une portée universelle dans le sens où elle s'impose par la force des choses et ne dépend toujours pas des spécificités d'un système juridique. **Elles ne sont donc pas susceptibles d'être exécutées automatiquement**.

2. Les clauses non valides, écartées de la « smart contractualisation »

Le *smart contract* face à l'indétermination ou la complexité. L'indétermination de la règle (le standard), les notions de standard ou notions à contenu variable, qui sont de plus en plus fréquentes dans notre droit[23], en sont l'une des illustrations. Ces notions font référence au raisonnable, au grave, à la bonne foi, etc., et leur variabilité les exclut *a priori* du *smart contract*, sauf à faire l'objet d'une objectivisation qui trouvera toujours comme limite l'appréciation du juge.

Par exemple :

- Sous influence européenne, le droit de la consommation libéralise certaines pratiques autrefois encadrées dans le détail en posant la limite de la pratique « déloyale » (ex. loteries, ventes liées…).
- L'existence d'une contrefaçon ne se limite pas au constat de la reproduction ou de la représentation d'une œuvre sans autorisation car elle peut être couverte par une exception au droit d'auteur, qui fait référence à des standards « flous » : « les lois du genre » en matière de caricature, « la courte citation » qui n'est pas quantifiée par le Code de la propriété intellectuelle, « le but d'information », etc.

[23] Cf. Colloque organisé par le CUERPI, sous la direction de J.-M. Bruguière sur « Les standards en propriété intellectuelle », publié par Dalloz dans la collection Thèmes et commentaires en 2019.

La complexité des faits. D'autres clauses ne sont pas pertinentes en raison de la subjectivité ou de la complexité des conditions de déclenchement. Ce sont les faits qui sont complexes et nécessitent une appréciation : réception de travaux, agrément d'une œuvre commandée…
Selon Lionel Charbonnel, « dès lors que la question de l'exécution d'une obligation paraît devoir donner lieu à une appréciation, le *smart contract* me paraît plus difficile à mettre en œuvre. Le *smart contract* ne me paraît dès lors pas pertinent dans la mesure où le juge dispose en pratique, *in fine*, d'un pouvoir d'appréciation. Le *smart contract* doit donc porter sur une obligation dont l'exécution est simple à apprécier ; cette condition n'étant pas suffisante ».

Le *smart contract* face à l'ordre public de protection et de direction. Afin de pouvoir envisager une exécution automatique, encore faut-il que la matière concernée ne soit pas marquée de dispositions d'ordre public l'interdisant. Or le droit peut protéger la partie faible ou placée ponctuellement en position de faiblesse. De nombreux exemples existent en droit français : délais de grâce du juge, rupture brutale de relations commerciales établies, rupture abusive de crédit. Le droit du travail et le droit de la consommation en sont des domaines sensibles, lorsque la protection de la partie faible est en jeu. Toutefois, le droit de rétractation du consommateur, l'indemnisation retard du transport de voyageurs sont des cas dans lesquels le *smart contract* mis au service de la partie faible peut être utile. Concernant l'ordre public de direction, il s'agit notamment du conflit avec le droit des procédures collectives (suspension d'un paiement ou rupture d'un contrat dont la continuation a été autorisée). *A priori*, la catégorie des clauses non valides rassemble les dispositions qui ne seront pas l'objet d'un *smart contract* du fait de leur indétermination, de leur complexité ou de leur soumission aux dispositions d'ordre public de protection ou de direction.

3. Les clauses exclues

Ces clauses s'examinent à partir du plan de l'ouvrage précité de Jean-Marc Mousseron.

Titre I. Dispositions communes initiales. Au rang des dispositions initiales, l'équipe a conclu que des dispositions du préambule, des définitions ou encore des clauses relatives au système juridique applicable ne sont pas susceptibles d'être traduites en *smart contract*.

Titre II. Dispositions relatives à la naissance des relations contractuelles. Concernant le Titre II de cet ouvrage de technique contractuelle, l'équipe a convenu, au regard des parties au

contrat, qu'il était peu pertinent d'automatiser une relation de co-traitance, de coordination ou encore une exécution testamentaire sur une *blockchain*. Ce sont des relations entre les parties basées sur un *intuitu personae,* où la *blockchain* dans ses caractéristiques d'immutabilité, traçabilité, sécurité, intégrité, confidentialité, authenticité ou encore le caractère auditable des transactions réalisées ne présente pas d'intérêt.

Titre III. Dispositions relatives à l'existence des relations contractuelles. Le Titre III de l'ouvrage a soulevé des discussions tant sur les effets personnels du contrat que sur les effets réels.

- Sur les effets personnels du contrat. Si les éléments relatifs au calendrier ou à la fixation des objectifs par avance sont susceptibles d'être programmables simplement, cela ne présente pas beaucoup d'intérêt de les automatiser sur une *blockchain*. À l'inverse, des obligations de confidentialité ou d'information se sont avérées trop complexes à traduire sous la forme d'un programme informatique. Pour cette raison également, l'équipe a fait le choix d'exclure les clauses relatives à la maintenance ou aux obligations de mise en garde. Relativement aux effets du contrat toujours, la question de la renonciation à l'exécution est fortement liée à l'intégralité du contrat et nous avons fait le choix d'une traduction par clause et non par contrat pour la librairie. Quant aux remèdes aux perturbations, comme l'exception d'inexécution, les délais de paiement ou les mises en demeure, il s'agit de clauses dont l'appréciation subjective de la mise en œuvre empêche une automatisation. La conclusion est identique lorsque les perturbations sont définitives, comme dans les hypothèses de garantie ou de responsabilité. La mise en œuvre de la garantie dans un contrat de vente impose que le défaut soit analysé ou prouvé avant que la partie puisse être actionnée, ce qui se fait à l'extérieur d'une *blockchain*. De même, pour le remboursement du prix d'un produit en cas d'action en garantie, il n'est pas possible d'automatiser un remboursement alors qu'en général le contrat prévoit que la garantie sera exclue si, après vérification par le fournisseur, le défaut trouve son origine dans une mauvaise utilisation du produit, négligence, accident ou défaut d'entretien de la part du client, ou d'un tiers, comme en cas d'usure normale du produit, ou encore de force majeure. Ajoutons qu'il paraît difficile d'automatiser l'engagement de garantie d'éviction prévue dans un contrat et applicable en cas de contrefaçon. Par exemple, le prestataire ne peut s'engager à payer toutes les sommes, quelle que soit leur nature (dommages et intérêts, frais, amende, etc.), qui seraient

réclamées par un tiers au client et ayant pour fondement une contrefaçon. Pour des raisons identiques, les clauses de force majeure ou de garantie de conformité ne peuvent pas être traduites en *smart contracts*. Enfin, il est difficile d'envisager un *smart contract* en matière de responsabilité et d'indemnisation du préjudice, car les clauses contractuelles prévoient souvent des réserves et de nombreux cas d'exclusion. Par ailleurs, tout dépendra du niveau d'engagement contractuel souscrit (obligation de moyen / obligation de résultat).

- Sur les effets réels. Le transfert de risque ne peut pas faire l'objet d'un *smart contract*, car la survenance d'un risque doit être analysée (faits et droit) et il sera difficile de ce fait de l'automatiser.

Titre IV. Dispositions relatives à l'extinction des relations contractuelles. L'extinction d'un contrat par un *smart contract* est une fonctionnalité programmable, mais inopportune ou impossible juridiquement. Par exemple, la résolution du bail pour non-paiement, qui pourrait être techniquement facile à mettre en place en cas de non-paiement, ne peut être légalement envisagée, le contrôle préalable du juge étant légalement indispensable. Il en va de même de la résiliation pour manquement par l'une ou l'autre des parties à l'une de ses obligations, au titre du contrat-cadre ou d'un bon de commande, l'autre partie étant autorisée, 30 jours après mise en demeure envoyée par lettre recommandée avec avis de réception restée sans effet, à mettre fin au contrat de plein droit sous réserve de tous dommages et intérêts auxquels elle pourrait prétendre du fait des manquements invoqués. Depuis la réforme du droit des obligations, un contrôle judiciaire est réalisé *a posteriori*. Le risque demeure et l'automatisation est déconseillée.

Titre V. Dispositions communes terminales. Qu'il s'agisse des clauses d'adaptation ou des clauses d'interprétation, elles sont difficiles à traduire sous la forme de *smart contracts*. Il en est de même pour les documents contractuels ou les clauses d'avenant. Si la *blockchain* peut présenter un intérêt pour la conservation des éléments, il n'est pas pertinent de traduire une clause en *smart contract* pour l'organiser.

B. Clauses éligibles au *smart contract* (zone verte)

C'est le *smart contract* utile et valide ! Celui qui fera l'objet d'une implémentation sur la plateforme que nous souhaitons réaliser. Les contributeurs de ce groupe ont donné leur accord de principe à certaines clauses, en émettant parfois des réserves ou des conditions.

1. Observations intermédiaires

Clauses pertinentes pour une traduction en *smart contract*. Les clauses pertinentes pour la mise en œuvre par *smart contract* sont, par définition, celles dont l'exécution automatisée est possible, utile, licite. Pour cela rappelons que les conditions de déclenchement doivent être objectives, simples, sans contrôle judiciaire *a priori* et en l'absence de règles d'ordre public contrariant l'exécution automatique. Il est apparu également que les clauses sur l'existence d'une sanction juridique ou d'un contrôle sont susceptibles de faire l'objet d'un *smart contract a posteriori* et non *a priori*. Tel est le cas par exemple de la garantie autonome mais non de l'astreinte.

2.Clauses

Titre II. Dispositions relatives à la naissance des relations contractuelles. Étant parvenue à la conclusion qu'aucune disposition initiale ne pouvait être traduite en *smart contract*, l'équipe s'est attachée aux dispositions relatives à la naissance des relations contractuelles.

Sur l'opération, nous avons travaillé principalement sur **les pactes d'actionnaires**. En effet, la titrisation possible sous la forme de *tokens* rend la valeur circulable et le *smart contract* pertinent. Les **clauses d'option d'achat** peuvent être intéressantes à traduire en *smart contract*. En effet, certains cas pouvant déclencher une option d'achat pour les autres actionnaires peuvent être automatisés (notamment, décès, faillite, fin d'un contrat de travail, un actionnaire n'est plus le bénéficiaire économique d'un certain pourcentage de ses actions). Le **droit de préemption** à travers le processus d'émission de nouvelles actions peut être automatisé de sorte à enclencher automatiquement une requête aux actionnaires qui peuvent exercer leur droit de préemption ou non. Pour **le droit de suite** (*Tag-along*), si un actionnaire souhaite transférer des actions, une annonce peut être générée automatiquement aux autres actionnaires qui peuvent décider de vendre également leurs actions ou non selon les modalités de la clause. Il en va de même pour **l'obligation de suite** (*Drag-along*) : si un actionnaire ou un groupe d'actionnaires

qui détient plus d'un certain pourcentage d'actions décide de transférer ses actions, la vente de toutes les actions est autorisée par le *smart contract* selon les termes de la clause et une notification est faite aux autres actionnaires pour les informer du cas de *Drag-along*.

Sur les modalités (rétractation ou conditions), la **condition résolutoire**, notamment le droit de rétractation du consommateur, pourrait se gérer dans un *smart contract* – envoi de la notification par le consommateur et remboursement automatique de sa commande à réception du produit par le vendeur. De même, les **clauses de dédit ou d'arrhes** sont adaptables pour les *smart contracts*.

Sur le prix et plus précisément ses correctifs (indexation ou révision), il serait possible et utile d'automatiser **des clauses de remises de fin d'année** (RFA), qui peuvent être accordées à un client en fonction de paliers de chiffres d'affaires annuels encaissés par un prestataire au titre de la vente de produits ou services.

Ex. : Si le volume de chiffres d'affaires annuel encaissé l'année civile N atteint l'une des tranches visées, la remise appliquée sur le chiffre d'affaires encaissé issu des produits facturés au client sera calculée au cours du premier trimestre de l'année civile N+1. Attention toutefois, car, l'application des RFA est souvent conditionnée au respect des obligations contractuelles et n'est donc pas automatique. En matière de distribution, le *smart contract* pourrait être envisagé pour automatiser la révision des prix dans les contrats d'une certaine durée (annuelle, mensuelle). Par exemple, les prix des prestations sont révisés automatiquement le 1er janvier de chaque année et pour la première fois le 1er janvier de l'année suivant la signature du contrat, par application de la formule de révision suivante :

Pn = Po x (Sn/So)

dans laquelle :

Pn = représente les prix recalculés et applicables pour l'année

Po = représente les prix initiaux

Sn = représente le dernier indice Syntec connu au jour de la révision des prix

So = représente l'indice Syntec connu au jour de l'entrée en vigueur du contrat.

Titre III. Dispositions relatives à l'existence des relations contractuelles. Ce sont les dispositions qui présentent le plus d'intérêt pour les *smart contracts*.

- **Sur les effets personnels**

La **remise de documents obligatoires** peut être automatisée avec un *smart contract*. Le prestataire est souvent contraint de communiquer à son client certaines informations sociales au cours de l'exécution du contrat. Ces communications pourraient peut-être être faites automatiquement dans un *smart contract*. Ex. : conformément aux articles L. 8221-1 et suivants et D. 8222-4 et suivants du Code du travail relatifs au travail dissimulé, le prestataire s'engage à remettre au client, à la signature des présentes, puis automatiquement tous les six mois, les attestations et avis requis par l'article D. 8222-5, 1° et 2° du Code du travail. Le prestataire s'engage également, conformément à l'article D. 8222-5 3° du même code, à remettre au client, à la signature des présentes, puis automatiquement tous les six mois, une attestation sur l'honneur certifiant que chacun de ses salariés qui participera à l'exécution des prestations, sera employé régulièrement au regard des articles L. 1221-10, L. 3243-2, et R. 3243-1 du Code du travail. Dans les pactes d'actionnaires, certaines notifications peuvent également être traduites en *smart contracts*. Par exemple, la clause Communication : « Toute communication à une des parties à la présente convention devra être faite par lettre recommandée envoyée à l'adresse indiquée au regard de son nom au début de la présente convention ».

Ce sont bien évidemment les clauses relatives au **paiement** qui sont susceptibles d'être traduites en *smart contract*. Il est possible d'automatiser le processus de transfert du prix dès la réalisation de la condition (signature du contrat, livraison, etc.). Dans un contrat de service, il peut s'agir du paiement d'une somme déterminée donnant accès à un bien. Ce sont les contrats dans lesquels le paiement donne accès à un service « standardisé », tel l'accès à une chambre d'hôtel, le paiement permettant l'ouverture automatique de la porte d'une chambre ; l'absence de renouvellement du paiement rendant l'accès au contraire impossible, par exemple par la désactivation du badge donnant accès à une salle de sport. Certaines limites sont à conserver, telles que la protection de la vie privée et du domicile ainsi que de ne pas donner un droit de rétention de fait. Selon les services, cette automatisation pourra s'avérer complexe par exemple lorsqu'un cas de mauvaise exécution requiert une analyse particulière par les parties. Plus simplement, dans le cadre d'une cession d'actions, le transfert de fonds peut être automatisé (y compris sur un compte à l'extérieur de la chaîne). Dans un contrat de prêt, la survenance et l'exigibilité des intérêts peuvent être automatisées. Exemple de clause « Intérêts » : « Le prêt porte intérêt au taux annuel de [□] %. Les intérêts commencent à courir dès le versement du montant du prêt. Les intérêts seront payables au prêteur semestriellement les [□] et [□] de

chaque année, la première fois le [□], au lieu indiqué par le prêteur. Le prêteur se réserve le droit d'adapter en tout temps le taux d'intérêt à son entière discrétion moyennant un préavis de six semaines ». De même, le paiement d'un amortissement dû semestriellement par exemple peut être automatisé. Exemple de clause « Amortissement » : « Un montant en euros [□] sera payable au prêteur semestriellement les [□] et [□] de chaque année, la première fois le [□], au lieu indiqué par le prêteur ; ce montant sert à l'amortissement de l'emprunt ».

Concernant **le séquestre et la libération de fonds**, une somme due en contrepartie de l'exécution d'une obligation contractuelle peut faire l'objet d'un séquestre sur un *smart contract* et être libérée une fois l'obligation exécutée. Par exemple, le client s'engage à verser dès la signature du contrat la somme de 500 euros. Cette somme sera séquestrée et libérée au vendeur dès réception par le client de son produit.

Les remèdes aux perturbations (retard d'exécution) peuvent faire l'objet d'un *smart contract*.

Tel est le cas des **pénalités en cas de non-respect des délais impératifs**. L'application de pénalités en cas de non-respect de délais impératifs (SLA) pourrait se gérer par un *smart contract* à condition que le contrat prévoie bien une obligation de résultat quantifiée et des délais précis. Toutefois, les clauses de pénalités comportent souvent les réserves suivantes : sauf cas de force majeure ou manquement du client directement à l'origine du non-respect du délai. Or le *smart contract* ne peut pas gérer ces réserves.

L'on pourrait également envisager un *smart contract* pour organiser les **pénalités en matière de maintenance corrective.** Ainsi, la clause suivante pourrait être traduite en langage de programmation :

« Tout manquement du prestataire à ses obligations au titre de la maintenance corrective entraîne l'application du seul fait de la constatation du manquement, sans mise en demeure préalable, sans autres formalités judiciaires, d'une pénalité calculée comme suit :

- pour les anomalies bloquantes : le dépassement du délai prévu au titre de la résolution de l'anomalie incrémente le compteur de pénalité de 2 unités par jour ouvré de retard ;
- pour les anomalies majeures : le dépassement du délai prévu au titre de la résolution de l'anomalie incrémente le compteur de pénalité de 1 unité par jour ouvré de retard ;
- pour les anomalies mineures : le dépassement du délai prévu au titre de la résolution de l'anomalie incrémente le compteur de pénalité de 1 unité tous les 2 jours ouvrés de retard.

Lors du bilan trimestriel, le client sera en droit d'exiger le paiement par le prestataire d'une somme de 100 euros hors taxe par unité comptabilisée sur la période.

Le montant total des pénalités de retard par an dues est plafonné à 20% du montant annuel HT des prestations de maintenance forfaitaire ».

Un autre exemple de clause :
« En cas de non-respect de ses obligations, le fournisseur s'expose aux pénalités dans les cas suivants :

- en cas de retard :

Le fournisseur s'engage à respecter les délais de livraison indiqués dans les bons de commande. À défaut le client pourra appliquer au fournisseur des pénalités calculées selon la formule suivante :
P = (M x R)/1.000
P = Pénalité
M = Montant de la commande
R = Nombre de jours de retard.

- en cas de défaut relevé au moment de la livraison :

Le fournisseur dispose de 5 jours ouvrés pour livrer au client ou à son client final les produits sans défaut. Au-delà de ce délai, le client pourra appliquer des pénalités de retard calculé conformément à la formule ci-dessus.

Tel est le cas également des pénalités en cas de retard de paiement. Le non-paiement à l'échéance peut entraîner l'application d'intérêts de retard contractuels automatique. Le versement de ces intérêts pourrait être géré dans un *smart contract*. Par exemple, « en cas de retard de paiement, total ou partiel, le prestataire se réserve le droit de réclamer le versement d'intérêts de retard dont le montant sera calculé par application d'un taux d'intérêt fixé à 3 fois le taux de l'intérêt légal en vigueur ».

Sur les perturbations définitives (responsabilité, garantie), la question de la clause pénale est importante. Si la clause pénale est suffisamment précise quant à la nature des manquements qui légitiment son application, son versement pourrait être automatisé dans un *smart contract*. L'exemple suivant de clause pénale a été proposé.
Conformément au règlement européen n° 853/2004 du 29 avril 2004 fixant des règles spécifiques d'hygiène applicables aux denrées alimentaires d'origine animale, le grossiste s'engage à acheminer la viande hachée au restaurateur dans un camion frigorifié dont la température intérieure ne dépasse pas 2 degrés Celsius. Si la température de l'intérieur du

camion frigorifié du grossiste venait à dépasser 2 degrés Celsius, le grossiste indemnise le restaurateur d'un éther, unité de monnaie virtuelle de la *blockchain* Ethereum (art. 1235-1 alinéa 2). Cette faculté peut-elle être codée dans un *smart contract* ? L'avantage serait qu'au lieu de transférer les sommes à la fin de chaque « tour », il serait possible de séquestrer le montant des indemnités et ne permettre leur délivrance que si le grossiste et le restaurateur étaient d'accord. En cas de non-consensus, la somme resterait séquestrée et ne pourrait être délivrée que par un tiers désigné (on peut imaginer un huissier de justice). Toutefois, est-ce dans la nature de la clause pénale que de prévoir un séquestre et donc une anticipation d'un risque ? Elle a pour nature d'être contraignante et indemnitaire. Pour cela, il est nécessaire que les manquements se soient réalisés. La programmation du *smart contract* peut aussi contredire l'esprit d'une clause.

- **Sur les effets réels (conditions suspensives – transfert des risques)**

La **fiducie** est l'opération par laquelle un constituant (personne physique ou morale) transfère la propriété de biens, de droits (ex. : une créance) ou de sûretés, qu'ils soient présents ou futurs, à un fiduciaire qui, les tenant séparés de son propre patrimoine, agit dans un but précis au profit d'un bénéficiaire. Ce bénéficiaire peut être un tiers, tel le créancier, dans le cadre d'une fiducie-sûreté, mais il peut aussi être le constituant. La technique de la fiducie suppose la rédaction (et la publication) d'un contrat de fiducie. On pourrait tout à fait envisager d'automatiser par un *smart contract* le transfert de l'actif.

Titre IV. Dispositions relatives à l'extinction des relations contractuelles. Concernant, les causes de l'extinction du contrat (causes – le terme extinctif – l'anéantissement rétroactif), **l'échéance d'un contrat** pourrait être inscrite dans un *smart contract* afin d'en déclencher automatiquement les effets. Par exemple, l'obligation de paiement d'un loyer prend fin automatiquement. La prorogation du contrat pourrait aussi se gérer dans le *smart contract*. Pour un contrat de prêt, en cas de durée déterminée du contrat, il est possible d'automatiser la survenance de la fin du contrat à une date précise. Ou en cas de remboursement du prêt par l'emprunteur, si l'emprunteur décide de rembourser le montant du prêt au prêteur, il est possible d'automatiser la fin du contrat. La clause serait ainsi écrite : « L'emprunteur se réserve le droit de rembourser en tout temps, avec 30 jours de préavis, tout ou partie du montant du prêt ainsi que les intérêts courus au jour du remboursement, et de mettre fin de cette manière au présent contrat ». Également, si le contrat prévoit que le prêteur peut demander le remboursement dans

un certain délai, la notification de la demande de remboursement peut être automatique avec pour conséquence la fin du contrat dès que le remboursement est exécuté.
Exemple de clause : « Au jour de l'exigibilité ou de la résiliation du contrat de prêt, selon la disposition contractuelle applicable, l'emprunteur remboursera au prêteur l'intégralité du montant du prêt restant dû ainsi que tous les intérêts courus à cette date. »
Sur les **effets de l'extinction du contrat** (effet rétroactif ou non – autres obligations : positives ou négatives), le *smart contract* pourrait être utilisé pour gérer la résiliation de deux contrats liés. Par exemple, la mise en œuvre d'une résiliation anticipée de l'un des deux contrats entraînera de plein droit et sans notification la résiliation de l'autre contrat.

Titre V Dispositions communes terminales. Les clauses de différend et notamment **la clause compromissoire** seraient intéressantes à traduite en *smart contract* si l'on admettait la résolution des litiges par la *blockchain*. La *blockchain* peut être un outil technique d'assistance, d'optimisation et d'amélioration de la justice, à condition d'avoir identifié préalablement les usages pertinents. D'un autre côté, la *blockchain* n'est plus un moyen, mais une fin en soi, y compris pour « dire le droit » et « rendre justice » dans des « juridictions distribuées ». Dans cette hypothèse, aux conceptions traditionnelles du droit et de la justice se substitue une base technologique distribuée proposant un nouveau système juridique. Certains auteurs le nomment *Lex cryptographia*[24]. Plusieurs plateformes mettent ainsi au service de la résolution des litiges une infrastructure reposant sur les caractéristiques de la *blockchain*, à savoir la décentralisation, l'immutabilité, la confidentialité, l'intégrité, l'authenticité et la non-répudiation et le caractère auditable. Ces plateformes se caractérisent par une indépendance plus ou moins forte et plus ou moins clairement affirmée au système juridique traditionnel, si bien qu'il est difficile de saisir dans quel courant philosophique elles se situent. Alors que se développent ces plateformes de résolution des litiges reposant sur une technologie *blockchain*, la question de l'encadrement par les dispositions du Code de procédure civile des services de justice en ligne et automatisée se pose. S'il est impossible d'assimiler de tels services à une procédure contentieuse, il est également difficile de les faire entrer dans les catégories d'arbitrage ou de modes amiables de règlement des différends. Nous ne ferons donc pas le choix de travailler sur les clauses compromissoires renvoyant à ces plateformes, considérant que, même si techniquement le *smart contract* est programmable, sa légitimité est discutable.

[24] P. DE FILIPPI, A. WRIGHT, *Blockchain and the Law: The Rule of Code*, Cambridge, Harvard University Press, 2018 ; R. KOULU,« Blockchains and Online Dispute Resolution: Smart Contracts as an Alternative to Enforcement », *SCRIPTed*, 2016, p. 40-69 ; K. WERBACH, N. CORNELL, « Contracts Ex Machina », *Duke Law Journal*, 2017, n° 67, p. 313-382.

Chapitre 2. Réalisation de la librairie de *smart contracts*

Le travail de sélection des clauses qui se prêtent à une traduction en *smart contract* nous permet dans un second temps de choisir les clauses sur lesquelles l'équipe doit concentrer son travail (section 1). Ces clauses font l'objet de test (section 2). L'étape suivante est la création de la librairie (section 3) et l'encadrement juridique de son utilisation (section 4).

Section 1 : Sélection des clauses

La sélection des clauses traduites en *smart contracts* est une étape délicate du projet. En effet, la clause sélectionnée doit correspondre tant aux exigences juridiques que techniques (chapitre 1). La programmation d'un *smart contract* est une étape longue et coûteuse. Deux listes ont alors été réalisées : les clauses prioritaires (§2) et les clauses secondaires (§1) susceptibles d'être traduites ultérieurement.

Paragraphe 1 : Clauses susceptibles d'être traduites dans une prochaine version de la librairie

Nous avons identifié sept clauses susceptibles d'être traduites en *smart contract* à l'occasion de projets ultérieurs : la clause d'indexation (A), la demande de renouvellement automatique de consentement en matière de cookies (B), la gestion automatique du droit de rétractation (C), la gestion automatique des révisions de prix (D), la suppression automatique des données personnelles dans les contrats de sous-traitance à travers deux hypothèses (E et F) et enfin, les clauses d'un contrat de cession de droits d'auteur (G).

A. Clause d'indexation

Définition et contexte. La clause d'indexation (également appelée, dans les textes, « clause d'échelle mobile ») permet de faire varier automatiquement le prix d'un contrat en fonction de l'application des modifications subies par une valeur suivant une valeur de référence et selon une périodicité prévue dans le contrat[25]. Son but est de permettre une adaptation du prix au contexte économique présent : elle préserve l'intérêt du contrat pour les parties en période d'inflation (ou déflation) ou encore dans des secteurs connaissant des variations de prix et de coûts importantes.

Contrats concernés. Elle concerne donc les contrats à exécution successive. Spécialement ceux de longue durée ou à durée indéterminée. Elle est particulièrement utilisée dans les contrats de bail, spécialement dans le contrat de bail commercial qui est appelé à se poursuivre sur une longue durée. On peut la rencontrer dans un contrat de travail (mais pas sur le SMIC), un prêt, une vente avec prix à payer en plusieurs fois. La clause d'indexation ne doit pas être confondue avec d'autres clauses, ou mécanismes légaux, permettant également une adaptation du contrat dans le temps : clause de révision, clause d'imprévision (ou de *hardship*), clause de renégociation (par ex. le mécanisme légal de révision triennale du loyer dans le bail commercial). Ces autres mécanismes peuvent coexister dans un contrat avec la clause d'indexation (par ex. le bail commercial va comporter une clause d'indexation, ce qui n'empêche pas les parties de recourir à la révision triennale ou à la révision judiciaire, autre mécanisme). La particularité de la clause d'indexation par rapport à ces autres mécanismes est son automaticité qui la rend éligible au *smart contract*. En effet, les autres mécanismes supposent, nous semble-t-il, une démarche volontaire d'une des parties au contrat ou encore le recours au juge.

Précisions sur les deux clauses (les textes applicables et le régime, les réserves, les variantes). On retrouve cette clause aux articles 1343 al. 2 du Code civil (admission de

[25] « La clause d'échelle mobile [...] est une stipulation par laquelle les parties s'accordent sur l'indexation du loyer, permettant une révision automatique en cours de bail selon une périodicité convenue. La clause s'applique de plein droit, sans formalité, à condition que soient prévus un indice et une périodicité et que les termes employés ne laissent aucun doute sur le caractère automatique de la révision... » CA Nancy, 6 mars 2000, n° 96/01217 : JurisData n° 2000-143521.

principe) et L. 112-1 à L. 112-4 du Code monétaire et financier (interdiction du recours à certains indices). Il existe des dispositions particulières, comme :

- la conversion d'usufruit de l'article 760 du Code civil,
- le bail commercial des articles L. 144-11 et L. 144-12 du Code de Commerce,
- la location-gérance de fonds de commerce à l'article L. 145-39 du Code de commerce,
- l'interdiction de l'indexation du salaire sur le SMIC dans les conventions ou accords collectifs à l'article L. 3231-3 du Code du travail,
- le prêt viager hypothécaire de l'article L. 315-9 du Code de la consommation,
- la location-vente assortie d'une promesse de vente de l'article L.313-55 du Code de la consommation.

Le domaine est assez peu régi par la législation, il existe donc une certaine liberté contractuelle.

Le choix de l'indice. Le choix de l'indice est important, car il s'agit de celui qui sera retenu dans le cadre d'un *smart contract***.** L'indice (ou la valeur) choisi doit être explicitement stipulé au contrat. Il faut clairement indiquer au contrat le recours au mécanisme d'indexation (en bail commercial, la simple mention d'un indice pourrait être comprise comme le rappel à titre indicatif des règles de modification judiciaire du loyer). Nous pouvons citer quelques exemples de valeurs : le cours d'une matière première (par ex. valeur or) ou d'une monnaie étrangère (clause de garantie de change), le prix d'une marchandise ou d'un service (clause d'échelle mobile), un indice (par ex. les indices publiés par l'INSEE ou un indice élaboré par les parties… indice simple ou composite…). L'article L. 112-1 al. 2 du Code monétaire et financier[26] interdit des indices basés sur : le SMIC, le niveau général des prix ou salaires, le niveau des prix ou salaires sans lien avec l'activité des parties ou l'objet du contrat (appréciation large par la jurisprudence). Il existe des règles spécifiques sur les marchés publics, baux ruraux, baux d'habitation… Il est possible de changer d'indice. Le contrat peut prévoir le cas où l'indice choisi cesserait d'exister. On peut indiquer que le contrat ne cessera pas et qu'un indice de remplacement sera automatiquement substitué. On peut en référer à l'indice de remplacement éventuellement publié par l'INSEE et/ou une clause selon laquelle les parties conviendront d'un nouvel indice ou à défaut s'en remettront au président du tribunal judiciaire.

[26] **Art. L. 112-1 al. 2 CMF :** « Est réputée non écrite toute clause d'un contrat à exécution successive, et notamment des baux et locations de toute nature, prévoyant la prise en compte d'une période de variation de l'indice supérieure à la durée s'écoulant entre chaque révision ». En principe on doit tenir compte de l'indice anniversaire publié lors de la modification.

Les modalités de l'indexation. Il existe une grande liberté contractuelle sur ce point. On peut indexer la totalité ou une partie du prix, l'autre étant fixe ou basée sur un autre calcul. La partie fixe du prix peut couvrir des frais fixes. Or cela suppose une structure de prix en différentes composantes. Selon l'arrêt de la Cour de cassation 3e civ., 14 janv. 2016, n° 14-24681**,** la clause ne prévoyant qu'une indexation à la hausse et non à la baisse est interdite. Cette décision porte sur le bail commercial, mais la solution pourrait être généralisée. Il faut donc qu'il y ait une réciprocité. Il y a un doute sur la validité de la clause permettant, certes, la hausse comme la baisse, mais fixant comme plancher le prix initial (le loyer initial du bail). Il serait alors possible de poser des seuils de déclenchement de l'indice, notamment l'absence de révision pour des variations de moins de 1 % du prix initial ou du prix dernièrement révisé, la révision ne pouvant dépasser *x* % du prix initial ou dernièrement révisé.

Les conditions de déclenchement de la clause. Deux modalités sont possibles. Ne permettre que le déclenchement à la hausse étant exclu par la jurisprudence récente (précitée), il reste tout de même des choix à faire. Soit le **déclenchement automatique est « de plein droit »,** c'est-à-dire que si le créancier ne demande pas effectivement le nouveau prix, il peut faire un rappel de loyers avec effet rétroactif, sous réserve de la prescription (5 ans en principe). Et inversement, le débiteur peut demander un remboursement du trop-perçu si c'est à la baisse. Les parties peuvent indiquer au contrat que le fait de ne pas demander la réévaluation ne vaut pas renonciation. L'intérêt du *smart contract* serait d'éviter les « oublis ». Le nouveau prix serait automatiquement calculé et figurerait dans la prochaine échéance. Soit le **déclenchement est à l'initiative de la partie la plus diligente** et dans ce cas on peut imaginer que les parties ne songent pas à réclamer l'application de l'indice.

Périodicité. Sur ce point, la liberté contractuelle l'emporte. En général, il s'agit d'un an pour le bail commercial, plus court (2-3 mois) pour d'autres contrats (par ex. contrat de fourniture).

Exemple de rédaction de la clause

Par dérogation au régime légal de révision des loyers prévu à l'article L. 145-38 du Code de commerce

ou

sans préjudice du régime légal de révision des loyers prévu à l'article L. 145-38 du Code de commerce

les parties conviennent d'indexer le prix (« loyer ») selon les variations de :

[citer l'indice : « L'indice des loyers commerciaux », et sa source : « publié par l'INSEE », en s'assurant de sa validité au regard de l'activité des parties ou de l'objet du contrat].

À cette fin, le prix (« loyer ») variera à la hausse comme à la baisse selon la périodicité suivante :

tous les
- ... ans / mois

à la date anniversaire de la prise d'effet du contrat
OU
à la date suivante : (JJ/MM).

La variation du prix sera calculée selon la formule suivante :

$$P1 = \frac{P0 \times Ic}{Ib}$$

Avec :
P1 : prix révisé
P0 : dernier prix en vigueur au jour de la révision
Ic : Indice de comparaison publié au jour de la révision
Ib : indice de base, en vigueur à l'entrée en vigueur OU la signature du présent contrat ; indiquer la référence / puis l'indice de comparaison de la dernière révision

OPTIONS :

- Les parties conviennent expressément que l'indexation convenue par les présentes s'appliquera de plein droit et sans formalité ni notification préalable. Elle résultera de l'indication du nouveau prix lors de la prochaine échéance.

- L'application de cette clause d'indexation se fera à l'initiative de la partie la plus diligente. La demande d'application de l'indexation doit être formée par lettre recommandée avec accusé de réception ou par telle opération informatique avec courriel de confirmation.

Dans le cas où l'une ou l'autre des parties ne se prévaudrait pas de la présente clause, le paiement, selon l'ancien taux, fait par le débiteur ou son encaissement par le créancier ne pourra en aucun cas être interprété comme la renonciation implicite à invoquer le jeu de l'indexation. Une éventuelle renonciation ne pourra résulter que d'un accord écrit entre les parties.
Disparition de l'indice. Au cas où, pour quelque raison que ce soit, l'indice convenu par les présentes pour la variation du prix (« loyer ») cesserait d'être publié, l'indexation sera faite en prenant pour base l'indice de remplacement.

B. Demande de renouvellement automatique de consentement en matière de cookies

Définition de la clause. Cette clause est la traduction de deux principes importants en matière de traitement de données personnelles : tout traitement de données doit avoir une durée limitée et certains traitements de données nécessitent le recueil du consentement exprès et préalable de la personne concernée. On pourrait utiliser un *smart contract* pour automatiser (i) la suppression automatique des données collectées (adresses IP) par les cookies et/ou (ii) la demande de renouvellement de consentement en cas d'utilisation de cookies.

Contexte d'utilisation de la clause. Les éditeurs de sites internet utilisent très souvent des cookies (des traceurs) qui collectent les adresses IP des internautes (données personnelles) à des fins diverses : pour faire fonctionner le site, mais aussi parfois à des fins de mesures d'audience, publicité ciblée, ou encore partage sur les réseaux sociaux, etc. Ces cookies donnent lieu à l'apposition d'un bandeau « Cookies » en bas des sites internet. Certains cookies nécessitent l'accord exprès et préalable de l'internaute. Lorsque cet accord est nécessaire, les données collectées ne peuvent être conservées au-delà d'une durée maximum de 13 mois, ce qui signifie que l'éditeur du site doit supprimer les adresses IP collectées et renouveler le recueil de consentement de l'internaute au bout des 13 mois pour appliquer de nouveaux traceurs. On pourrait utiliser un *smart contract* (ou 2) pour automatiser : la suppression automatique des adresses IP tracées à l'issue du délai de 13 mois et la demande de renouvellement du consentement par finalité de cookies à l'issue du délai de 13 mois.

Textes applicables et régime juridique. Sur le fondement de l'article 6 du règlement général sur la protection des données (RGPD), mais également au regard de la délibération de la Commission nationale de l'informatique et des libertés (CNIL) du 19 juillet 2019, les éditeurs de sites internet ont l'obligation de recueillir le consentement des utilisateurs préalablement au dépôt de certains cookies (de publicité ciblée, de réseaux sociaux ou de mesure d'audience sous conditions) et de leur fournir un moyen de s'y opposer. La durée de validité de ce consentement est de 13 mois maximum. Les traceurs nécessitant un recueil du consentement ne peuvent être utilisés en écriture ou en lecture tant que l'utilisateur n'a pas préalablement manifesté à cette fin sa volonté, de manière libre, spécifique, éclairée et univoque par une déclaration ou par un acte positif clair.

S'agissant du caractère libre du consentement, la CNIL considère que le consentement ne peut être valable que si la personne concernée est en mesure d'exercer valablement son choix et ne subit pas d'inconvénients majeurs en cas d'absence ou de retrait du consentement. Concrètement, l'éditeur doit informer l'internaute de chaque finalité des cookies et recueillir son consentement par finalité avec une case à cocher.

Il conviendra de vérifier si une suppression automatique des adresses IP et le renouvellement automatique du consentement sont possibles par un *smart contract* et si c'est techniquement pertinent.

Exemple de rédaction

Extrait d'un bandeau cookies :

« Nous utilisons des cookies à des fins de régie publicitaire

□ J'accepte □ Je refuse

Vos données sont supprimées automatiquement et votre consentement aux cookies est à nouveau recueilli au bout de 13 mois en application d'un *smart contract* ».

C. Gestion automatique du droit de rétractation

Définition de la clause. Un consommateur exerce son droit de rétractation et se fait automatiquement rembourser par le vendeur. Les textes applicables sont l'article L. 221-5 du Code de la consommation sur les contrats conclus à distance et hors établissement et les articles

L. 221-18 et suivants du Code de la consommation. Le consommateur qui conclut un contrat à distance ou un contrat hors établissement a un droit de rétractation de 14 jours. L'exercice du droit de rétractation pourrait se gérer dans un *smart contract* – envoi de la notification par le consommateur et remboursement automatique de sa commande à réception du produit par le vendeur.

Régime juridique. Le consommateur dispose d'un délai de 14 jours pour exercer son droit de rétractation d'un contrat conclu à distance, à la suite d'un démarchage téléphonique ou hors établissement, sans avoir à motiver sa décision ni à supporter d'autres coûts que ceux prévus aux articles L. 221-23 à L. 221-25.

Le délai mentionné au premier alinéa court à compter du jour :

1° de la conclusion du contrat, pour les contrats de prestation de services et ceux mentionnés à l'article L. 221-4 ;

2° de la réception du bien par le consommateur ou un tiers, autre que le transporteur, désigné par lui, pour les contrats de vente de biens. Pour les contrats conclus hors établissement, le consommateur peut exercer son droit de rétractation à compter de la conclusion du contrat.

Le consommateur exerce son droit de rétractation en informant le professionnel de sa décision de se rétracter par l'envoi, avant l'expiration du délai prévu à l'article L. 221-18, du formulaire de rétractation mentionné au 2° de l'article L. 221-5 ou de toute autre déclaration, dénuée d'ambiguïté, exprimant sa volonté de se rétracter.

Le consommateur renvoie ou restitue les biens sans retard excessif et, au plus tard, dans les quatorze jours suivant la communication de sa décision de se rétracter à moins que le professionnel ne propose de récupérer lui-même ces biens.
Le consommateur ne supporte que les coûts directs de renvoi des biens, sauf si le professionnel accepte de les prendre à sa charge ou s'il a omis d'informer le consommateur que ces coûts sont à sa charge.
Néanmoins, pour les contrats conclus hors établissement, lorsque les biens sont livrés au domicile du consommateur au moment de la conclusion du contrat, le professionnel récupère les biens à ses frais s'ils ne peuvent pas être renvoyés normalement par voie postale en raison de leur nature.

Lorsque le droit de rétractation est exercé, le professionnel rembourse le consommateur de la totalité des sommes versées, y compris les frais de livraison, sans retard injustifié et au plus tard dans les quatorze jours à compter de la date à laquelle il est informé de la décision du consommateur de se rétracter.
Pour les contrats de vente de biens, à moins qu'il ne propose de récupérer lui-même les biens, le professionnel peut différer le remboursement jusqu'à récupération des biens ou jusqu'à ce que le consommateur ait fourni une preuve de l'expédition de ces biens, la date retenue étant celle du premier de ces faits.

Exemple de rédaction

Le client dispose d'un délai de 14 jours pour exercer son droit de rétractation, sans avoir à motiver sa décision ni à supporter de frais. Le délai court à compter du jour de la conclusion du contrat.

Le jour où le contrat est conclu n'est pas compté dans le délai qui commence à courir au début de la première heure du premier jour et prend fin à l'expiration de la dernière heure du dernier jour du délai. Si ce délai expire un samedi, un dimanche ou un jour férié ou chômé, il est prorogé jusqu'au premier jour ouvrable suivant.

Le client exerce son droit de rétractation en informant le vendeur de sa décision de se rétracter par l'envoi, avant l'expiration du délai, du formulaire de rétractation dont un modèle figure dans le contrat, ou de toute autre déclaration, dénuée d'ambiguïté, exprimant sa volonté de se rétracter.

Lorsque le droit de rétractation est exercé, le vendeur rembourse le client de la totalité des sommes versées, 14 jours à compter de la date à laquelle il est informé de la décision du client de se rétracter, sous réserve d’avoir réceptionné dans ce délai le produit vendu.

Le vendeur effectue ce remboursement en utilisant le même moyen de paiement que celui utilisé par le client pour la transaction initiale.

D. Gestion automatique des révisions de prix

Définition de la clause. Il s'agit de la clause de révision des prix dans un contrat de longue durée (2 ans et plus) entre deux professionnels ou entre un professionnel et un consommateur[27]. Lorsqu'un contrat de longue durée est conclu « B to B » ou « B to C », une clause de révision des prix faisant référence à un indice, est souvent prévue. Il pourrait être intéressant d'automatiser dans un contrat « B to B » de préférence, via un *smart contract*, la révision annuelle des prix, en intégrant la formule de calcul prévue au contrat.

Régime juridique. Dans les contrats-cadres du Code civil, il peut être convenu que le prix sera fixé unilatéralement par l'une des parties, à charge pour elle d'en motiver le montant en cas de contestation. Dans les contrats de prestation de service, à défaut d'accord des parties avant leur exécution, le prix peut être fixé par le créancier, à charge pour lui d'en motiver le montant en cas de contestation. Lorsque le prix ou tout autre élément du contrat doit être déterminé par référence à un indice qui n'existe pas ou a cessé d'exister ou d'être accessible, celui-ci est remplacé par l'indice qui s'en rapproche le plus.

Dans le Code de commerce, sauf dans les cas où la loi en dispose autrement, les prix des biens, produits et services relevant antérieurement au 1er janvier 1987 de l'ordonnance n° 45-1483 du 30 juin 1945 sont librement déterminés par le jeu de la concurrence. Les conditions générales de vente comprennent notamment les conditions de règlement, ainsi que les éléments de détermination du prix tels que le barème des prix unitaires et les éventuelles réductions de prix. Une convention écrite conclue entre le fournisseur, à l'exception des fournisseurs de produits mentionnés à l'article L. 443-2, et le distributeur ou le prestataire de services mentionne les obligations réciproques auxquelles se sont engagées les parties à l'issue de la négociation commerciale, dans le respect des articles L. 442-1 à L. 442-3. Cette convention est établie soit dans un document unique, soit dans un ensemble formé par un contrat-cadre et des contrats d'application. Sans préjudice des articles L. 442-1 à L. 442-3, tout avenant à la convention mentionnée au I fait l'objet d'un écrit qui mentionne l'élément nouveau le justifiant. La convention [sus]mentionnée fixe, aux fins de concourir à la détermination du prix convenu, les obligations suivantes : les conditions de l'opération de vente des produits ou des prestations de services, y compris les réductions de prix, et le cas échéant les types de situations dans

[27] Textes applicables : articles L. 410-2 et L. 441-1 du Code de commerce, article L. 441-3 du Code de commerce, articles 1164, 1165 et 1167 du Code civil.

lesquelles et les modalités selon lesquelles des conditions dérogatoires de l'opération de vente sont susceptibles d'être appliquées.

Pour la traduction en *smart contract*, il conviendra de faire attention au déséquilibre significatif dans la fixation du prix et dans les conditions de révision du prix aux pratiques anticoncurrentielles et aux règles sur les prix dans certains secteurs ou zones.

Exemple de rédaction

Les prix seront révisés automatiquement à la date anniversaire du contrat. La révision des prix se fera selon la formule suivante :
Pn = Po x (In/Io), dans laquelle :
Pn = représente les prix recalculés et applicables pour l'année
Po = représente les prix initiaux
In = représente le dernier indice ____ connu au jour de la révision des prix
Io = dernier indice connu au jour de l'entrée en vigueur du contrat
En cas de disparition de l'indice, les parties détermineront d'un commun accord la nouvelle référence. Ce nouvel indice devra être choisi de telle sorte qu'il soit le plus proche possible de l'indice disparu et qu'il respecte l'esprit des parties.

Le nouveau prix sera automatiquement appliqué à toutes les prestations effectuées après la date de la révision tarifaire.

E. Suppression automatique des données personnelles (hypothèse 1)

Définition de la clause. Cet engagement est la traduction d'un principe important en matière de traitement de données personnelles : tout traitement de données doit avoir une durée limitée. Il s'agit de l'art. 4 de la Loi informatique et libertés (LIL) et de l'article 5 du RGPD. On pourrait utiliser un *smart contract* pour automatiser l'engagement du responsable de traitement de suppression automatique des données personnelles d'une personne à l'issue de la durée de traitement qui aura été prédéfinie. Cet engagement peut s'appliquer lorsqu'un responsable de traitement traite les données personnelles d'une personne physique.

Régime juridique. Selon l'article 4 de la LIL / article 5 du RGPD : *les données 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. Le responsable de traitement doit s'engager à conserver les données pour une durée limitée. Cette durée est variable. Elle est fonction de la finalité (du but poursuivi) du traitement et ne devra pas être excessive (par exemple, le temps de la relation contractuelle pour les informations figurant dans un fichier client). La CNIL ou la loi fixe des durées maximales pour certains traitements. En dehors de ces durées imposées, il appartient au responsable de traitement de fixer lui-même la durée en fonction de l'utilité des données au regard du but poursuivi. Au-delà de cette durée, le responsable de traitement doit effacer ou anonymiser les données. On ne retrouve pas forcément cet engagement dans un contrat passé entre le responsable de traitement et la personne concernée par le traitement. En effet, des traitements de données peuvent être réalisés en dehors de l'exécution d'un contrat (article 5 de la LIL) ou du recueil de consentement de la personne (ex : le respect d'une obligation légale). Réserve technique : nous ignorons si une suppression automatique de données est possible par un *smart contract* et si elle est techniquement pertinente.

Exemple de rédaction

Extrait d'une clause relative à la protection des données personnelles d'un contrat ou encore d'une politique d'utilisation des données (ou charte de confidentialité).

« Pour la finalité [À COMPLÉTER]

Le responsable de traitement s'engage à conserver les données personnelles pour une durée de [À COMPLÉTER] à compter de [PRÉCISER]. À l'issue de cette durée, les données personnelles sont automatiquement et définitivement supprimées en application d'un *smart contract* ».

F. Suppression automatique des données personnelles dans un contrat de sous-traitance (hypothèse 2)

Définition de la clause. Cette clause presque identique à la précédente repose sur le même principe important en matière de traitement de données personnelles : tout traitement de données doit avoir une durée limitée. On pourrait utiliser un *smart contract* pour automatiser l'engagement du sous-traitant de suppression automatique des données personnelles au terme du contrat de sous-traitance. Cet engagement peut s'appliquer lorsqu'un responsable de

traitement a recours à un sous-traitant pour traiter des données personnelles pour son compte. Dans ce cas, le responsable de traitement donne ses instructions au sous-traitant, notamment sur la durée des traitements. Le sous-traitant doit s'engager à détruire les données qu'il détient à l'expiration de cette durée, sauf instruction contraire et expresse du responsable de traitement visant à obtenir une restitution de ces données. Selon l'article 28 du RGPD, lorsqu'un traitement doit être effectué par un sous-traitant pour le compte d'un responsable du traitement, le traitement est régi par un contrat ou un autre acte juridique, qui lie le sous-traitant à l'égard du responsable du traitement, définit l'objet et la durée du traitement, sa nature et sa finalité, etc.

Le contrat entre le responsable de traitement et le sous-traitant doit prévoir qu'au terme du contrat, le sous-traitant s'engage, au choix des parties :
- à détruire toutes les données personnelles ou
- à renvoyer toutes les données au responsable de traitement et à détruire les copies existantes.

Exemple de rédaction

Extrait d'une clause relative à la protection des données personnelles d'un contrat de sous-traitance de données personnelles :

« Au terme du contrat, soit le [À COMPLÉTER] le sous-traitant s'engage à détruire toutes les données personnelles traitées par l'application d'un *smart contract* ».

G. Clauses d'un contrat de cession de droits d'auteur

Illustrations de clauses en droits d'auteur. Les contrats d'exploitation des droits de la propriété intellectuelle (PI) offrent de nombreux intérêts pour leur automatisation par les *smart contracts*. Le sujet du contrat est un bien complètement dématérialisé, ce qui facilite la mise en place d'une relation contractuelle complétement dématérialisée elle aussi. Le droit de PI est souvent enregistré dans un registre qu'il soit national (INPI, UK IPO…), régional, notamment européen (EUIPO, EPO), voire international (OMPI), et ces registres pourraient servir de tiers de confiance / oracle dans le cadre d'une opération sur une *blockchain* par exemple. Les opérations sur les titres de PI sont soumises à des formalités importantes qui peuvent être (et sont déjà largement) automatisées. Il est possible de choisir le niveau de difficulté en fonction du type de droit (brevet, droits d'auteur, droit des marques) et du type de contrat (licence ou cession) ou du nombre d'inventeur(s) / auteur(s) ou de leur statut salarial. Ainsi, la cession

d'une marque contre une somme forfaitaire sera une opération relativement circonscrite par rapport à un contrat de licence de droits d'auteur avec multiples auteurs et le versement de royalties qui pourrait courir pendant des décennies. On pourrait retenir trois types de contrats en fonction de leur fréquence d'utilisation, de leur importance économique, et de leur niveau de complexité :

- le contrat d'édition, contrat classique entre l'auteur d'un manuscrit et son éditeur ;
- le contrat de licence (non exclusive) approuvé par n'importe quel utilisateur de plateforme de partages de contenus type YouTube, Instagram ou Facebook lorsqu'il met à disposition ses œuvres originales sur ces plateformes. On peut aussi penser aux licences Creative Commons ;
- le contrat de cession d'un brevet.

À ce stade, nous nous focaliserons sur le contrat d'édition. Les clauses susceptibles d'être traduites sont les suivantes.

Clause de définition du cadre règlementaire spécifique. Clause par laquelle le contrat d'édition rappelle qu'il est soumis aux règles de l'article L. 132-17-8 du Code de la propriété intellectuelle

Clause de définition de l'œuvre sujette du contrat ainsi que de son (ou de ses) auteur(s).

Cession de droits d'auteur. Clause par laquelle le propriétaire de droits d'auteur transfère l'intégralité de ses droits (à l'exception du droit moral) sur une œuvre à un éditeur dans le cadre, par exemple, d'un contrat d'édition

Durée de la cession / durée du droit : la durée de la cession est liée à la durée du droit de PI. En fonction du droit de PI, le calcul et la durée du droit peuvent varier, ce qui présente des défis intéressants pour l'automatisation de cette clause. Pour les droits d'auteur, les dates de la finalisation de l'œuvre, de sa publication, mais surtout du décès de son auteur jouent un rôle sur la durée des droits, les droits d'auteur protégeant l'œuvre pendant la vie de son auteur et 70 ans après sa mort. Dans le cadre d'une exécution automatique du contrat, il faudra donc un mécanisme pour obtenir, par l'intermédiaire d'un tiers de confiance (comme peut l'être l'état civil), une information fiable sur le décès de l'auteur.

Remise du manuscrit : l'auteur a l'obligation de remettre à l'éditeur le manuscrit à une date donnée. Cela requiert de faire la preuve de l'existence et du transfert, dans les temps, de l'œuvre. On pourrait imaginer lier le *smart contract* à un service de *time-stamping*[28] via une *blockchain* pour assurer la mise en œuvre de cette étape.

Gestion collective. Clause qui définit les conditions dans lesquelles l'auteur confie à l'éditeur le soin de percevoir pour son compte et de lui reverser les rémunérations des droits en provenance d'organismes de gestion collective

Reddition des comptes et règlement des droits. Clauses définissant les obligations de communication de l'éditeur à l'auteur des chiffres de ventes et des recettes liées à l'exploitation de l'œuvre

Avances sur droits. Versement par l'éditeur d'une somme à l'auteur avant la finalisation de l'œuvre

Publication et exploitation permanente. L'éditeur s'engage à publier et à exploiter l'œuvre.

Rémunération. Elle définit le % du prix de vente revenant à l'auteur en fonction des types d'édition, du volume de vente, du marché géographique (en France / hors de France), de l'exploitation directe ou par un tiers des droits cédés.

Les textes applicables en France sont :

- obligations contractuelles communes – Code de la propriété intellectuelle : articles L. 131-1 à L. 131-9
- conditions spécifiques au contrat d'édition – Code de la propriété intellectuelle : articles L. 132-1 à L. 132-17.

Texte applicable au Royaume-Uni : Copyright, Designs and Patents Act 1988, sections 90 -95, cf. par exemple s. 90[29].

[28] Ce mécanisme, aussi appelé horodatage, associe une date et une heure à une donnée informatique ou un événement. Il précise donc l'instant où une opération a été effectuée.

[29] https://www.legislation.gov.uk/ukpga/1988/48/section/90 et https://www.legislation.gov.uk/ukpga/1988/48/section/94.

Exemple de rédaction

- En France :

Modèle type de contrat d'édition du SNE (Syndicat national de l'édition) de 2014, https://www.sne.fr/app/uploads/2017/10/2-Contrat-edition-LG_SNE_version-25-nov-2014-1.pdf

Par exemple :

« 1.1 - L'auteur cède à titre exclusif à l'éditeur, qui accepte pour lui-même et ses ayants droit, les droits de reproduction et de représentation afférents à l'œuvre de sa composition qui a pour titre provisoire ou définitif [ ], à l'exception toutefois des droits d'adaptation audiovisuelle qui font l'objet d'un contrat écrit sur un document distinct conformément à l'article L.131-3, alinéa 3 du Code de la propriété intellectuelle. »

- In English, UK law applicable:

From Thomson Practical Law - Assignment of copyright (pro-assignor):

[Pursuant to and for the consideration set out in the Main Agreement OR In consideration of the sum of £ [AMOUNT] (receipt of which the Assignor expressly acknowledges)], the Assignor hereby assigns to the Assignee absolutely the following rights [in the United Kingdom OR throughout the world]:

(a) the entire copyright and all other rights in the nature of copyright subsisting in the Works [and the Future Works];

(b) any database right subsisting in the Works [and the Future Works] [and in all preliminary drafts or earlier versions of the Works [and the Future Works]]; and

(c) all other rights in the Works [and the Future Works] of whatever nature, whether now known or created in the future, to which the Assignor is now, or at any time after the date of this agreement may be, entitled by virtue of the laws in force in the United Kingdom [and in any other part of the world],

in each case [for the whole term including any renewals, reversions, revivals and extensions] and together with all related rights and powers arising or accrued, including the right to bring, make, oppose, defend, appeal proceedings, claims or actions and obtain relief (and to retain any damages recovered) in respect of any infringement, or any other cause of action arising from

ownership, of any of these assigned rights, whether occurring before, on or after the date of this agreement.

Paragraphe 2 : Sélection définitive de cinq clauses pour la librairie

Les cinq clauses qui ont été retenues par les membres du projet de recherche sont : la clause d'exclusion (A), la clause d'un contrat d'option (B), la clause de contrat à terme (C), la clause de « *buy or sell* » (D) et enfin la clause de préemption (E). Pour chaque clause, une présentation permet d'en connaître le cadre juridique et de proposer un exemple de rédaction. Cette proposition de rédaction, traduite en anglais pour rendre cette librairie accessible à nos partenaires européens, est une étape indispensable pour l'écriture en pseudo-code et enfin la traduction en langage informatique. Ces étapes ont été respectées et sont décrites pour chacune des clauses traduites dans la librairie.

A. Clause d'exclusion

1. Présentation juridique

Définition de la clause. La clause d'exclusion est un droit accordé aux associés d'une société permettant d'exclure un des leurs à titre de sanction. Cette clause doit être mentionnée dans les statuts ou adoptée par décision soit du dirigeant (le terme président fait référence à la société par actions simplifiée, le terme de dirigeant est générique), soit de l'organe collégial compétent, soit des associés lors d'une assemblée générale. Une telle clause intervient dans un contexte de dissolution, de redressement ou de liquidation judiciaire d'un associé. Les juges se sont déclarés incompétents pour exclure un associé à défaut de disposition légale leur conférant expressément ce pouvoir[30].

Régime juridique. La validité des clauses statutaires d'exclusion est expressément reconnue par la loi dans les sociétés par actions simplifiée (SAS), dans les sociétés européennes (SE),

[30] Droit commun - Code civil Article 545, article 1193 ; Société par actions simplifiée - Code de commerce Article L. 227-16, article L. 227-17, article L. 227-18 ; Société européenne non cotée - Code de commerce Article L. 229-12, article L. 229-14 ; Société d'exercice libéral - Loi n° 90-1258 du 31 décembre 1990 Article 21 alinéa 2 ; Société à capital variable - Code de commerce Article L. 231-6 alinéa 2 et 3 ; Actions à dividende prioritaire sans droit de vote - Code de commerce Article L. 228-35-10 ; Groupement agricole d'exploitation en commun - Code rural et de la pêche maritime Article R. 323-38.

dans les sociétés d'exercice libéral (si prévu par le décret réglementant la profession), dans les sociétés à capital variable, dans les groupements agricoles d'exploitation en commun ou encore pour les actions à dividende prioritaire sans droit de vote. Dans les SAS ou les SE, la volonté d'ajouter ou de modifier une clause d'exclusion dans les statuts doit être unanime. Dans d'autres sociétés, une majorité « simple » sera nécessaire.

Afin qu'une clause d'exclusion soit valide, elle doit remplir différentes conditions :

1° Certaines de ces conditions ont été dégagées par un arrêt de la Cour de cassation du 8 mars 2005[31].

2° Les motifs pouvant entraîner l'exclusion doivent être formellement inscrits dans la clause. L'exclusion d'un associé pour un motif non prévu dans la clause ou imprécis ne sera pas valable.

3° La clause doit avoir préalablement déterminé l'organe compétent pour décider de l'exclusion ainsi que de la procédure à suivre. Si la compétence n'est pas déterminée, ce rôle pourra être confié à l'assemblée des associés (dans ce cas l'associé concerné par la démarche d'exclusion ne pourra pas être privé de son droit de vote) ou un autre organe social. La procédure doit stipuler que l'associé concerné doit être informé des motifs qui lui sont reprochés ainsi que de la possibilité qu'il a de s'exprimer et de se défendre concernant les faits qui lui sont reprochés.

4° La clause d'exclusion doit être conforme à l'intérêt social et à l'ordre public.

5° L'associé exclu doit être indemnisé. Cette indemnisation doit être prévue et déterminée dans la clause d'exclusion. Le montant de cette indemnisation peut être librement fixé, à condition qu'il ne soit pas dérisoire. L'auteur de l'indemnisation devra également être déterminé dans la clause.

Exemple de rédaction

Une clause d'exclusion permet d'exclure un associé de la société lorsqu'il a réalisé un acte déterminé. La clause est appliquée à partir du moment où les titres que l'associé exclu détenait sont rachetés.

Toute procédure d'exclusion d'un associé devra être inscrite de manière détaillée dans la clause et se déroulera de la manière suivante.

- L'associé dont l'exclusion est envisagée devra, dans un premier temps, en être informé.

[31] Cass. Com., 8 mars 2005, n° 02-17692.

- À cette occasion, l'associé devra être informé du motif pour lequel son exclusion est envisagée ainsi que le droit qu'il a de s'exprimer et de se défendre. Les motifs retenus contre lui devront obligatoirement être clairement précisés dans la clause d'exclusion.
- L'organe compétent devra se prononcer sur l'exclusion de l'associé. La décision de l'exclusion devra être prise en application des modalités de décision déterminées dans les statuts.
- Suite à la décision prise par l'organe décisionnaire, s'il est décidé que l'associé est exclu, il sera procédé à un calcul du prix de cession des titres de l'associé. Ce calcul devra se faire en application des modalités prévues par la clause d'exclusion. La clause devra préciser également la date à laquelle la valorisation des titres sera effectuée ainsi que la personne qui en sera chargée.
- Pour conclure l'exclusion, l'associé exclu devra céder ses titres. Ils pourront être rachetés soit par les autres associés, soit par un tiers désigné à cet effet par la société. La clause d'exclusion peut prévoir de retirer ces droits non pécuniaires à l'associé en cours d'exclusion. Cela concerne le droit à l'information sur la société, le droit de participer aux assemblés et le droit de vote.

Suite à la cession de ses titres, l'associé est exclu de la société.

2. Traduction en anglais de la clause

Definition of the clause. The exclusion clause is a right granted to the partners of a company allowing them to exclude one of their members as a sanction.
This clause must be mentioned in the articles of association or taken by a decision of either the manager (the president refers to the simplified joint stock company, the term manager is generic) of the company, or of the competent collegial body of the company, or of the associates during a general assembly. Such a clause comes into play in a context of dissolution, reorganization or judicial liquidation of a partner. The judges have declared themselves incompetent to exclude a partner in the absence of a legal provision expressly conferring the power to do so.

Legal framework. The validity of statutory exclusion clauses is expressly recognized by the law in simplified joint stock companies (SJSC), in European companies (EC), in liberal practice companies (if provided for by the decree regulating the profession), in companies with variable

capital, in agricultural joint ventures or even for non-voting preferred shares. In SJSC or SE companies, the will to add or modify an exclusion clause in the articles of association must be unanimous. In other companies, a "simple" majority will be required.

In order for an exclusion clause to be valid, it must meet several conditions:

1° Some of these conditions have been identified by a decision of the Court of Cassation of March 8, 2005.

2° The grounds for exclusion must be formally stated in the clause. The exclusion of a partner for a reason not provided for in the clause or for an imprecise reason will not be valid.

3° The clause must have previously determined the body competent to decide on the exclusion as well as the procedure to follow. If the competence is not determined, this role can be entrusted to the assembly of associates (in this case the associate concerned by the exclusion process cannot be deprived of his right to vote) or to another corporate body. The procedure must stipulate that the partner concerned must be informed of the reasons for which he is accused and of the possibility of expressing his views and defending himself in relation to the facts of which he is accused.

4° The exclusion clause must be consistent with the company's interest and with public policy.

5° The excluded partner must be compensated. This compensation must be provided for and determined in the exclusion clause. The amount of this compensation may be freely determined, provided that it is not derisory. The author of the compensation must also be determined in the clause.

Example of drafting

An exclusion clause makes it possible to exclude a partner from the partnership when he has performed a certain act. The clause is applied from the moment when the securities held by the excluded partner are bought back.

Any procedure of exclusion of a partner must be detailed in the clause and will proceed as follows:

- The partner whose exclusion is contemplated must first be informed.
- On this occasion, the associate must be informed of the reason for which his exclusion is being considered, as well as the right he has to express himself and to defend himself. The grounds for exclusion must be clearly specified in the exclusion clause.
- The competent body must decide on the exclusion of the associate. The decision to exclude the associate must be taken in accordance with the decision-making procedures set out in the articles of association.

- Following the decision taken by the decision-making body, if it is decided that the associate is excluded, a calculation of the transfer price of the securities of the associate will be made. This calculation will have to be made in application of the terms and conditions provided for by the exclusion clause. The clause must also specify the date on which the valuation of the securities will be carried out as well as the person in charge of it.
- To conclude the exclusion, the excluded partner will have to sell his securities. They can be repurchased either by the other associates, or by a third party designated for this purpose by the company. The exclusion clause can provide for the withdrawal of these non-pecuniary rights from the partner during the exclusion. This concerns the right to information about the company, the right to participate in the meetings and the right to vote.

Following the transfer of his shares, the partner is excluded from the company.

3. Traduction en langage informatique

a) Template de la clause d'exclusion

Exclusion de plein droit d'un associé

L'exclusion de plein droit d'un associé intervient en cas de dissolution, de redressement ou de liquidation judiciaire de celui-ci.

b) Pseudo-code

```
Importer ERC1644 ou npt quel standard permettant de forcer un transfert //
Importer libraire oracle
// l'utilisation de l'oracle est asynchrone, donc la récupération de l'information se fait en deux fonctions
Définir variables:

Adresse de l'associé,
Totale de ses actions,
informationsExclusion
informationsSociete
```

Fonction requêteOracle () {

Appel à la fonction de la librairie de l'oracle qui va chercher l'information sur l'API de infogreffe de l'existence d'une procédure.

}

Fonction callbackOracle () {

Si l'information a été récupérée et si existence procédure collective == "OUI", alors

→ utilise fonction du token représentant l'action pour forcer le transfert de toute la balance de l'associé vers l'adresse de la société
}

c) Code en langage solidity

```
event statutRequete(string description);
address adresseAssocie;

uint256 totalActionsAssocie;
bytes informationsExclusion;
bytes informationsSociete;

IERC1644 public tokenAction; // ERC20 being used as the underlying asset
function requeteProcedureColl() payable {

if (provable_getPrice(&quot;URL&quot;) &gt; this.balance) {
emit statutRequete(&quot;La requete ne peut pas etre envoye, car pas assez de ether&quot;);

} else {
```

```
emit statutRequete(&amp;quot;La requete a pu etre envoye, en attente d une reponse...&amp;quot;);

provable_query(&amp;quot;URL&amp;quot;,
&amp;quot;json(https://api.datainfogreffe.fr/api/v1/Entreprise/ProceduresCollectives/{numeroSiren}?{t
okenId}).ExistenceProcedure&amp;quot;);

}
}

function __callback(
bytes32 _myid,
string result
)

public
{
require(msg.sender == provable_cbAddress());
if (keccak256 (bytes(result)) == keccak256 (&amp;quot;OUI&amp;quot;) )
{
tokenAction.controllerTransfer(adresseAssocie, this, totalActionsAssocie,

informationsExclusion, informationsSociete);

emit ControllerTransfer(this,
adresseAssocie,this,totalActionsAssocie,informationsExclusion, informationsSociete);

}
}
```

B. Clause d'un contrat d'option

1. Présentation juridique

Définition de la clause. Un contrat d'option confère à son détenteur un droit d'acheter (option d'achat ou « *call* ») ou de vendre (option de vente ou « *put* ») un instrument financier dit sous-jacent (devises, actions, valeurs mobilières, etc.) à un prix, appelé prix d'exercice (ou « *strike* ») et à une date déterminée à l'avance[32]. Pour les options dites « américaines », ce droit peut être exercé à tout moment durant la durée de vie de l'option et jusqu'à la date d'échéance préalablement fixée (date d'exercice). Un contrat d'option permet à la personne détentrice de celle-ci de se réserver le choix durant la période déterminée d'acheter/de vendre ou non l'actif sous-jacent pour lequel il a acheté une option. Par exemple, un acquéreur souhaite acheter une maison et trouve une demeure qui semble l'intéresser. Il apprécierait, néanmoins, de prendre le temps de réfléchir à cet achat tout en s'assurant de pouvoir acquérir cette maison si, finalement, il la choisit. L'acquéreur va alors acheter une option auprès du vendeur, lui conférant un droit d'acquérir la maison au prix d'exercice et durant une période déterminée. L'obtention de ce droit se fait en l'échange du versement d'une prime (c'est le prix de l'option) qui restera définitivement acquise par le vendeur.

À l'issue de la période couverte par l'option, soit l'acquéreur a décidé d'acheter la maison et cet achat se fera au prix d'exercice (hors le prix de l'option). Soit l'acquéreur n'a pas exercé son option et ne dispose plus de la possibilité d'acquérir la maison au prix d'exercice. Il s'expose alors au risque que le prix ait augmenté.

Le contrat d'option constitue donc une certaine forme de couverture pour l'acquéreur ou le vendeur contre une potentielle hausse ou baisse du prix de l'actif sous-jacent. Il s'agit d'un droit que l'on retrouve dans les promesses unilatérales de contrat de vente ou d'achat.

Régime juridique. Le contrat d'option crée des obligations pour chacune des parties au contrat. Concernant l'auteur du contrat d'option, il se verra obligé de conclure le contrat si le bénéficiaire de l'option décide de lever l'option. En effet lorsqu'il propose le contrat d'option, l'auteur agit comme s'il avait déjà donné son consentement définitif quant à la réalisation de la vente ou de l'achat. Il s'engage à conclure le contrat si un bénéficiaire lève l'option. L'auteur

[32] Option de souscription ou d'achat d'actions - Code de commerce : Article L. 225-177 et s. ; Relatifs à la comptabilité - Code de commerce : Article L. 123-25 et s., Article L. 232-5 ; Autres articles - Code de commerce : Article L. 622-7, article L. 626-18, article 1124 du Code civil.

ne peut donc ni se rétracter ni négocier tant que le bénéficiaire utilise son droit d'option sur ce contrat, ou qu'il décide de lever cette option. Une exception à ce principe existe lorsque la possibilité, pour l'auteur, de se rétracter est expressément prévue dans le contrat. En revanche, tant qu'aucun bénéficiaire potentiel n'exerce son droit d'option ou ne lève l'option dont il dispose pour conclure le contrat, l'auteur est libre et n'est engagé auprès de personne. L'auteur s'engage également à ne pas conclure de contrat avec des tiers jusqu'à l'expiration du délai d'option.

Concernant le bénéficiaire, il peut être amené à devoir verser une prime à l'auteur du contrat afin de pouvoir bénéficier d'un droit d'option. Cette prime agit comme une indemnité puisqu'elle est la contrepartie de l'immobilisation du contrat pendant la durée d'exercice du droit d'option de la part du bénéficiaire. Cette somme est due de manière définitive à l'auteur.

Le contrat d'option doit remplir des conditions spécifiques afin d'être valide. Les éléments essentiels du contrat doivent être déterminés étant donné que son auteur a donné son consentement définitif de contracter. Dans cette logique, l'auteur du contrat doit être capable de mesurer la portée de son engagement.

Le droit d'option constitue également une condition de validité du contrat, l'auteur doit consentir ce droit au bénéficiaire et il doit également veiller à ne pas restreindre son exercice.

L'option dont dispose le bénéficiaire a une durée de validité fixée par les parties. À l'issue de ce délai, le bénéficiaire ne dispose plus de son droit d'option et l'auteur du contrat n'est plus engagé vis-à-vis du bénéficiaire.

Exemple de rédaction

La personne qui propose (appelée « promettant ») le contrat d'option d'achat ou le contrat d'option de vente s'engage à conclure le contrat envisagé selon les dispositions qu'il a préalablement fixées.

La personne intéressée (appelée « bénéficiaire ») par le contrat du promettant va pouvoir disposer d'un droit d'acheter (option d'achat, appelée « *call* ») ou d'un droit de vendre (option de vente, appelée « *put* ») une quantité d'actif sous-jacent à un prix fixé par le promettant (appelé « prix d'exercice »).

Dans les deux cas, si le bénéficiaire souhaite obtenir une option sur l'achat ou la vente d'actif sous-jacent, il devra verser une prime (appelée « prix de l'option ») au promettant qui l'acquerra définitivement.

Cette option confère au bénéficiaire le droit de choisir, durant une période déterminée, s'il souhaite ou non lever l'option et conclure le contrat. Durant cette période le promettant ne pourra ni se rétracter, ni négocier les dispositions du contrat, ni s'engager auprès d'un tiers pour le même contrat.

À l'issue de cette période, deux situations sont envisageables :

- soit le bénéficiaire décide de lever l'option, auquel cas le contrat va être conclu et le bénéficiaire versera le prix d'exercice au promettant.

- soit le bénéficiaire décide de ne pas lever l'option, auquel cas le contrat ne sera pas conclu et le promettant sera libéré de tout engagement vis-à-vis du bénéficiaire. Le promettant ne percevra pas le prix d'exercice, mais il gardera le prix de l'option.

2. Traduction en anglais

Definition of the clause. An option contract gives the holder the right, but not the obligation, to buy (a 'call option' or 'call') or sell (a 'put option' or 'put') an underlying asset (currencies, securities, real estate….) at a later date at an agreed upon price (known as the 'strike price'). Options known as "American Options" allow the holder to exercise the option at any time during the duration of the option, that is to say, at any time before and on the date of expiration. An option allows the holder to choose, during the specified period, to buy or sell the underlying asset for which he bought the option. Example: A potential buyer is interested in buying a house but wants more time to decide whether to buy the house or not. He wants to secure the right to buy the house if and when he ultimately choses to do so. He will buy an option from the seller of the house that will give him a right to buy the house at a specific price during a specified period of time. In order to have this right, he pays a premium (the price of the option) to the seller. The seller will keep this premium even if there is ultimately no sale.
At the end of the option period, either the buyer has decided to buy the house and this sale will be done at the strike price or the buyer has not exercised his option and therefore cannot buy

the house at the strike price. If the buyer does not exercise his option, the house is back on the market and its value may rise above or fall below the strike price agreed in the option.
The option contract is therefore a kind of insurance policy for the buyer or the seller against a potential increase or decrease in the price of the underlying asset. This type of right can be found in unilateral promises to sell or buy.

Legal framework. The option contract creates obligations for each of the parties to the contract. The grantor of the options contract (also known as the optionor) will have to conclude the contract if the beneficiary of the option choses to exercise the option. When the grantor offers the option contract, he has already agreed to sell or buy the asset. He commits to entering the contract if the beneficiary exercises the option. Thus, as a rule, the grantor cannot withdraw or renegotiate the offer as long as the holder benefits from the option or if he decides to exercise the option. However, an exception to this rule can be agreed in the contract, enabling the grantor to withdraw before the holder exercises the option. The grantor also undertakes not to enter into a contract with third parties until the option expires. As for the holder, he may have to pay a fee to the grantor to benefit from the option. This fee compensates the grantor for offering to be bound to the holder. This fee is irrevocably due to the grantor. The option contract must meet specific conditions to be valid. The key elements of a contract must be determined since that the grantor has already given his final consent to the contract. Accordingly, the grantor must be able to ascertain the scope of his commitment. The option right is also a condition of the validity of the contract. The grantor must grant this right to the holder and also must not restrict its exercise. The parties agree the duration of the option. After the expiration date, the holder no longer has his option right and the grantor is no longer bound to the holder. Thus, the holder is the only one who can conclude an option contract.

Examples of drafting

The person who offers an option contract to buy or sell (and who is known as the grantor or optionor) agrees to enter into the proposed contract under the conditions he has set.

The person interested by the contract of the grantor (known as the ‘holder’ or the ‘beneficiary’) has a right to buy (a ‘call’ option) or a right to sell (a ‘put’ option) an agreed quantity of the underlying asset at a price set by the grantor (the ‘strike price’).

In both cases, if the beneficiary wants an option to buy or sell an underlying asset, he will need to pay a premium (the 'option premium') to the grantor who will own that premium permanently.

This option grants the beneficiary the right to choose, during a set time period, whether to exercise the option and enter into the contract. During that time period the grantor can neither withdraw from nor renegotiate the contract, nor can he commit towards a third party regarding the same contract.

At the end of the period, there are two possible situations:

- the beneficiary chooses to exercise the option in which case the contract will be concluded, and the beneficiary will pay the strike price to the grantor.

- the beneficiary chooses not to exercise the option in which case the contract will not be concluded, and the grantor will be freed of any commitment towards the beneficiary. The grantor will not receive the strike price but will keep the option premium.

3. Traduction en langage informatique

a) Template du contrat d'option

Entre

[.....], d'une part (ci-après le Concédant) et,

[....], d'autre part (ci-après le Bénéficiaire/ titulaire de l'option),

ci ensemble les parties,

En contrepartie de la somme de [prix de l'option] payée par le Bénéficiaire au Concédant, ce dernier concède irrévocablement au Bénéficiaire pendant [durée d'exercice], une option d'achat permettant d'imposer au Concédant de vendre [les Actifs sous-jacents] au Bénéficiaire à [Prix d'exercice].

Les Parties ont décidé de soumettre l'exécution de cet accord à un *Smart contract* déployé sur le dispositif d'enregistrement électronique partagé Ethereum (ci-après la "*Blockchain*"), à l'adresse suivante :

La souscription, l'exercice et la fermeture de l'option se feront par l'intermédiaire du *Smart contract*.

[L'Actif] est représenté par un jeton numérique, au sens de l'article …. disponible dans la *Blockchain* à l'adresse suivante : …..

Le paiement de l'option d'achat et de [l'Actif] se feront dans la *Blockchain* avec le jeton numérique suivant : [...] disponible à l'adresse suivante : …….

En annexe du présent contrat figure le code source du *Smart contract*.

b) Pseudo-code

Contrat {
//Importer les tokens nécessaires Importer Token pour représenter l'Actif Importer Token pour représenter la monnaie de paiement
//Définir les états du contrat d'option Etats du contrat : mort, initialisé, ouvert, exercé, fermé

//Déclarer les variables du contrat d'option quantitéActif prixActif prixOption dateExpiration
//Définir les adresses du Concédant et du Bénéficiaire adresseConcédant adresseBénéficIaire

fonction Constructeur (tokenActif, tokenPaiement, quantitéActif, prixActif, prixOption, dateExpiration) {
 	
Vérifie que les adresses des tokens sont existantes, que la quantité et le prix de l'actif, ainsi que la date d'expiration sont supérieures à 0.
 	
Etat du contrat sur None.
 	
end
}

fonction initialisation () { Après avoir vérifié que contrat sur None et que le concédant est bien la personne interagissant avec la fonction, transfère l'Actif au *Smart contract* qui le tient en séquestre. État du contrat sur initialisé

end }

fonction acheterOption () {
 	
Après avoir vérifié que l'état est sur initialisé, que la date d'expiration ne soit pas dépassée, et que le Bénéficiaire est bien la personne interagissant avec la fonction → le bénéficiaire paye le prix de l'option
 	
État du contrat sur "acheté"
 	
end
}

fonction fermerOption () {
 	
Après avoir vérifier que l'état est sur initialisé ou que l'état est sur ouvert et que la date d'expiration est inférieure au moment où est déclenchée cette fonction.
 	
Vérifie que l'adresse est celle du concédant.
 	
Renvoie les Actifs au Concédant.
 	
État : fermé.
 	
end
}

Fonction exercerOption () {

Vérifie que état est ouvert, date d'expiration pas encore passée, que personne interagissant avec la fonction est le Bénéficiaire → paye le prix * quantité, transfère l'actif représenté au bénéficiaire,

Récupère les actifs

État : exercé

end
}

c) Code en langage solidity

```
pragma solidity ^0.6.0;

import "interfaces/my_IERC20.sol";
import "interfaces/my_IERC1400.sol";

contract clauseOption {

// déclaration des différents états d'une option
enum optionStates {
STATUS_UNUSED, // par défaut : 0
STATUS_PENDING,
STATUS_CLOSED
}

// Represents an option parameters
struct OptionRight {
address promisor;
```

```
address recipient;
uint256 priceOption;
uint256 priceExercise;
uint256 expirationDate;
optionStates status;
}

// Mapping from partition UID to Option
mapping ( uint256 => OptionRight) public options;

IERC1400 public tokenStock; // token qui représente la partition (ERC1410)
IERC20 public tokenPayment; // token qui représente le coin (ERC20)

//address public recipient; // le bénéficiaire (buyer)
//address public promisor; // le promettant (seller)
//uint256 public priceExercise; // prix de l'exercice (au moins la valeur de la partition)
//uint256 public priceOption; // prix à payer par le bénéficiaire pour le droit d'option
//uint256 public expirationDate; // date après laquelle le bénéficiaire ne peut plus exercer son droit d'option

constructor(address _tokenStockAddress, address _tokenPaymentAddress) public {
require(_tokenStockAddress != address(0));
require(_tokenPaymentAddress != address(0));

tokenStock = IERC1400(_tokenStockAddress);
tokenPayment = IERC20(_tokenPaymentAddress);
}

//------------------------------------------------------------
```

```
// Option right events

//-------------------------------------------------------

event StartOption(address indexed _promisor, address indexed _recipient, address escrow,
uint256 _partitionUid, uint256 _priceExercise, uint256 _priceOption, uint256
_expirationDate);

//event Opened(address indexed _recipient, uint256 _priceOption);

//event Closed(address indexed _promisor, uint256 _amountPayment);


//-------------------------------------------------------

// Option right functions

//-------------------------------------------------------

// The promisor initiate the Option Exercise on a partition he owns

function startOption(address recipient, uint256 partitionUid, uint256 priceExercise, uint256
priceOption, uint256 duration) public returns (bool){

require(options[partitionUid].status != optionStates.STATUS_PENDING, "Une partition peut
être l'objet de plusieurs options, mais pas simultanément");

require(tokenStock.getPartitionOwner(partitionUid) == msg.sender, "Seul le promettant peut
initier une option");

require(tokenStock.getPartitionStatus(partitionUid) == 0, "la partition cible de l'option doit être
active - STATUS_ACTIVE is 0");

require(priceExercise > 0);

require(priceOption > 0);

require(duration > 0, "la durée d'une option est exprimée en jours");

require(tokenPayment.balanceOf(recipient) >= priceOption + priceExercise);

require(tokenPayment.allowance(recipient, address(this)) >= priceOption);
```

```
options[partitionUid].promisor = msg.sender;

options[partitionUid].recipient = recipient;

options[partitionUid].priceOption = priceOption;

options[partitionUid].priceExercise = priceExercise;

options[partitionUid].expirationDate = now + duration * 1 days;

options[partitionUid].status = optionStates.STATUS_PENDING;

// promisor send previously a transaction to ERC1400 to approve EscrowAccount on partitionUid for priceExercice

require(tokenStock.allowanceEscrow(msg.sender, address(this), partitionUid) == true, "le contract de séquestre doit être autorisé à modifier le status de la partition");

// Transfer partitions[partitionUid].owner vers sequestre (address(this)) sur contract ERC1400

// changement status partition en CONFINED (pas de chgt de owner)

// mise à jour de la variable mapping escrows (fait sur le contract ERC1400)

uint256 endDate = now + duration * 1 days;

//tokenStock.confinePartition(msg.sender, recipient, partitionUid, endDate, priceExercise);

// the promisor is the origin (msg.sender) of the transactions call

tokenStock.confinePartition(recipient, partitionUid, endDate, priceExercise);

// Transfer du priceOption sur le contract ERC20

tokenPayment.transferFrom(recipient, msg.sender, priceOption);

// recipient doit avoir mis une allowance auparavant et avoir une balanceOf > priceOption

emit StartOption(msg.sender, recipient, address(this), partitionUid, priceExercise, priceOption, endDate);

return true;

}
```

```
// The recipient accepts during the option exercice duration

function recipientAccept(uint256 partitionUid) public returns (bool){

require(msg.sender == options[partitionUid].recipient);

require(tokenPayment.balanceOf(msg.sender) >= options[partitionUid].priceExercise);

require(now <= options[partitionUid].expirationDate);

require(options[partitionUid].status == optionStates.STATUS_PENDING);

require(tokenStock.getPartitionStatus(partitionUid) == 1, "la partition cible de l'option doit être en séquestre - STATUS_CONFINED is 1");

// Transfer de la partition au coût de priceExercise

require(tokenStock.allowanceEscrow(options[partitionUid].promisor, address(this), partitionUid) == true);

// the recipient is the origin (msg.sender) of the transactions call

tokenStock.escrowTransfer(options[partitionUid].promisor, options[partitionUid].priceExercise, partitionUid);

require(tokenStock.stopOptionByRecipient(partitionUid));

options[partitionUid].status = optionStates.STATUS_CLOSED;

return true;

}

// The recipient denies during the option exercice duration

function recipientDeny(uint256 partitionUid) public returns (bool){

require(msg.sender == options[partitionUid].recipient);

require(now <= options[partitionUid].expirationDate);

require(options[partitionUid].status == optionStates.STATUS_PENDING);
```

```
require(tokenStock.getPartitionStatus(partitionUid) == 1, "la partition cible de l'option doit être en séquestre - STATUS_CONFINED is 1");

// levée du séquestre de la partition

require(tokenStock.stopOptionByRecipient(partitionUid));

options[partitionUid].status = optionStates.STATUS_CLOSED;

return true;

}

// The promisor stop the option exercice after the expiration date

function stopOption(uint256 partitionUid) public returns (bool){

require(msg.sender == options[partitionUid].promisor);

require(now > options[partitionUid].expirationDate);

require(tokenStock.stopOptionByPromisor(partitionUid));

options[partitionUid].status = optionStates.STATUS_CLOSED;

return true;

}

// Help to debug

function isOrigin() public returns (address){

return tokenStock.whoIsOrigin();

}

    RRRRRR.          }
```

C. Clause d'un contrat à terme

1. Présentation juridique

Définition de la clause. Un contrat à terme désigne un engagement entre deux contractants leur permettant de vendre ou acheter de l'actif sous-jacent à une date future prédéterminée (le terme) et à un prix déterminé. Contrairement aux contrats d'option, ces contrats sont dit « fermes », c'est-à-dire qu'il existe un engagement mutuel entre le vendeur et l'acquéreur (contrat synallagmatique qui engage les deux parties). Ainsi, à l'arrivée du terme, l'acheteur a l'obligation de verser la somme convenue pour acquérir l'actif que le vendeur lui livrera en contrepartie. Un contrat à terme s'apparente finalement à un simple contrat dont on ne va que retarder l'échéance. Le contrat à terme constitue à l'origine une certaine forme de couverture contre une éventuelle hausse ou baisse d'un produit/actif sous-jacent. Ce contrat pourra, par exemple, protéger les entreprises commerciales contre les risques de changes ou des variations désavantageuses de taux d'intérêt. Par exemple, un actionnaire détient des actions à un certain prix actuel. Il craint une baisse de la valeur de ses actions dans les jours qui viennent et souhaite les vendre avec une certaine sécurité. Il va alors conclure un contrat à terme avec un acheteur qui aspire, quant à lui, à une hausse de la valeur de ces actions. À l'arrivée du terme, bien que la valeur des actions ait chuté (ou même augmenté), l'actionnaire vendra ses actions au prix qu'elles avaient à la formation du contrat. Le contrat à terme s'apparente donc à une garantie du prix de vente et/ou d'achat. Ce type de contrat est fortement utilisé en matière de spéculation. En effet, il est possible de conclure des contrats de vente ou d'achat à terme en pariant sur l'évolution future du cours des actions. Chacune des parties fait un pari, pensant être gagnante.

Régime juridique. Le terme est une forme particulière de l'obligation. Il s'agit d'un événement futur et certain dont dépend l'exécution du contrat et plus précisément l'exigibilité ou l'extinction de l'obligation. Le terme est à distinguer de la condition qui constitue un évènement futur et incertain.

Le terme n'a donc aucun effet concernant l'existence de l'obligation, mais seulement concernant son exigibilité. L'obligation existe alors dès la formation du contrat, mais son exécution ne peut être exigée qu'à la survenance du terme. Si l'événement dont dépend l'exigibilité est futur et certain, le contrat est à terme, mais s'il est futur et incertain, le contrat est à condition. Le terme peut être certain, lorsque l'on connaît la date à laquelle l'événement va se produire (par exemple lors de la vente d'une maison à une date précise), et incertain,

lorsque l'on ignore cette date (par exemple dans le cadre d'un contrat d'assurance vie, on ne sait pas quand la personne va décéder, mais malheureusement cela arrivera bien un jour). Attention, dans le contrat à terme incertain, c'est seulement la date qui est incertaine et non pas l'événement, sinon le contrat est à condition. Lorsque des parties à un contrat ont décidé d'affecter un terme à un contrat, suspendant l'exigibilité de l'obligation sans que le délai du terme ne soit prononcé, le juge peut fixer ce délai. Lors d'un contrat à terme, tout ce qui est exigé à terme ne peut être réclamé avant l'échéance. Cependant, il est possible que certains éléments du contrat soient payés/réalisés en avance. Dans ce cas, ils ne pourront être répétés à l'arrivée du terme.

Exemple de rédaction

Le terme est une modalité d'un contrat de vente ou d'achat. Le contrat entre deux contractants est à terme lorsque son exécution dépend d'un événement futur et certain. La date de formation du contrat diffère donc de la date d'exécution du contrat.
Dès la formation du contrat, l'acheteur et le vendeur s'engagent mutuellement. Il n'est possible pour aucune des parties de se rétracter ou de négocier les dispositions du contrat.
Le vendeur fixe un prix correspondant à la valeur de la part d'actif sous-jacent qu'il souhaite vendre. Le prix ainsi que la quantité d'actif sous-jacent sont fixés à la formation du contrat et ne pourront être différents à l'arrivée du terme lorsque le contrat sera exécuté. L'acheteur et le vendeur vont déterminer une date à laquelle le contrat devra être exécuté. Cette date constitue le terme du contrat.
Tant que l'échéance n'est pas atteinte, aucune des parties au contrat ne pourra réclamer son dû. À l'arrivée du terme, l'acheteur devra verser la somme convenue lors de la formation du contrat en l'échange de quoi le vendeur lui cédera la part d'actif sous-jacent convenue.

2. Traduction en anglais

Definition. A future contract (or simply 'futures') has been defined in EU regulations as "a contract to buy or sell a commodity or financial instrument [at] a designated future date at a price agreed upon at the initiation of the contract by the buyer and seller. Every futures contract has standard terms that dictate the minimum quantity and quality that can be bought or sold, the smallest amount by which the price may change, delivery procedures, maturity date and other characteristics related to the contract." The standardisation of futures contracts is the key difference with forward contracts which operate in a similar way but are private and

customisable agreements traded over-the-counter. Futures (and also forward) contracts bind both parties to the transfer of the asset. At the expiration date, the buyer has an obligation to pay the agreed amount to acquire the asset that the seller will deliver in return.

A futures contract is, in essence, a normal contract whose performance is postponed. Futures contracts are designed to offer coverage against the price fluctuations of an asset. They can, for instance, protect a company against variations in foreign exchange rates or interest rates.

For example: A shareholder fearing that the price of his share is about to decrease will want to sell them now at a set price. He thus enters a futures contract with a buyer (who, conversely, bets that the share price will rise). At the designated date, the price of the shares will be that decided at the initiation of the contract rather than the current price, irrespective of whether this spot price is higher or lower than the agreed price.

A futures contract is therefore akin to a guarantee on the sale and/or purchase price.

This type of contract is very commonly used for speculation and hedging. Indeed, it is possible to enter into futures sales or purchase contracts to bet on future fluctuations in share prices, the seller and buyer making opposite bets on future price changes.

Legal framework. In French law, futures contracts come under the category of obligations with a suspensive term ("obligations à terme"). A suspensive term is a specific characteristic of the obligation. A suspensive term consists in a future and certain event which postpones the performance of the obligation. An obligation with a suspensive term differs from a conditional obligation whose performance depends upon a future and uncertain event.
A suspensive term has no impact on the existence of the obligation but impacts its performance. The obligation exists as soon as the contract has been formed but its performance can only be requested upon the occurrence of the term.

Note that the date of the term can itself be either certain or uncertain. The date of the term can be certain when we know for sure when the future event is going to happen, for instance when the date of the sale of a house has been specified. However, the date of the term can also be uncertain, when we do not know when the future event is going to happen, for instance, for a life insurance contract when, although we know that the person is going to die, we cannot be

sure when. Here, it is just the date of the event that is unknown not the fact that it is going to happen, otherwise it would be a conditional obligation.

When the parties to a contract have decided to set a term but the duration of the term is not ascertainable, the court can fix the duration of the term.

What is due with a term cannot be claimed before the expiration of the term. However, it may be that some parts of the contract are performed in advance. In that case, what has been performed in advance may not be recovered.

Examples of drafting

A suspensive term adds a distinctive feature to a sale or purchase contract. With a suspensive term, the performance is due upon the occurrence of a future and certain event. Thus, the date of the performance of the contract differs from that of its formation.
The seller and the buyer are bound at the date of formation of the contract. Neither party can withdraw from the contract or renegotiate its content.

The seller sets a price for the quantity of underlying asset he wants to sell. The price and quantity of the underlying asset are set at the time of the formation of the contract and cannot be different when the contract is performed at the advent of the term.

The buyer and the seller agree the date of performance of the contract. This date is the term of the contract.

Until the arrival of the term, none of the parties to the contract can claim what they are due.

Upon the arrival of the term, the buyer will pay the price agreed at the initiation of the contract and the seller will transfer the ownership of the asset.

3. Traduction en langage informatique

a) Template du contrat à terme

Entre

[.....], d'une part (ci-après le Vendeur) et,

[....], d'autre part (ci-après l'Acheteur),

Ci-ensemble les parties,

Le Vendeur s'engage envers l'Acheteur à lui céder [l'Actif] au prix de [....] à la date de [...]. L'Acheteur s'engage envers le Vendeur à prendre livraison de [l'Actif] au prix de [...] à la date de [...].

Les Parties ont décidé de soumettre l'exécution de cet accord à un *Smart contract* déployé sur le dispositif d'enregistrement électronique partagé Ethereum (ci-après la "Blockchain"), à l'adresse suivante :

[L'Actif] est représenté par un jeton numérique, au sens de l'article …. disponible dans la Blockchain à l'adresse suivante : …..

Le paiement de [l'Actif] se fera dans la Blockchain avec le jeton numérique suivant : [...] disponible à l'adresse suivante : …..

En annexe du présent contrat figure le code source du *Smart contract*.

b) Pseudo-code

Contrat {
//Importer les tokens nécessaires Importer Token pour représenter l'Actif Importer Token pour représenter la monnaie de paiement
//Définir les états du contrat d'option États du contrat : mort, initialisé, ouvert, exercé, fermé
//Déclarer les variables du contrat d'option quantitéActif prixExercice prixOption dateExpiration
//Définir les adresses du Concédant et du Bénéficiaire adresseConcédant adresseBénéficIaire
fonction Constructeur (tokenActif, tokenPaiement, quantitéActif, prixExercice, prixOption, dateExpiration) {

Vérifie que les adresses des tokens sont existantes, que la quantité et le prix d'exerice, ainsi que la date d'expiration sont supérieures à 0. État du contrat sur None. end }

fonction initialisation () {
 	
Après avoir vérifié que le contrat est sur None et que le concédant est bien la personne interagissant avec la fonction, transfère l'Actif au *Smart contract* qui le tient en séquestre.
 	
État du contrat sur initialisé
 	
end
}

fonction acheterFuture() {
 	
Après avoir vérifié que l'état est sur initialisé, et que le Bénéficiaire est bien la personne interagissant avec la fonction → le Bénéficiaire transfère le prix du Future et du bien qu'il s'engage à acheter qui est séquestré.
 	
État du contrat sur "acheté"
 	
end
}

Fonction recupererActifs() {

Vérifie que état est ouvert, que personne interagissant avec la fonction est le Bénéficiaire, que date de “réalisation du Future est bonne” → transfère les actifs au Bénéficiaire

État : exercé

end
}

c) Code en langage solidity

```
pragma solidity ^0.6.0;

import "interfaces/my_IERC20.sol";

import "interfaces/my_IERC1400.sol";

contract clauseForward {

// déclaration des différents états d'une vente à terme

enum forwardStates {

STATUS_UNUSED, // par défaut : 0

STATUS_PENDING,

STATUS_CLOSED

}

// Represents a forward sale parameters

struct ForwardSale {

address seller;

address recipient;

uint256 priceForward;

uint256 expirationDate;
```

```
forwardStates status;

}

// Mapping from partition UID to the forward sale

mapping ( uint256 => ForwardSale) public forwards;

IERC1400 public tokenStock; // token qui représente la partition (ERC1410)

IERC20 public tokenPayment; // token qui représente le coin (ERC20)

address tokenStockAddress;

constructor(address _tokenStockAddress, address _tokenPaymentAddress) public {

require(_tokenStockAddress != address(0));

require(_tokenPaymentAddress != address(0));

tokenStock = IERC1400(_tokenStockAddress);

tokenPayment = IERC20(_tokenPaymentAddress);

tokenStockAddress = _tokenStockAddress;

}

//------------------------------------------------------------

// Forward sale events

//------------------------------------------------------------

event StartForwardSale(address indexed _seller, address indexed _recipient, address
escrow, uint256 _partitionUid, uint256 _priceForward, uint256 _expirationDate);

//------------------------------------------------------------

// Option right functions

//------------------------------------------------------------
```

```
// The seller initiates the forward sale on a partition he owns

function startForwardSale(address recipient, uint256 partitionUid, uint256 priceForward,
uint256 duration) public returns (bool){

require(forwards[partitionUid].status != forwardStates.STATUS_PENDING, "Une
partition peut être engagée dans une autre vente à terme");

require(tokenStock.getPartitionOwner(partitionUid) == msg.sender, "Seul le possesseur de
la partition peut initier une vente à terme");

require(tokenStock.getPartitionStatus(partitionUid) == 0, "la partition cible de la vente
doit être active - STATUS_ACTIVE is 0");

require(priceForward >= 0);

require(duration > 0, "la durée de la vente à terme est exprimée en minutes");

require(tokenPayment.balanceOf(recipient) >= priceForward +
tokenStock.getPartitionAmount(partitionUid));

require(tokenPayment.allowance(recipient, address(this)) >= priceForward);

require(tokenPayment.allowance(recipient, tokenStockAddress) >=
tokenStock.getPartitionAmount(partitionUid));

forwards[partitionUid].seller = msg.sender;

forwards[partitionUid].recipient = recipient;

forwards[partitionUid].priceForward = priceForward;

forwards[partitionUid].expirationDate = now + duration * 1 minutes;

forwards[partitionUid].status = forwardStates.STATUS_PENDING;

// the seller sends previously a transaction to ERC1400 to approve EscrowAccount on
partitionUid for priceExercice

require(tokenStock.allowanceEscrow(msg.sender, address(this), partitionUid) == true, "le
contract de séquestre doit être autorisé à modifier le status de la partition");
```

```
// Transfer partitions[partitionUid].owner vers sequestre (address(this)) sur contract ERC1400

// changement status partition en CONFINED (pas de chgt de owner)

// mise à jour de la variable mapping escrows (fait sur le contract ERC1400)

uint256 endDate = now + duration * 1 minutes;

tokenStock.confinePartition(recipient, partitionUid, endDate, tokenStock.getPartitionAmount(partitionUid));

emit StartForwardSale(msg.sender, recipient, address(this), partitionUid, priceForward, endDate);

return true;

}


// The sale is launched

function launchSale(uint256 partitionUid) public returns (bool){

require(msg.sender == forwards[partitionUid].recipient || msg.sender == forwards[partitionUid].seller, "la vente peut être déclenchée soit par le vendeur, soit par acheteur");

require(tokenPayment.balanceOf(forwards[partitionUid].recipient) >= forwards[partitionUid].priceForward + tokenStock.getPartitionAmount(partitionUid));

require(now >= forwards[partitionUid].expirationDate);

require(forwards[partitionUid].status == forwardStates.STATUS_PENDING);

require(tokenStock.getPartitionStatus(partitionUid) == 1, "la partition cible doit être en séquestre - STATUS_CONFINED is 1");

// la partition est déconfinée

tokenStock.deconfinePartition(forwards[partitionUid].seller, 1234);

// Transfer du priceForward sur le contract ERC20
```

```
tokenPayment.transferFrom(forwards[partitionUid].recipient,
forwards[partitionUid].seller, forwards[partitionUid].priceForward);

// Transfer de la partition

require(tokenStock.allowanceEscrow(forwards[partitionUid].seller, address(this),
partitionUid) == true);

tokenStock.escrowExplicitTransfer(forwards[partitionUid].seller,
forwards[partitionUid].recipient, tokenStock.getPartitionAmount(partitionUid),
partitionUid);

forwards[partitionUid].status = forwardStates.STATUS_CLOSED;

return true;

}

}
```

D. Clause de « *buy or sell* »

1. Présentation juridique

Définition de la clause. La clause « *buy or sell* » va permettre à un actionnaire de proposer la vente de ses titres à un prix déterminé à un autre actionnaire ou à défaut d'acquérir les siens au prix auquel il était prêt à les lui céder. Cette clause est notamment valable dans les sociétés avec deux associés égalitaires (détenant chacun 50% du capital social). Un associé (A) va alors pouvoir faire une offre de vente de ses parts à un autre associé (B). Si l'associé (B) refuse l'achat des parts de (A) au prix annoncé, l'associé (A) se devra d'acheter les parts de (B) au prix qu'il a fixé lors de sa proposition de vente. La clause « *buy or sell* » est inscrite dans un contrat d'actionnaire ou dans un contrat de société et permet d'éviter les situations de blocage en cas de mésentente entre les associés, qui pourraient porter préjudice la société. En effet au sein d'une société avec deux associés égalitaires, il est possible que l'un d'eux (ou les deux) souhaite mettre fin à son association avec la société. La clause « *buy or sell* » permet alors de résoudre

ce conflit et chaque partie a la possibilité de déclencher le jeu de la clause, ce qui aboutira nécessairement à la sortie de la société de l'un des associés.
Voici son fonctionnement en différentes étapes.

• L'avis de vente

L'actionnaire souhaitant mettre fin à son association avec la société (« actionnaire offrant ») doit en informer l'autre actionnaire (« actionnaire restant ») par le biais d'un avis écrit appelé « avis de vente » dans lequel il propose de racheter au prix d'achat et selon les modalités de l'avis, la **totalité** des actions détenues par l'autre actionnaire.

• La période d'option (ou période d'élection)

Suite à la réception de l'avis de vente, l'actionnaire restant dispose d'une période de X jours (délai à déterminer – liberté contractuelle) appelée « période d'option » pour informer l'actionnaire offrant de son choix :

- soit de racheter la **totalité** des actions détenues par l'actionnaire offrant au prix d'achat et selon les modalités de l'avis de vente ;
- soit de vendre la **totalité** des actions qu'il détient à l'actionnaire offrant au prix d'achat et selon les modalités de l'avis de vente.

Dans le cas où l'actionnaire restant n'a pas informé l'actionnaire offrant de son choix dans le délai imparti, il est réputé avoir accepté l'option selon laquelle il cède à l'actionnaire offrant la **totalité** des actions qu'il détient au prix d'achat et selon les modalités de l'avis de vente.

• Achat et vente d'actions

Si, donc, l'actionnaire restant choisit de ne pas racheter les actions détenues par l'actionnaire offrant, celui-ci doit racheter la **totalité** des actions détenues par l'actionnaire restant.

• Date de clôture

Tout achat ou vente d'actions prennent fin à la « date de clôture » soit le 60e jour civil à compter du dernier jour de la période d'option.

Régime juridique. La clause « *buy or sell* » n'est gouvernée par aucun texte en particulier. Cependant, il faudra veiller à respecter un certain nombre de dispositions qui pourraient être

applicable à cette clause bien qu'elles ne soient pas propres à celle-ci. Nous retrouvons principalement les règles relatives à l'offre de contracter et à l'acceptation (C.Civ. article 1113 et suivants), et celles relatives aux promesses (C.Civ. article 1124). S'agissant de la détermination du prix en matière de vente l'article 1591 du Code civil pourra être applicable, ainsi que les articles 1592 et 1843-4 du Code civil lorsqu'il est fait appel à un tiers pour déterminer le prix. Les clauses « *buy or sell* » ont pu susciter quelques interrogations, mais leur validité intrinsèque n'est guère discutée actuellement, même s'il convient de prendre certaines précautions.

La finalité de cette clause sera toujours la sortie de la société d'un des associés. Mais il est possible que cette sortie se fasse contre sa volonté même s'il a initialement consenti à l'insertion de cette clause au sein du pacte d'associés. Cependant il faut veiller à ne pas confondre clause « *buy or sell* » et clause d'exclusion. La seconde intervient dans un but disciplinaire ce qui n'est pas le cas de la première, qui intervient dans un but de protection de la longévité de la société.

À la suite de cette clause, l'un des associés va inévitablement devoir céder ses titres, par conséquent, les textes relatifs à la détermination du prix de cession sont applicables (C.Civ. article 1124 et article 1591) ainsi que ceux relatifs à l'intervention d'un tiers pour déterminer ce prix (qu'il soit un simple « tiers » C.Civ article 1592 ou un « expert » C.Civ article 1843-4).

Étant donné que la clause « *buy or sell* » fonctionne sur le principe des promesses unilatérales de ventes, il existe un risque de rétractation du promettant. Cependant, l'article 1124 du Code civil apporte certaines dispositions protégeant le bénéficiaire de ce risque. Cette clause comporte un second risque, celui que, du fait que les associés s'engagent respectivement à céder leurs titres et racheter ceux de l'autre, l'opération soit requalifiée en promesse synallagmatique.

Exemple de rédaction

Cette clause de sortie alternative s'applique en cas de différend entre associés suffisamment grave et susceptible de bloquer le fonctionnement de la société, voire mettre en péril la poursuite de l'activité de cette dernière (ci-après un « **Désaccord grave et persistant** »).

Ce différend devra avoir fait l'objet d'une **notification** par l'associé concerné à son coassocié sans qu'aucune solution n'ait été trouvée dans les 15 jours suivant ladite notification.

À défaut de règlement amiable à l'expiration d'un délai de 15 jours à compter de la demande écrite de la partie la plus diligente, chacun aura la possibilité de notifier à l'autre associé, par lettre recommandée avec accusé de réception sa volonté de céder la totalité de sa participation dans la société.

L’auteur de la notification indiquera le prix et les modalités de cession de sa participation.
Le coassocié **bénéficiaire** de cette offre sera tenu de choisir, dans le délai de 30 jours calendaires de la première présentation de la notification, entre les deux options suivantes :

- soit, accepter d’acquérir la totalité des titres de la société dont son coassocié est titulaire aux prix et aux conditions de l’offre ;

- soit, refuser d’acquérir les titres de son coassocié et lui céder la totalité des titres de la société dont il est titulaire aux prix et conditions mentionnés dans l’offre initiale.

Dans les deux cas, la réponse devra être effectuée par lettre recommandée avec accusé de réception, ou tout autre procédé équivalent.

2. Traduction en anglais

Definition of the clause. The buy-sell clause (also known as buyout or shotgun clause) allows a shareholder to make an offer to sell his shares at a fixed price to another shareholder but will require him, if his offer is declined, to acquire the shares of the other shareholder at the price he offered to sell his shares for.
This clause is notably useful in entities with two equal partners (each holding 50% of the company). Shareholder A can make an offer to sell his shares to Shareholder B. If B declines to purchase the shares of A at the set price, A will have to buy the shares of B at the set price.

Use of the clause A buy-sell clause is a provision included in a shareholders’ agreement or in a partnership agreement to avoid deadlock in the event of a disagreement between partners, which could be detrimental to the company. In a partnership with two equal shareholders, one (or both) of them may wish to end their partnership in the company. The buy-sell clause enables to solve this conflict as each party can trigger the mechanism which will result in the exit from the company of one of the partners.

Here are the different stages of the process:

- The notice of sale

The shareholder wishing to terminate his association with the company (the "offering shareholder") must inform the other shareholder (the "remaining shareholder") through a written notice called "the notice of sale" in which he offers to sell his shares to the other shareholder or to purchase, at the price and according to the terms set in the notice, all the shares held by the other shareholder.

- The option period

Upon receipt of the notice of sale, the remaining shareholder must, within a defined period set in the shareholders' agreement and known as the "option period", notify the offering shareholder of his choice:

- Either to purchase all the shares held by the offering shareholder at the price and under the terms set in the notice;
- Or to sell all his shares to the offering shareholder at the price and under the terms set in the notice.

In the event that the remaining shareholder has not informed the offering shareholder of his choice within the given timeframe, he is deemed to have agreed to sell all his shares to the offering shareholder at the price and under the terms set in the sale notice.

- Buying and selling shares

Thus, if the remaining shareholder chooses not to purchase the shares held by the offering shareholder, the offering shareholder must purchase all the shares of the remaining shareholder.

- Closing date

Any share purchase or sale ends at "closing date", ie 60 days after the last day of the option period.

Legal framework. The existence of buy-sell clauses may have raised some questions in France, but their inherent validity is hardly ever discussed any more, even though they need to be handled with caution.
The purpose of this clause will always be the exit of the company of one of the shareholders. This exit may be forced even though the shareholder will have agreed to the inclusion of such a clause in the shareholders' agreement. It is therefore important to distinguish the buy-sell clause from the clause of forced removal of a shareholder. The latter is actioned through a disciplinary action against a shareholder which is not the case of the former, which is aimed at ensuring the permanence of the company.

If the buy-sell clause is triggered, one of the shareholders will have to sell his shares. Therefore, the statutory provisions relevant to the determination of the sale price (Civil Code, article 1124 and article 1591) are applicable and so are the provisions relevant to the intervention of a third party in the determination of the price, be it any 'third party' (Civil Code, article 1592) or an 'expert' (Civil Code, article 1843-4).

Since the buy-sell clause operates following the principle of a unilateral promise of sale, there is a risk of revocation from the promisor. However, article 1124 of the Civil Code provides some protections for the promisee.

The buy-sell clause contains another risk: given that the partners in the company commit to sell or buy each other's shares, such mechanism risks being reclassified as a reciprocal promise.

Examples of drafting

This buy-sell clause applies in the event of a dispute between shareholders which is significant enough to block the operation of the company or even jeopardise the continuation of the company's activity (hereafter a "Serious and persistent disagreement").

This disagreement must have been the subject of a notification by one shareholder to the other and not have been resolved within 15 days following said notice.

If an amicable settlement is not reached within 15 days following the written request by the issuer of the notice, each shareholder will have the opportunity to notify the other shareholder, by registered letter with proof of receipt, of his intention to sell his entire interest in the company.

The issuer of the notice will indicate the price and the terms of the sale of his interest.

The remaining shareholder must choose, within 30 calendar days of the first presentation of the notice, between the two following options:

- Agree to buy all the shares owned by the offering shareholder, at the price and terms of the offer;
- Or, refuse to buy the shares of the offering shareholder but then have to sell him all his shares in the company at the price and under the conditions set in the initial offer.

In both cases, the decision must be communicated by registered letter with proof of receipt, or any other equivalent means.

3. Traduction en langage informatique

a) Template de la clause « buy or sell *»*

À declarer:

Tokendepaiement

TokenActionERC1644

#Éventuellement le délai

2 protagonistes: un initiateur et un associé désigné

#L’initiatieur active la clause de buy or sell

1

Fonction qui prend en para: l’associe désigne, le prix de l’action, (le délai)

If msg.sender = associe

If il n’a pas déjà utilisé cette fonction

If associe designe = associe

2

→ Tous les tokens représentant les actions de l’initiateur vont dans le *Smart contract* ou mapping, ainsi que ceux de l’associé désigné.

3

Fonction recupereAction (....) :

Si délai pas depasse

Si msg.sender == associe designe

Require (Tokendepaiement. transfer(balance qu’il faut))

Associe designe recupere tous les tokens

Si délai depasse(||si il refuse):

Si msg.sender == initiateur

Require (Tokendepaiement. transfer(balance qu’il faut))

Initiateur récupère tous les tokens

Si délai depasse + certain délai encore depasse:

Si msg.sender == la societe

La societe recupere tous les tokens

b) Pseudo-code

Acheteur commit argent à un séquestre

→ acheteur doit commit plus que le prix du bien pour être incité à confirmer la réception

→ prévoir mécanisme / fonction permettant à l'acheteur de récupérer son argent si vendeur n'a pas envoyé bien / acheteur n'a pas reçu bien après un certain délai.

Acheteur confirme réception du bien

Vendeur récupère argent

c) Code en langage solidity

```
pragma solidity ^0.6.0;

import "interfaces/my_IERC20.sol";
import "interfaces/my_IERC1400.sol";

contract clauseSellorbuy {

// déclaration des différents de l'avis de vente
```

```
enum sellorbuyStates {
STATUS_UNUSED, // par défaut : 0
STATUS_PENDING,
STATUS_CLOSED
}

// Represents a notice of sale parameters
struct NoticeSale {
address offerer; // actionnaire offrant
address remainer; // actionnaire restant
uint256 pricePurchase;
uint256 expirationDate;
sellorbuyStates status;
}

// Mapping from partition UID to Option
mapping ( address => NoticeSale ) public notices;

IERC1400 public tokenStock; // token qui représente la partition (ERC1410)
IERC20 public tokenPayment; // token qui représente le coin (ERC20)

constructor(address _tokenStockAddress, address _tokenPaymentAddress) public {
require(_tokenStockAddress != address(0));
require(_tokenPaymentAddress != address(0));

tokenStock = IERC1400(_tokenStockAddress);
tokenPayment = IERC20(_tokenPaymentAddress);
}
```

```
//------------------------------------------------------------

// Sell or Buy events

//------------------------------------------------------------

event StartSellorbuy(address offerer, address remainer, address escrow, uint256
pricePurchase, uint256 endDate);

//------------------------------------------------------------

// Sell or Buy functions

//------------------------------------------------------------

// get current time

function getCurrentTime() public view returns (uint256) {

return now;

}

// The offerer initiates the Notice of Sale on all the partitions of the remainer

function startSellorbuy(address remainer, uint256 pricePurchase, uint256 duration) public
returns (bool){

uint i=0;

uint256 nbPartition;

nbPartition = tokenStock.getHolderNbuid(remainer);

require(nbPartition > 0);

uint256[] memory uidList = new uint256[](nbPartition);

uidList = tokenStock.partitionsOf(remainer);

for(i=0; i<nbPartition; i++) {

require(tokenStock.getPartitionStatus(uidList[i]) != 1, "Les partitions ne doivent pas être
STATUS_CONFINED, soit 1");
```

```
}

require(pricePurchase >= tokenStock.balanceOf(remainer), "Le prix de l'exercice doit être supérieur ou égal à la valeur des partitions détenues par l'actionnaire restant (remainer)");

require(duration > 0, "la durée d'un avis de vente est exprimée en jours");

require(tokenPayment.balanceOf(msg.sender) >= pricePurchase, "L'actionnaire offrant (offerer) doit posséder suffisamment de token pour acquérir les partitions de l'actionnaire restant (remainer)");

require(notices[remainer].status != sellorbuyStates.STATUS_PENDING, "Il ne peut pas y avoir un avis de vente en cours sur les partitions du remainer");

require(notices[msg.sender].status != sellorbuyStates.STATUS_PENDING, "Il ne peut pas y avoir un avis de vente en cours sur les partitions de l'actionnaire offrant (offerer)");

require(tokenStock.balanceOf(msg.sender) == tokenStock.balanceOf(remainer), "la valeur des partitions de offerer et remainer doit être égale");

notices[remainer].offerer = msg.sender;

notices[remainer].remainer = remainer;

notices[remainer].pricePurchase = pricePurchase;

notices[remainer].expirationDate = now + duration * 1 minutes;

notices[remainer].status = sellorbuyStates.STATUS_PENDING;

// the offerer sends previously a transaction to ERC1400 to approve EscrowAccount on all its partitions (in case of remainer deny)

nbPartition = tokenStock.getHolderNbuid(msg.sender);

uidList = tokenStock.partitionsOf(msg.sender);

for(i=0; i<nbPartition; i++) {

if(uidList[i] != 0) {
```

```
require(tokenStock.allowanceEscrow(msg.sender, address(this), uidList[i]) == true, "le contract de séquestre doit être autorisé à modifier le status de toutes les partitions de l'actionnaire offrant (offerer)");

}

}

// the offerer sends previously a transaction to ERC20 to approve EscrowAccount for the transfer of the priceDelta (in case of remainer accept)

uint256 priceDelta;

priceDelta = pricePurchase - tokenStock.balanceOf(remainer);

require(tokenPayment.allowance(msg.sender, address(this)) >= priceDelta);

// Transfer partitions[partitionUid].owner vers sequestre (address(this)) sur contract ERC1400

// changement status partition en CONFINED (pas de chgt de owner)

// mise à jour de la variable mapping escrows (fait sur le contract ERC1400)

uint256 endDate = now + duration * 1 minutes;

//tokenStock.confinePartition(msg.sender, recipient, partitionUid, endDate, priceExercise);

// the offerer is the origin (msg.sender) of the transactions call

for(i=0; i<nbPartition; i++) {

if(uidList[i] != 0) {

tokenStock.confinePartition(msg.sender, uidList[i], endDate, tokenStock.getPartitionAmount(uidList[i]));

}

}

emit StartSellorbuy(msg.sender, remainer, address(this), pricePurchase, endDate);

return true;
```

```
}

// The remainer accepts during the notice of sale duration

// The partitions are transfered from remainer to offerer

function remainerAccept() public returns (bool){

address offerer = notices[msg.sender].offerer;

require(msg.sender == notices[msg.sender].remainer);

require(tokenPayment.balanceOf(offerer) >= notices[msg.sender].pricePurchase);

require(now <= notices[msg.sender].expirationDate);

require(notices[msg.sender].status == sellorbuyStates.STATUS_PENDING);

// le remainer doit avoir mis au préalable une allowance sur chacune de ses partitions pour autoriser le transfert

uint i=0;

uint256 nbPartition;

nbPartition = tokenStock.getHolderNbuid(msg.sender);

require(nbPartition > 0);

uint256[] memory uidList = new uint256[](nbPartition);

uidList = tokenStock.partitionsOf(msg.sender);

// le offerer doit avoir mis au préalable une allowance sur le contract ERC20 pour autoriser le débit du coût priceDelta lié à l'exercice

uint256 priceDelta;

priceDelta = notices[msg.sender].pricePurchase - tokenStock.balanceOf(msg.sender);

require(tokenPayment.allowance(offerer, address(this)) >= priceDelta);

// Transfer des partitions au coût de pricePurchase de msg.sender vers offerer
```

```
for(i=0; i<nbPartition; i++) {
require(tokenStock.allowanceEscrow(msg.sender, address(this), uidList[i]) == true);
tokenStock.escrowExplicitTransfer(msg.sender, offerer,
tokenStock.getPartitionAmount(uidList[i]), uidList[i]);
}

tokenPayment.transferFrom(offerer, msg.sender, priceDelta);
notices[msg.sender].status = sellorbuyStates.STATUS_CLOSED;

// on déconfine les partitions de offerer
uint nbPart = tokenStock.getHolderNbuid(offerer);
uint256 uidPart;
for(i=0; i<nbPart; i++) {
uidPart = tokenStock.getUid(offerer, i+1);
tokenStock.deconfinePartition(offerer, uidPart);
}

// l'avis de vente est clos
notices[msg.sender].status = sellorbuyStates.STATUS_CLOSED;

return true;
}

// The remainer denies during the notice of sale duration
// The partitions are transfered from offerer to remainer
function remainerDeny() public returns (bool){
address offerer = notices[msg.sender].offerer;
require(msg.sender == notices[msg.sender].remainer);
```

```
require(tokenPayment.balanceOf(notices[msg.sender].remainer) >=
notices[msg.sender].pricePurchase);

require(now <= notices[msg.sender].expirationDate);

require(notices[msg.sender].status == sellorbuyStates.STATUS_PENDING);

// Les partitions de offerer doivent être en séquestre

uint i=0;

uint256 nbPartition;

nbPartition = tokenStock.getHolderNbuid(offerer);

require(nbPartition > 0);

uint256[] memory uidList = new uint256[](nbPartition);

uidList = tokenStock.partitionsOf(offerer);

for (i=0; i<nbPartition; i++) {

require(tokenStock.getPartitionStatus(uidList[i]) == 1, "les partitions de offerer doivent être
en séquestre - STATUS_CONFINED is 1");

}

// on déconfine les partitions de offerer

uint nbPart = tokenStock.getHolderNbuid(offerer);

uint256 uidPart;

for(i=0; i<nbPart; i++) {

uidPart = tokenStock.getUid(offerer, i+1);

tokenStock.deconfinePartition(offerer, uidPart);

}

// le remainer doit avoir mis au préalable une allowance sur le contrat ERC20 pour permettre
le paiement des partitions
```

```
uint256 priceDelta;

priceDelta = notices[msg.sender].pricePurchase - tokenStock.balanceOf(offerer);

require(tokenPayment.allowance(msg.sender, address(this)) >= priceDelta);

// Transfer des partitions de offerer vers msg.sender

for(i=0; i<nbPartition; i++) {

//require(tokenStock.allowanceEscrow(offerer, address(this), uidList[i]) == true);

tokenStock.escrowExplicitTransfer(offerer, msg.sender,
tokenStock.getPartitionAmount(uidList[i]), uidList[i]);

}

tokenPayment.transferFrom(msg.sender, offerer, priceDelta);

// l'avis de vente est clos

notices[msg.sender].status = sellorbuyStates.STATUS_CLOSED;

return true;

}

// The controller forces the transfer after the notice of sale duration

// The partitions are transfered from remainer to offerer (or controller)

function controllerForce(address remainer) public returns (bool){

require(notices[remainer].status == sellorbuyStates.STATUS_PENDING);

require(now > notices[remainer].expirationDate);

require(tokenStock.getHolderStatus(tx.origin) == 1, "le compte originaire de la transaction
doit être le controller");

// le offerer doit avoir positionné une allowance sur ERC20, sinon le transfert est effectué
sur le compte du controller

address offerer = notices[remainer].offerer;
```

```
uint256 priceDelta;
priceDelta = notices[remainer].pricePurchase - tokenStock.balanceOf(offerer);
require(tokenPayment.allowance(offerer, address(this)) >= priceDelta);

// on déconfine les partitions de offerer
uint i=0;
uint256 nbPartition;
uint256 uidPartition;
nbPartition = tokenStock.getHolderNbuid(offerer);
for(i=0; i<nbPartition; i++) {
uidPartition = tokenStock.getUid(offerer, i+1);
if(uidPartition != 0) {
tokenStock.deconfinePartition(offerer, uidPartition);
}
}

// Le controller transfert les partitions de remainer vers offerer
nbPartition = tokenStock.getHolderNbuid(remainer);
require(nbPartition > 0);
for (i=0; i<nbPartition; i++) {
uidPartition = tokenStock.getUid(remainer, i+1);
if(uidPartition != 0) {
tokenStock.controllerTransfer(remainer, offerer, uidPartition);
}
}

// l'avis de vente est clos
```

```
notices[remainer].status = sellorbuyStates.STATUS_CLOSED;

return true;

}

// Le controller annule l'avis de vente

function controllerRemove(address remainer) public returns (bool){

require(tokenStock.getHolderStatus(tx.origin) == 1, "le compte originaire de la transaction doit être le controller");

// boucle sur les partitions de Alice (offerer)

uint i=0;

uint256 nbPartition;

uint256 uidPartition;

address offerer = notices[remainer].offerer;

nbPartition = tokenStock.getHolderNbuid(offerer);

for(i=0; i<nbPartition; i++) {

uidPartition = tokenStock.getUid(offerer, i+1);

if(uidPartition != 0) {

tokenStock.deconfinePartition(offerer, uidPartition);

}

}

uint256 delta = notices[remainer].pricePurchase - tokenStock.balanceOf(remainer);

if (delta != 0) {

tokenPayment.decreaseAllowanceFrom(offerer, delta);

}

notices[remainer].expirationDate = now;
```

```
notices[remainer].status = sellorbuyStates.STATUS_CLOSED;

}

}
```

E. Clause de préemption

1. Présentation juridique

Définition de la clause. La clause de préemption prévoit d'accorder une préférence à un ou plusieurs actionnaires par un promettant lui aussi actionnaire, pour la cession de ses titres. Souvent, les bénéficiaires de cette clause sont l'ensemble des actionnaires lorsqu'elle est prévue dans les statuts, ou les signataires du pacte lorsque celle-ci est prévue dans un pacte d'actionnaires. Dans ce dernier cas, cette clause est dite extrastatutaire. La clause de préemption est couramment utilisée pour éviter la dilution des participations capitalistiques des actionnaires bénéficiaires de la préférence. En quelque sorte, ce mécanisme contractuel permettra la maîtrise du capital et de ses détenteurs, ce qui implique une forme de stabilité en dépit des cessions de titres qui peuvent intervenir, ainsi qu'un contrôle des droits politiques et financiers qui en découlent. La clause de préemption, telle qu'elle est pratiquée en droit des sociétés, n'est soumise à aucun texte légal ou réglementaire. Elle est régie par la liberté contractuelle dévolue aux parties par les règles du droit commun des contrats (article 1102 du Code civil).

Régime juridique. Il n'y a aucun régime légal propre aux clauses de préemption. Les règles de droit commun des contrats s'appliqueront faute de règle spéciale, notamment au moment de la formation (capacité, consentement, contenu), auxquelles il faudra ajouter les règles du droit commun de la vente, particulièrement la détermination du prix de cession des titres (Article 1124 du Code civil – Articles 1591 et 1592 du Code civil – Article 1843-4 du Code civil lorsqu'un tiers intervient). Conformément à l'esprit libéral qui anime toute clause, la jurisprudence commerciale considère la clause de préemption comme licite. Cependant, des limites ont été posées par les tribunaux qui considèrent que ce type de clause ne doit pas porter atteinte à l'ordre public sociétaire (dispositions statutaires et/ou l'intérêt social et ce indépendamment de toute clause statutaire ou extrastatutaire). Enfin, dans le contexte de sociétés cotées, la clause doit être publiée, par une communication à l'Autorité des marchés financiers (AMF) (article L.233-11 du Code de commerce). Il est recommandé de prévoir une

durée déterminée à la clause de préemption, ainsi que sa reconduction. Une durée indéterminée expose la clause de préemption à une résiliation unilatérale de l'une ou l'autre des parties, ce qui contreviendrait à l'automaticité de la clause recherchée. Le prix de cession devra être déterminé préalablement à la mise en œuvre de la clause de préemption, soit par les parties dans le cadre d'un pacte, soit par le recours à un expert et selon les modalités entérinées dans le pacte.

Exemple de rédaction

Tout transfert de titres réalisé par l'un ou l'autre actionnaire (ci-après dénommé le « **Cédant** ») sera soumis à un droit de préemption (ci-après dénommé le « **Droit de Préemption** ») au profit des autres actionnaires (ci-après dénommé le/les « **Bénéficiaire(s)** »).

Tout projet de transfert devra être notifié par le Cédant au Bénéficiaire (ci-après dénommée la « **Notification** »), au moins 30 jours avant la date prévue pour la réalisation de la transmission, par lettre recommandée avec accusé de réception (ou par lettre remise en mains propres), avec l'indication :

- du nombre et de la nature des titres dont le transfert est projeté,

- des nom, prénom et domicile ou dénomination et siège social de chacun des bénéficiaires du transfert, ainsi que, s'il s'agit d'une personne morale, des nom, prénom et domicile ou dénomination et siège social des personnes qui, le cas échéant, la contrôlent

- du prix ou de la valeur retenue pour le transfert considéré,

- des modalités de paiement de prix (notamment son délai) et de toutes autres conditions du transfert et, notamment, la date de réalisation prévue, laquelle ne pourra être fixée avant l'expiration du délai de 30 jours prévu à l'article X des présentes.

Une copie de l'engagement d'acquisition du Bénéficiaire du transfert devra être jointe à la Notification, accompagnée de son engagement d'adhérer au pacte d'actionnaires, concomitamment au transfert.

Le Bénéficiaire disposera d'un délai de 30 jours à compter de la réception de la Notification qui lui a été faite selon les modalités prévues à l'article XX des présentes, pour notifier au Cédant (ci-après dénommée la « **Préemption** »), par lettre recommandée avec accusé de réception (ou par lettre remise en mains propres), qu'il entend exercer ou non son Droit de Préemption, ainsi que le nombre de titres qu'il souhaite préempter (à titre irréductible et, le cas échéant, à titre réductible).

Chaque Préemption sera inconditionnelle et irrévocable pour le nombre de titres maximums disponibles.

L'absence de réponse du Bénéficiaire dans ce délai de 30 jours vaudra renonciation à exercer son Droit de Préemption.

Par le seul fait de la Préemption, le transfert sera réalisé au profit du Bénéficiaire, sous réserve du paiement du prix de cession (ou de la partie du prix de cession payable comptant) au jour de la remise des ordres de mouvement et autres pièces nécessaires, le tout par dérogation aux dispositions de l'article 1583 du Code civil, à un prix égal à celui proposé par le tiers acquéreur ou résultant des conditions du transfert envisagé.

2. Traduction en anglais

Definition of the clause. The pre-emption clause (also known as the Right of First Refusal clause) gives existing shareholders the right to buy the shares of an owner who is selling before a third party can do. This clause will benefit all shareholders when it is included in the articles of association of the company but only the parties to a shareholders' agreement when it is included in a shareholders' agreement.
The pre-emption clause is often used to avoid dilution of existing shareholdings, ensuring that the relative control of the company remains the same. The clause enables the shareholders to maintain control on who is a shareholder in the company and how power and profits are shared.

Legal framework. There is no specific legal framework for pre-emption clauses. The general law of contract will apply, notably during contract formation (legal capacity to contract, consent, content of the contract). The general sales legislation will also apply, in particular in setting the sales price of the shares (Article 1124 of the Civil Code; Articles 1591 and 1592 of the Civil Code; and Article 1843-4 of the Civil Code when a third party is involved in the determination of the price). Although the courts consider pre-emption clauses to be lawful, they have imposed some limits so that they comply with the general principles of company law.

Lastly, for listed companies, the clause needs to be published, via a notification to the Financial Market Authority (Autorité des Marchés Financiers (AMF)), the French stock market regulator (Article L. 233-11 of the Commercial Code). It is advisable to put a time limit on any pre-emption clause and on its renewal. Without a time limit, a pre-emption clause could face being unilaterally terminated by one of the parties, which would prevent the desired automatic nature of the clause. The sales price would need to be set before the implementation of the clause, either by the parties in the shareholders' agreement or by using an expert under the conditions set in the said agreement.

Examples of drafting

Any transfer of shares done by one shareholder (thereafter the "Seller") is subject to pre-emption right (thereafter the "Pre-emption right") to the benefit of the other shareholders (thereafter the "Beneficiary/ies").

The Seller needs to notify any proposed transfer to the other shareholder(s) at least 30 days before the transfer of the shares, by registered letter with proof of receipt (or delivered by hand). The notification needs to specify:

- the number and type of shares he wishes to transfer,
- the surname, name and address or corporate name and head office the proposed transferee(s), as well as, in the case of a legal person, the surname, name and address or corporate name and head office of the owners,
- the proposed sale price at which he wishes to transfer the shares,

- the conditions of the payment (notably the payment period) and any other conditions to the payment and, notably, the completion date of the transfer, which cannot be set before the end of the 30-day period set in clause [X].

A copy of the proposed transferee's commitment to the transfer as well as his commitment to join the shareholders' agreement, upon the transfer of the shares, must be added to the transfer notice.

The Beneficiary of a pre-emption right will have 30 days, from receipt of the notification issued pursuant to clause [X] of this shareholders' agreement, to notify the Seller, by registered letter with proof of receipt (or delivered by hand), of his intent to use or not his pre-emption rights, and the number of shares he wishes to pre-empt the purchase of.

Each pre-emption will be unconditional and irrevocable for the maximum number of available shares.

If the Beneficiary does not respond to the transfer notification within the prescribed 30 days, he will be deemed to have waived his pre-emption right.

The pre-emption will complete the transfer of the shares to the beneficiary provided he pays the sales price (or the portion of the sales price that is payable in cash) on the day of the presentation of transfer orders and necessary documentation, notwithstanding the provisions of article 1583 of the Civil Code, for the same price as the third-party offered to purchase or for the price resulting from the conditions of the proposed transfer.

Pre-emption clause

Transfer orders and any other necessary document must, within 90 days of the Pre-emption exercised by the Beneficiary, be forwarded to him, against payment of the sales price (or of the portion of the sales price that is payable in cash).

If the Beneficiary does not pre-empt all or part of the shares the Seller wants to transfer, the Seller is free to transfer all the shares he intended to transfer to the proposed third-party transferee, under the conditions specified in this article, provided the third-party transferee is approved by the other shareholders pursuant to clause [XX].

Furthermore, where the Beneficiary of the pre-emption right does not exercise his right, he explicitly commits, as a shareholder of the company, and in due time, to approve the third-party transferee as a new shareholder of the company.

In the absence of full or partial pre-emption by one of the shareholders, the approval clause remains applicable and the Seller must abide by it.

3. Traduction en langage informatique

a) Template de la clause de préemption

La cession des actions, représentées par le *token* suivant : [...] est soumise à un droit de préemption des actionnaires, exercé par l'intermédiaire d'un *Smart contract*.

L'actionnaire souhaitant transférer ses titres à une tierce personne s'engage à notifier son projet de cession à la Société en indiquant l'identification du cessionnaire, le nombre de Titres dont la cession est envisagée et le prix par action offert.

Il peut ensuite initier le transfert à travers le *Smart contract*.

La Société devra notifier le projet de cession aux associés dans un délai de [...] jours en les informant des conditions d'exercice de leur droit de préemption à travers le *Smart contract*.

Ils disposeront alors de la faculté de préempter les titres proposés à la cession, au prix offert par le cessionnaire et indiqué dans le courrier de notification du projet de cession.

Tout associé souhaitant exercer son droit de préemption disposera d'un délai de 90 jours pour le faire, en indiquant le nombre de titres dont il souhaite faire l'acquisition à travers le *Smart contract*.

L'exercice du droit de Préemption par un ou plusieurs associés doit porter sur l'intégralité des actions concernées.

Dans le cas où les demandes de préemption dépasseraient le nombre d'Actions proposées, celles-ci seront réparties entre les associés au prorata de leur participation dans le capital de la Société et dans la limite de leurs demandes respectives, le solde étant réparti entre les associés dont les demandes n'auront pas été complètement satisfaites au prorata de leurs participations respectives dans le capital de la société.

Les titres non préemptés pourront être cédés aux conditions initiales de cession établi par l'associé cédant. Ces titres seront alors cédés par leur retrait, dans le *Smart contract*, au cessionnaire initial.

b) Pseudo-code

token qui représente action variable qui donne le nombre total de token
[A] veut transférer un nombre d'action de token à un quelqu'un Le transfert déclenché par [A] envoie les tokens dans un *Smart contrat.*
fonction pour les associés souhaitant exercer leur préemption (nbre d'action qu'ils veulent, token d'argent){

si [A] a au moins un token représentant une action c'est que c'est un associé (if msg.sender balance token rpst action > 1) :
si [A] n'a jamais interragi avec ce *Smart contract* :

son interaction avec le *Smart contract* incrément une variable nbre actions préemptées (var nbre actionspreemptes =+ 1)

dans un mapping on enregistre son adresse et le nombre d'actions qu'il veut mapping (address msg.sender → nbre d'action qui veut)

dans un autre mapping on enregistre adresse et argent

}

fonction répartition préemption (){

si délai dépassé :

si nbreaction preemptes = nbre action token:

transférer token action aux associés en l'échange de l'argent (transférer actions mapping[adress msg.sender] → nbre actions qui veut)

si nbreaction preemptes > nbre action token :
t
owing : (mec token balance / total token) * 100

(nbreaction * owing)/100

etc

}

if délai dépassé :

if nbreaction preemptes > nbre action transfere :

if owing > mapping [adress msg.sender] → nbre actions qu'ils veut :
owing = actions qu'il veut
sinon donner owing
}

Fonction transfertAction (nbreActionsVendues, prixAction, cessionnaire){

Vérifier que cessionnaire =/= associé
If (cessionnaire=/=associé): {
Transférer au SC les actions

Enregistrer variable nbreActions
Enregistrer variable prixActions

Event ouvert a la preemption (prixAction, nbreActions)
}
Else {
Transférer au SC en attente d'être retiré par le cessionnaire associé
}

}

Fonction exercicePreemption (token de paiement, nbreActions){

Vérifie que on est dans le délai

Vérifie que le msg.sender == associé de la Société
Vérifie que le msg.sender n'a pas déjà utilisé cette fonction

Associé peut envoyer son token de paiement au SC + nbre d'actions qu'il veut au SC
(vérifier qu'il envoie le token de paiement correspondant au nbre d'actions qu'il veut)

Mapping (adresse de l'associé → token de paiement)
Mapping (adresse de l'associé → nbre d'actions qu'il veut)

Le nombre d'actions voulu par l'associé incrémente une variable nbreActionsPreemptes

Fonction repartitionPreemption () {

Vérifie que délai de préemption soit écoulé
Vérifie que msg.sender = société

Si nbreActionsPreemptes < nbreActionsVendues:

→ échec de la préemption: transfère les token de paiement aux associés;
Si nbreActionsPreemptes = nbreActionsVendues:

→ Les associés reçoivent les actions qu'ils ont demandés et

Si nbreActionsPreemptes >nbreActionsVendues :

→ les associés reçoivent : (nbreActionsVendues * % de leur détention de la société)/100
Si résultat de l'opération > à ce que associé x avait demandé, retenir ce que l'associé x avait demandé.

⇒ transférer actions au prorata + solde

event...

Fonction recupererArgent () {

Vérifie que msg.sen
⇒ transférer tokens de paiement à l'associé cédant après avoir vérifié que répartition a été faite dans le cas d'exercice préemption

⇒ transférer token de paiement à l'associé cédant après avoir vérifié cessionnaire a bien mis son collatéral dans autre contrat / fonction s'il n'est pas associé.
}

c) Code en langage solidity

```
pragma solidity ^0.6.0;

import "interfaces/my_IERC20.sol";

import "interfaces/my_IERC1400.sol";

contract clausePreemption {

// Represents a preemption parameters

struct PreemptionRight {

address promisor;

address recipient;

uint256 expirationDate;

}

struct PreemptionResponse {

address promisor;
```

```
address recipient;
uint256 partitionUid;
uint256 pricePreemption;
uint256 expirationNotice;
uint256 expirationDate;
uint256 decisionDate;
bool decision;
}

// Mapping from partition UID to Option
mapping ( address => mapping (uint256 => PreemptionRight)) public preemptions;

mapping (address => uint256) public nbRecipient;
mapping (address => mapping (uint256 => PreemptionResponse)) public responses;

IERC1400 public tokenStock; // token qui représente la partition (ERC1410)
IERC20 public tokenPayment; // token qui représente le coin (ERC20)

//address public recipient; // le bénéficiaire de la préemption
//address public promisor; // le promettant
//uint256 public expirationDate; // la date de fin du droit de premption

constructor(address _tokenStockAddress, address _tokenPaymentAddress) public {
require(_tokenStockAddress != address(0));
require(_tokenPaymentAddress != address(0));

tokenStock = IERC1400(_tokenStockAddress);
tokenPayment = IERC20(_tokenPaymentAddress);
}
```

```
//-------------------------------------------------------------

// Preemption right events

//-------------------------------------------------------------

event StartPreemption(address indexed _promisor, address indexed _recipient, address escrow, uint256 _expirationDate);

//-------------------------------------------------------------

// Preemption right functions

//-------------------------------------------------------------

// The promisor initiate the Preemption for a given recipient

function startPreemption(address recipient, uint256 duration) public returns (bool){

require(tokenStock.getHolderStatus(msg.sender) == 0 , "Le émetteur (promisor) doit être un détenteur de titres");

require(tokenStock.getHolderStatus(recipient) == 0, "Le bénéficiaire (recipient) doit être un détenteur de titres");

require(duration > 0, "la durée d'une premption est exprimée en minutes");

uint256 index;

if( preemptions[msg.sender][0].promisor == address(0) ) {

nbRecipient[msg.sender] = 0;

}

preemptions[msg.sender][index].promisor = msg.sender;

preemptions[msg.sender][index].recipient = recipient;

uint256 endDate;

endDate = preemptions[msg.sender][index].expirationDate = now + duration * 1 minutes;
```

```
index = index + 1;

nbRecipient[msg.sender] = index;

emit StartPreemption(msg.sender, recipient, address(this), endDate);

return true;

}


function launchNotice(address recipient, uint256 partitionUid, uint256 pricePreemption,
uint256 duration) public returns (bool){

require(tokenStock.getHolderStatus(msg.sender) == 0 , "Le émetteur (promisor) doit être un
détenteur de titres");

require(tokenStock.getHolderStatus(recipient) == 0, "Le bénéficiaire (recipient) doit être un
détenteur de titres");

require(tokenStock.getPartitionOwner(partitionUid) == msg.sender, "Le émetteur doit être le
propriétaire de la partition");


uint256 index = nbRecipient[msg.sender];

uint j;

for ( uint i=0; i< index; i++) {

if ( preemptions[msg.sender][i].recipient == recipient ) {

j = i;

i = index;

}

}

require(preemptions[msg.sender][j].expirationDate > now);

require(tokenStock.allowanceEscrow(msg.sender, address(this), partitionUid) == true);


responses[recipient][partitionUid].promisor = msg.sender;
```

```
responses[recipient][partitionUid].recipient = recipient;

responses[recipient][partitionUid].partitionUid = partitionUid;

responses[recipient][partitionUid].pricePreemption = pricePreemption;

responses[recipient][partitionUid].expirationNotice = now + duration * 1 minutes;

responses[recipient][partitionUid].expirationDate =
preemptions[msg.sender][j].expirationDate;

responses[recipient][partitionUid].decisionDate = 0;

responses[recipient][partitionUid].decision = false;

}


// The recipient notifies his decision

function recipientDecision(uint256 partitionUid, bool decision) public returns (bool){

require(msg.sender == responses[msg.sender][partitionUid].recipient);

require(responses[msg.sender][partitionUid].expirationNotice >= now);

require(responses[msg.sender][partitionUid].expirationDate >= now);


responses[msg.sender][partitionUid].decisionDate = now;

responses[msg.sender][partitionUid].decision = decision;

return true;

}


function launchTransfer(address recipient, uint256 partitionUid) public returns (bool){

require(now >= responses[recipient][partitionUid].decisionDate, "now est postérieur à
responses.decisionDate");

require(tokenStock.getPartitionOwner(partitionUid) == msg.sender, "msg.sender doit être le
owner de la partition");
```

```
require(responses[recipient][partitionUid].decision == true, "le recipient doit être bénéficiaire et avoir notifié une décision positive");

require(tokenStock.allowanceEscrow(msg.sender, address(this), partitionUid) == true, "msg.sender a positionné une allowance à this sur la partition");

require(tokenPayment.allowance(recipient, address(this)) >= responses[recipient][partitionUid].pricePreemption, "le recipient a positionné une allowance à this pour le coût de la préemption");

//require("le recipient a positionné une allowance à ERC1400 pour le coût de la partition");

// paiment du coût de la préemption

tokenPayment.transferFrom(recipient, msg.sender, responses[recipient][partitionUid].pricePreemption);

// transfert of the partition from promisor (msg.sender) to recipient

tokenStock.escrowFreeTransfer(recipient, tokenStock.getPartitionAmount(partitionUid), partitionUid);

}

}
```

Section 2 : Test sur les *smart contracts*

Dans le cadre de ce projet, différentes clauses juridiques implémentées à l'aide de *smart contracts* sur la *blockchain* Ethereum sont soumises à une série de tests fonctionnels dans le but de s'assurer du bon fonctionnement de ces clauses en interaction les unes avec les autres.

Pour cette phase de test, la *blockchain* Ethereum est simulée par le module Python open source *ganache-cli*.

L'environnement de compilation, d'exécution et de test est géré par le module Python open source *brownie*.

Paragraphe 1 : Rappel synthétique des clauses pour la réalisation des tests

Avant d'entrer dans la partie technique du sujet, le fonctionnement et le cas d'usage des clauses qui vont être testées sont succinctement présentés ci-dessous. Seule la clause d'exclusion n'a pas pu faire l'objet de test, faute de temps.

A. Clause d'option

La clause d'option implique deux acteurs, le détenteur de la partition (Alice) et le bénéficiaire (Bob). Lorsqu'Alice accorde à Bob un droit d'option sur l'une de ses partitions, elle accepte de geler sa partition pendant un certain temps, la durée de l'option, pendant lequel Bob est invité à se positionner quant à l'acquisition de la partition d'Alice. Le droit d'option a un coût, en supplément du montant des partitions pouvant être transférées à Bob s'il décide d'acheter. Au-delà de la durée de l'option, si Bob ne s'est pas prononcé, la partition est dégelée et Alice peut en disposer à sa guise.

B. Clause de vente à terme

La clause de vente à terme implique deux acteurs, le détenteur de la partition (Alice) et le bénéficiaire (Bob). Lorsqu'Alice accorde à Bob un droit de vente à terme sur l'une de ses partitions, elle accepte de geler sa partition pendant un certain temps, la durée avant le terme, durée après laquelle Bob est libre d'acheter la partition à Alice pour son prix d'exercice plus un surcoût fixé au moment de l'émission du droit de vente (typiquement en prévision d'une augmentation de la valeur de l'action). La partition est alors transférée d'Alice à Bob contre la somme prévue. Si Bob refuse de terminer la vente après le terme, Alice garde sa partition qui est dégelée ainsi que le surcoût promis par Bob.

C. Clause de préemption

La clause de préemption implique deux acteurs ou plus et prend réellement son sens à partir de trois acteurs, un détenteur d'une partition (Alice) souhaitant vendre son action et des acheteurs potentiels (ici Bob et Charles). La clause de préemption permet à Alice d'émettre un avis de

vente à destination des acheteurs potentiels qu'elle souhaite pour une partition qu'elle possède (qui est alors gelée), les potentiels acheteurs peuvent ensuite indiquer leur volonté d'acheter ou non la partition au prix demandé dans le temps imparti fixé par Alice. Une fois leur réponse donnée, Alice choisit à sa discrétion l'acheteur à qui elle vend effectivement la partition, parmi ceux ayant donné une réponse positive. La partition est alors dégelée et transférée à l'acheteur choisi contre la somme prévue en *token* de paiement ERC20.

D. Clause « *buy or sell* »

La clause « *buy or sell* » implique deux acteurs (toujours Alice et Bob), tous deux détenteurs de la même valeur d'actions, c'est-à-dire que les valeurs cumulées en tokens de paiement ERC20 des partitions ERC1400 de chacun sont égales. La clause « *buy or sell* » est utile lorsqu'Alice et Bob ne peuvent plus continuer à partager la propriété des actions en question, Alice peut alors faire appel à cette clause pour émettre un avis d'achat ou vente portant sur la totalité de ses actions à destination de Bob, valable pour une durée donnée et au prix d'exercice de son choix supérieur à la valeur des partitions. Bob peut alors accepter l'offre et vendre à Alice la totalité de ses partitions au prix proposé, au contraire si Bob refuse l'offre ou s'abstient de répondre dans les délais il est contraint d'acheter la totalité des partitions d'Alice au prix initialement proposé par cette dernière.

Paragraphe 2 : Hypothèses de développement

Lors de la phase d'initialisation, les utilisateurs se voient crédités des *tokens* de paiement qui correspondraient à un investissement de leur part, mais qui, dans notre cas, sont fictifs.

> **Les *tokens* sont gérés par le *smart contract* ERC20, accessibles à tous les utilisateurs.**
>
> **Le *smart contract* ERC1400 gère les actions (partitions) pour un groupe d'utilisateurs enregistrés.**
>
> **Les mouvements de *tokens* sont effectués sur le *smart contract* ERC20.**
>
> **Les transferts, acquisitions, ventes, et gels de partitions sont effectués sur le *smart contract* ERC1400.**

Pour mettre en œuvre les clauses sous forme de *smart contracts* et permettre leur exécution automatique, un groupe d'utilisateurs doit être enregistré sur un *smart contract* de partitions ERC1400 représentant les actions de leur entreprise. Pour permettre l'acquisition d'actions, une méthode d'achat de partitions est implémentée. D'autres techniques d'acquisition de partitions pourront être mises en œuvre selon les cas d'usage considérés.

Les *smart contracts* qui codent l'exécution automatique des clauses sont enregistrés avec le rôle d'*escrow*[33], autrement dit de séquestre informatique, sur le *smart contract* ERC1400. Les utilisateurs lui délèguent la possibilité d'agir sur une de leur partition dans le cadre des méthodes définies dans le *smart contract* et pour un temps déterminé.

Nous avons fait le choix d'implémenter le séquestre d'une partition sous la forme d'un gel. Cela présente l'avantage de conserver la partition sur le compte de son possesseur. En revanche, celui-ci ne peut pas agir dessus pendant la durée du gel. C'est le *smart contract* d'*escrow* qui lève le séquestre, en dégelant la partition lorsque les conditions de la clause sont réunies.

Les *smart contracts* sont déployés dans la *blockchain* une seule fois. Lorsqu'une clause est initiée, une transaction est envoyée vers la méthode permettant d'instancier la clause stipulant les parties prenantes, les partitions ciblées, les coûts et la durée.

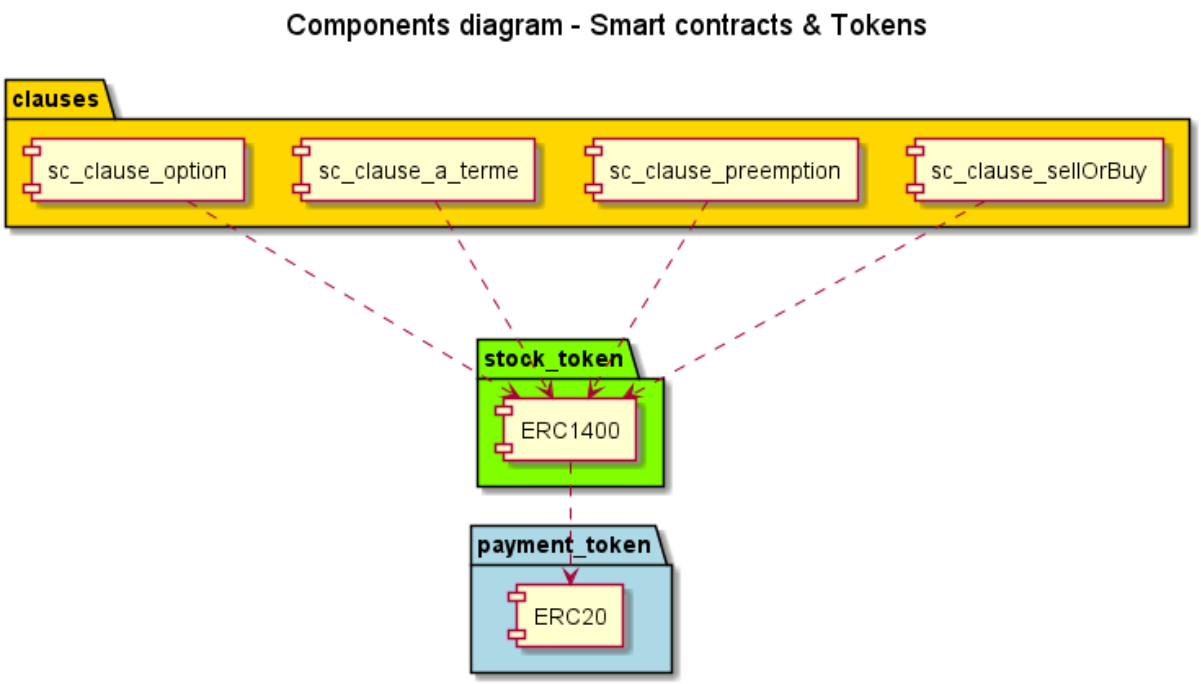
Par convention, le développeur agit depuis le compte de portefeuille account[0].

Composants de la librairie

La librairie de *smart contracts* est composée : - d'un *smart contract* de *tokens* de paiement, ERC20 - d'un *smart contract* de *tokens* de partitions, ERC1400, qui dépend du *token* de paiement - d'un *smart contract* par clause qui dépend des deux *smart contracts* de *tokens* précédents.

[33] Séquestre informatique, voir définition lexique p. 215.

Le diagramme suivant représente l'organisation de la librairie :



Tests fonctionnels

Déploiement des *smart contracts*

Les scénarios mis en œuvre pour effectuer les tests des différentes clauses comportent tous les étapes préliminaires suivantes.

Les comptes utilisateurs (généralement Alice et Bob) et développeur sont déclarés et chargés depuis le simulateur *ganache-cli*.

Les *smart contracts* de *tokens* (ERC20), de partitions (ERC1400) et de la clause à tester sont déployés dans la *blockchain* simulée par *ganache-cli*.

Chaque acteur acquiert des *tokens* de paiement ERC20.

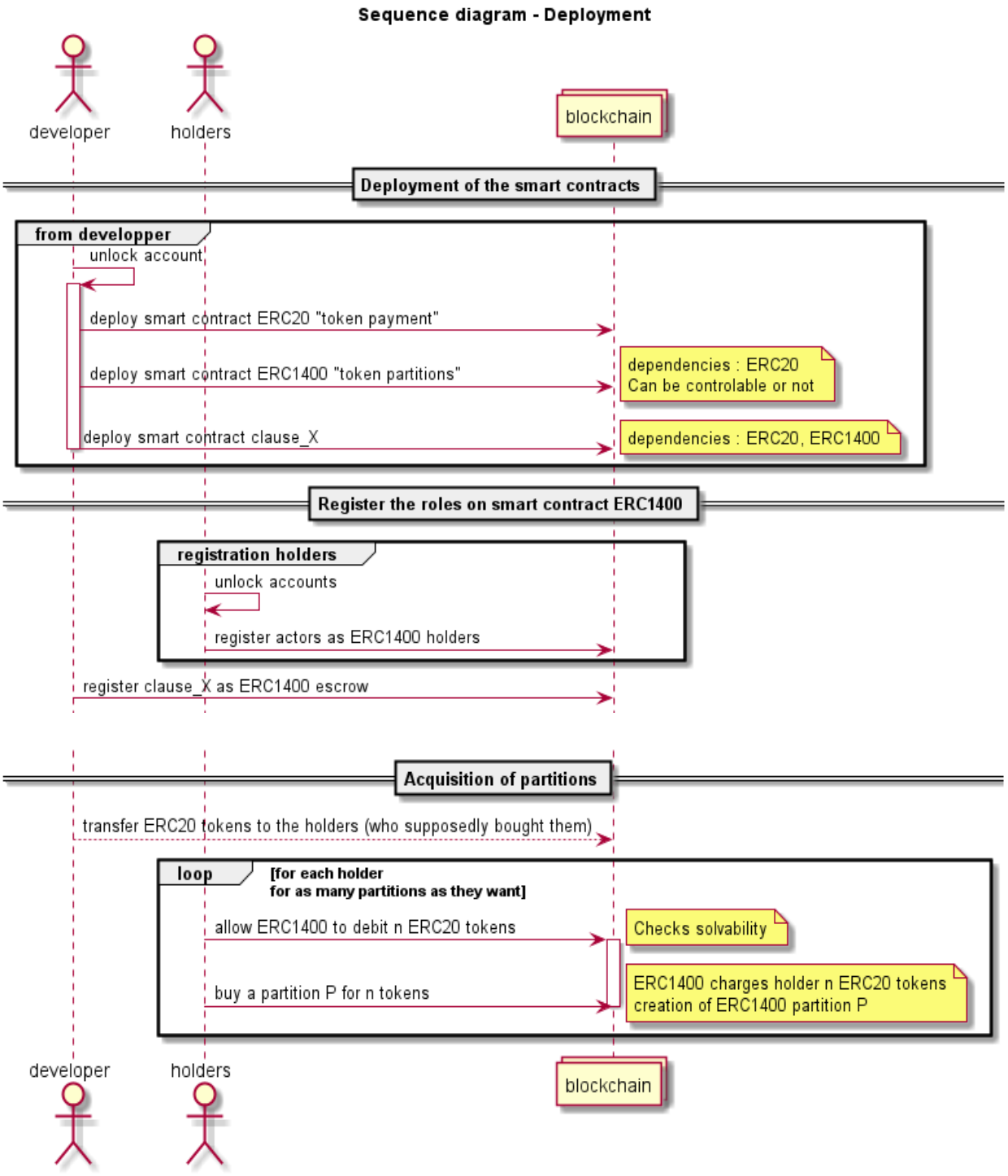
Les acteurs amenés à détenir des partitions s'enregistrent en tant que *holders* sur le *smart contract* de partition ERC1400.

Le *smart contract* de la clause à tester est enregistré comme *escrow* sur le *smart contract* de partition ERC1400 par le développeur.

Les acteurs souhaitant investir dans des partitions positionnent les autorisations nécessaires au débit de *tokens* de paiement ERC20 correspondant.

Les acteurs enregistrés investissent dans les partitions souhaitées (dans notre cas les partitions utiles aux tests).

La séquence des opérations est illustrée par le diagramme suivant :



Au cours de ces phases d'initialisation sont testées les différentes méthodes de ces *smart contracts* utiles au déploiement, à l'enregistrement des rôles par les *smart contracts*, au positionnement d'autorisations et à l'achat et transfert de *tokens*. À partir de valeurs données, ces tests vérifient que le comportement du système et l'état des variables concernées sont bien ceux attendus.

A. Clause d'option

Alice est détentrice d'une partition *P* sur laquelle Bob souhaite mettre une option pour l'acheter au prix d'exercice *e* supérieur à la valeur *n* de la partition, l'option à un coût *x* et est valable pour une durée *d*.

Déclaration du droit d'option

1. Bob autorise le *smart contract clause_option* à le débiter de *x tokens* de paiement ERC20.
2. Alice autorise le *smart contract clause_option* à séquestrer sa partition *P*.
3. Alice lance le droit d'option à destination de Bob, sa partition *P* est alors gelée pour la durée *d* de l'option et Bob est débité du coût *x* de l'option au profit d'Alice sur le *smart contract* de paiement ERC20. Alice est toujours propriétaire de la partition, mais ne peut en disposer.

Résolution de l'option

Bob accepte dans le temps imparti

Si Bob souhaite accepter l'offre d'Alice, il doit au préalable autoriser le *smart contract* de partitions ERC1400 à le débiter du prix *e* de la partition en *tokens* de paiement ERC20.

Il indique ensuite au *smart contract clause_option* sa décision d'acquérir la partition *P*, le tout avant que la durée *d* de l'option ne soit écoulée. La partition *P* est alors dégelée et transférée de Alice à Bob, Bob lui est débité du prix *e* de la partition en *tokens* de paiement ERC20 au profit d'Alice.

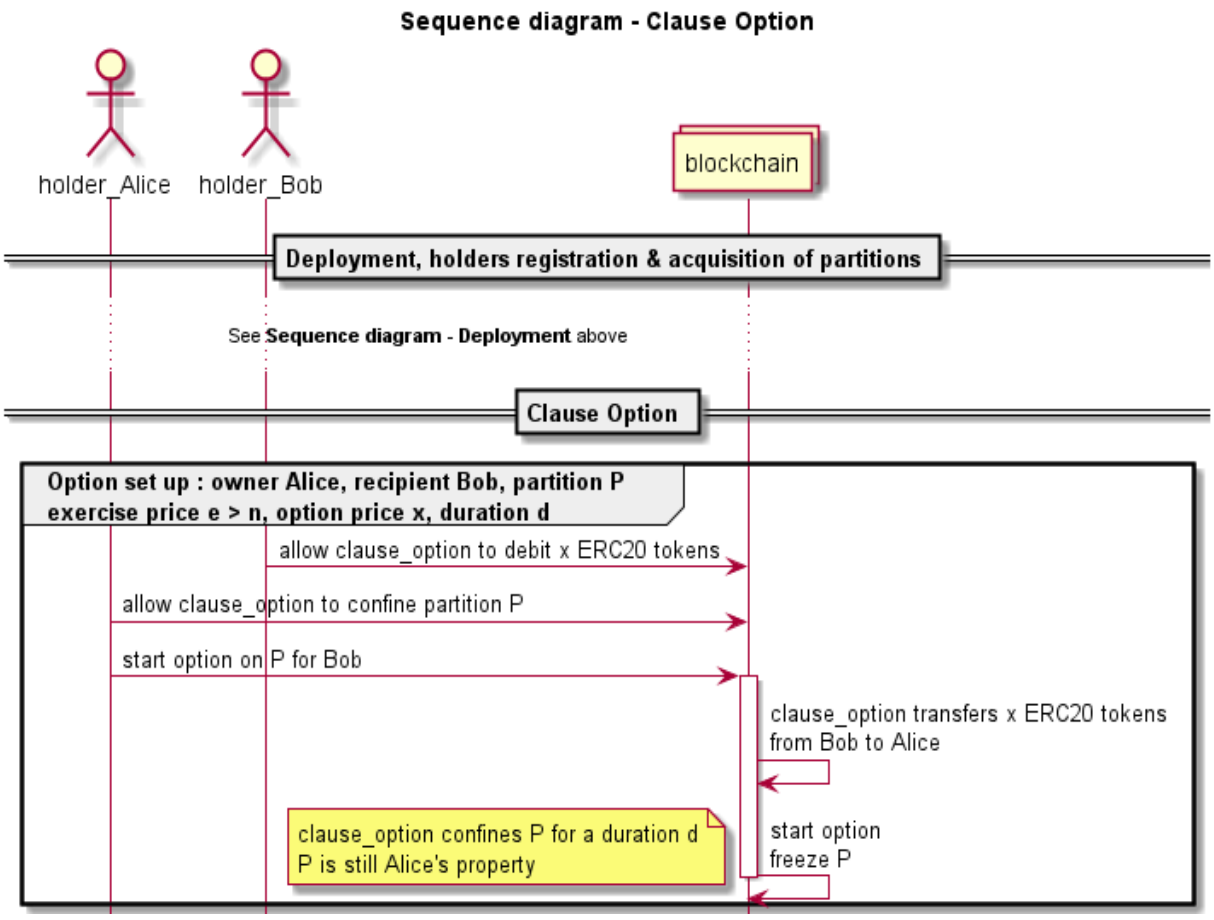
Bob refuse dans le temps imparti

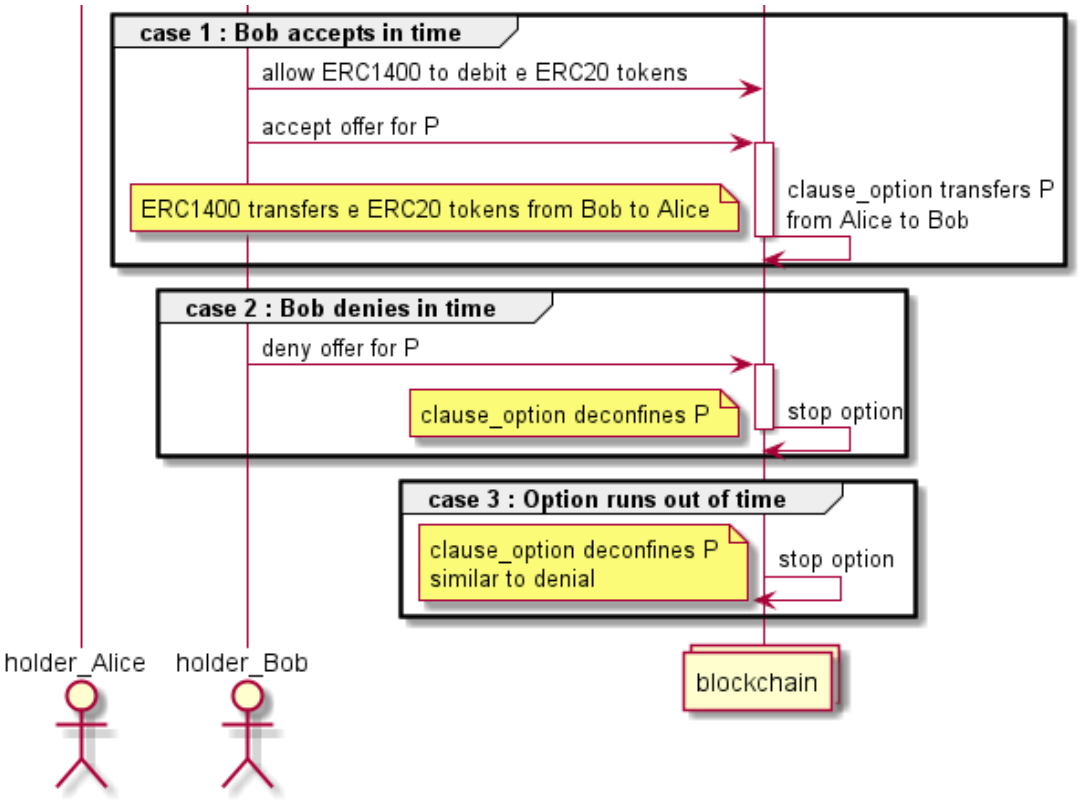
Si Bob ne souhaite pas accepter l'offre d'Alice, il indique simplement au *smart contract clause_option* son refus dans le temps imparti *d*, le droit d'option est alors levé et Alice reste propriétaire de la partition *P* qui est dégelée.

L'option expire

Si Bob reste indécis trop longtemps et n'indique pas son choix avant que la durée *d* de l'option ne soit écoulée, le droit d'option est levé et Alice peut à nouveau disposer de sa partition comme en cas de refus.

La séquence des opérations est illustrée par le diagramme suivant :





Au cours de ce déroulement, chaque cas de figure est testé afin de s'assurer que les *smart contracts* gèrent les différentes possibilités comme ils le doivent et que chaque étape intermédiaire fonctionne correctement.

B. Clause à terme

Alice est détentrice d'une partition *P* que Bob souhaite acheter après une durée *d* au prix d'exercice *e* supérieur à la valeur *n* de la partition, avec un surcoût de *x* au terme.

Déclaration du droit de vente

1. Bob autorise le *smart contract clause_forward* à le débiter de *x tokens* de paiement ERC20.
2. Alice autorise le *smart contract clause_forward* à séquestrer sa partition *P*.
3. Alice lance le droit de vente à terme à destination de Bob, sa partition *P* est alors gelée pour la durée *d* et Bob est débité du surcoût *x* de la vente au profit d'Alice sur le *smart contract* de paiement ERC20. Alice est toujours propriétaire de la partition, mais ne peut en disposer.

Résolution de la vente

Après écoulement de la durée prévue, Bob choisit de terminer ou non la vente.

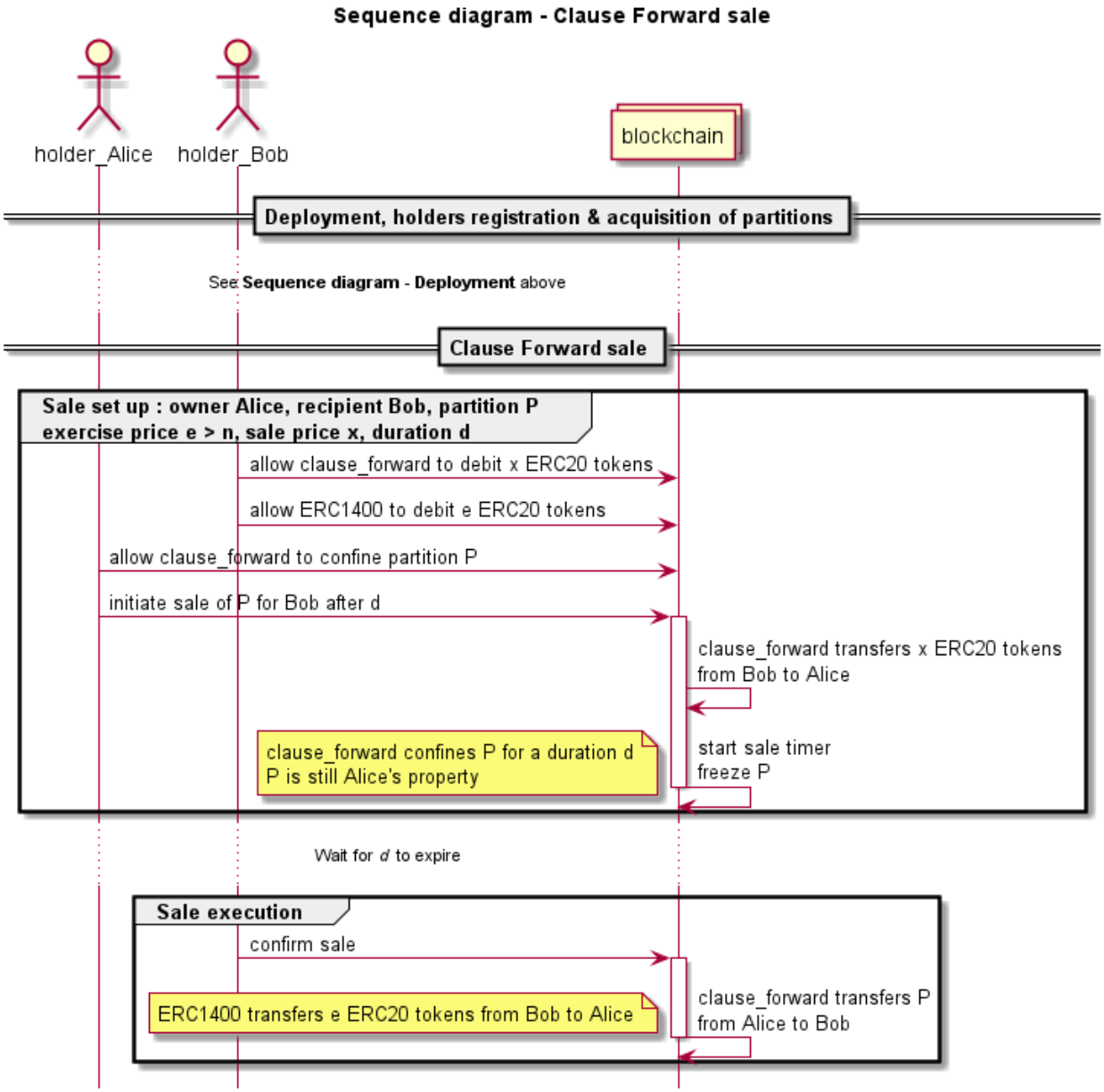
Bob accepte

Si Bob souhaite toujours conclure la vente et acheter la partition *P*, il doit au préalable autoriser le *smart contract* ERC1400 à le débiter de *e tokens* de paiement ERC20. Il indique ensuite au *smart contract clause_forward* sont intention de conclure la vente, la partition est alors transférée de Alice à Bob et dégelée, et Bob est débité de *e tokens* de paiement au profit d'Alice.

Bob refuse

Si Bob ne juge plus intéressant de conclure la vente une fois le terme dépassé, il peut annuler la vente. Il indique donc sa décision au *smart contract clause_forward*. Alice peut alors disposer à nouveau de la partition *P* et Bob ne récupère pas le surcoût *x* versé à Alice.

La séquence des opérations en cas de vente est illustrée par le diagramme suivant :



Au cours de ce scénario sont testées les méthodes permettant l'émission du droit de vente et la vente de l'action elle-même conformément aux différentes valeurs spécifiées.

C. Clause de préemption

Alice est détentrice d'une partition *P* de valeur *n* qu'elle souhaite vendre à Bob ou Charles avec un surcoût de préemption *x* dans un délai *d*.

Émission de l'avis de vente

1. Alice autorise le *smart contract clause_preemption* à séquestrer sa partition *P*.
2. Alice enregistre Bob et Charles en tant que destinataire potentiel des avis de vente qu'elle émettra pendant une durée *d*.
3. Dans la période fixée *d*, Alice crée les avis de vente pour *P* valables pour la durée *d* à destination de Bob et Charles, avec un coût de préemption *x*. Sa partition est alors gelée.

Réponses des destinataires

Bob souhaite acheter la partition *P* mise en vente par Alice, pour cela il doit :

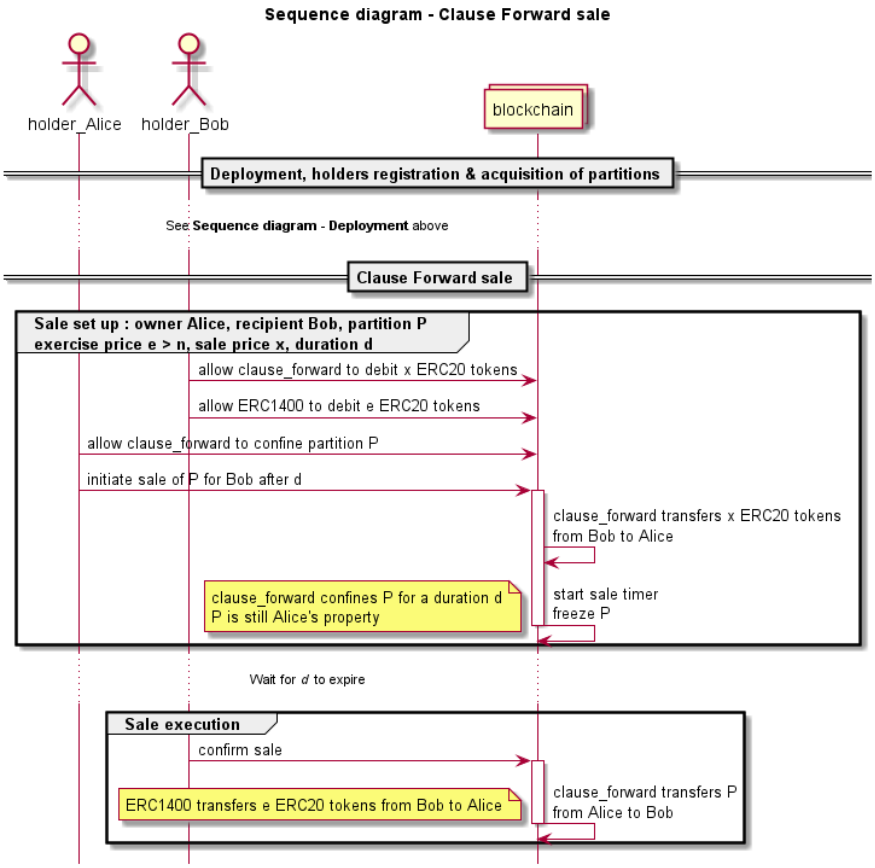
1. Autoriser le *smart contract clause_preemption* à le débiter de *x tokens* de paiement ERC20.
2. Autoriser le *smart contract* de partitions ERC1400 à le débiter de *n tokens* de paiement.
3. Informer le *smart contract clause_preemption* de sa volonté d'acheter *P* dans le temps imparti *d*.

Charles lui n'est pas intéressé par l'avis de vente, il lui suffit donc d'informer le *smart contract clause_preemption* de sa décision ou simplement d'ignorer l'offre.

Décision d'Alice et vente

Lorsqu'au moins un destinataire de l'avis de vente s'est prononcé intéressé par l'achat de la partition *P*, Alice peut choisir parmi eux celui à qui elle souhaite effectivement vendre. Dans notre scénario seul Bob est intéressé par *P* au prix demandé, Alice informe donc le *smart contract clause_preemption* de sa décision de vendre *P* à Bob. La partition *P* est alors transférée d'Alice à Bob contre *n* + *x tokens* de paiement ERC20.

La séquence des opérations est illustrée par le diagramme suivant :



Les tests exécutés aux différentes étapes de ce scénario permettent de s'assurer du bon déroulement de la vente et du positionnement des acteurs par rapport à l'offre d'Alice.

D. Clause « *buy or sell* »

Alice et Bob sont chacun détenteur d'un ensemble de partitions de valeur totale *n*. Alice fait une proposition valable pour une durée *d* portant sur la totalité des partitions pour un surcoût *x* donc au prix d'exercice $n + x = e > n$.

Émission de l'avis d'achat ou vente

1. Alice autorise le *smart contract clause_sell_or_buy* à geler l'ensemble de ses partitions.
2. Alice autorise le *smart contract clause_sell_or_buy* à débiter son compte de *x tokens* de paiement ERC20.
3. Alice autorise le *smart contract* ERC1400 à débiter son compte de *n tokens* de paiement ERC20.
4. Alice lance l'avis de vente ou achat pour l'ensemble des partitions à destination de Bob au prix d'exercice *e* pour la durée *d*.

Résolution de la vente

Bob choisit dans le temps imparti, d'accepter l'offre et donc de vendre ses partitions, ou de refuser l'offre et donc d'acheter les partitions d'Alice. Au-delà du temps imparti, il est contraint de vendre ses partitions à Alice.

Bob accepte et vend

1. Bob autorise le *smart contract clause_sell_or_buy* à geler l'ensemble de ses partitions.
2. Bob informe le *smart contract clause_sell_or_buy* dans le délai *d* de sa volonté d'accepter l'offre d'Alice. Ses partitions sont alors transférées en totalité à Alice qui verse à Bob le montant de *e tokens* de paiement ERC20.

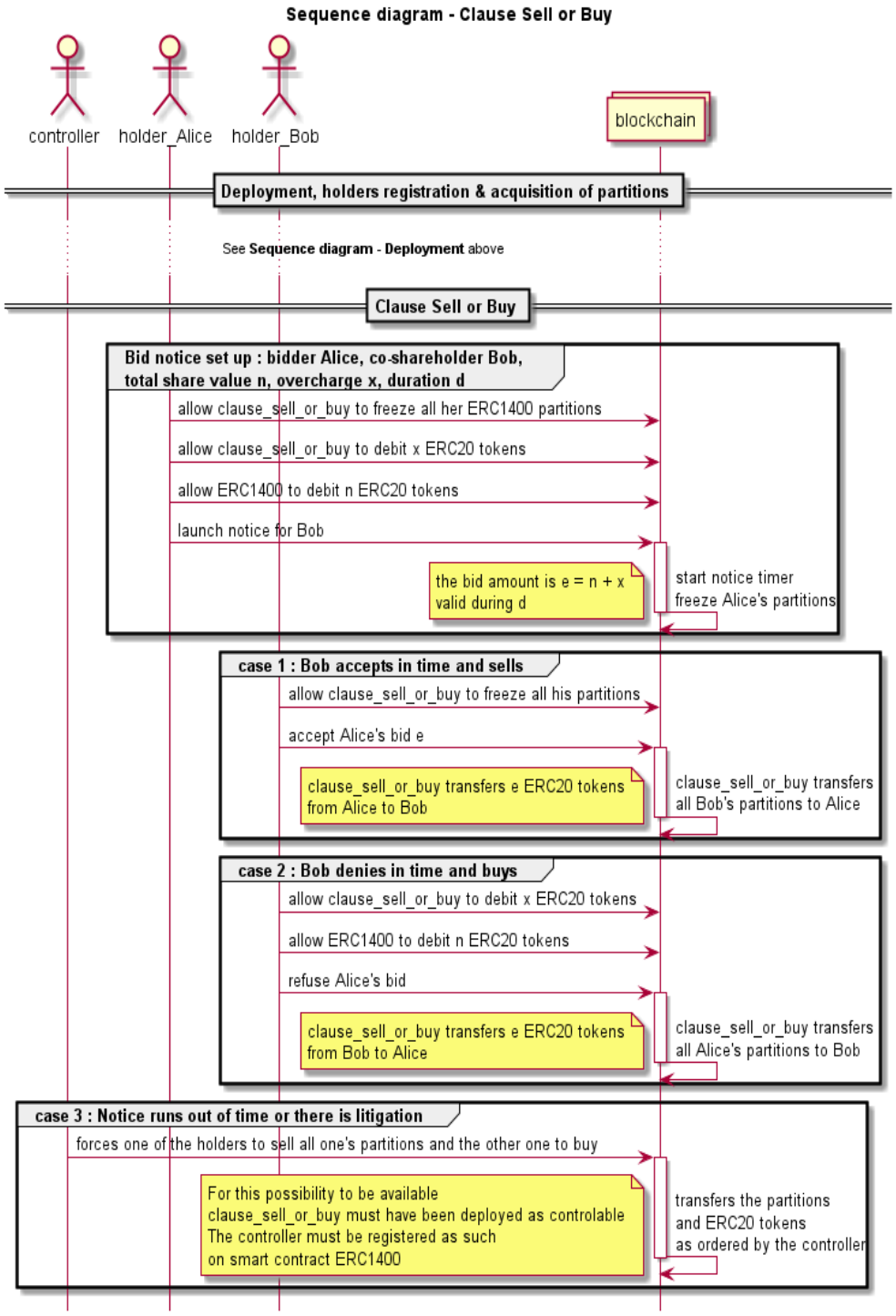
Bob refuse et achète

1. Bob autorise le *smart contract clause_sell_or_buy* à le débiter de *x tokens* de paiement ERC20.
2. Bob autorise le *smart contract* ERC1400 à le débiter de *n tokens* de paiement.
3. Bob informe le *smart contract clause_sell_or_buy* dans le délai *d* de sa volonté de refuser l'offre d'Alice. La totalité des partitions d'Alice est alors transférée à Bob au prix d'exercice proposé *e*.

Le délai est dépassé ou un litige entre les acteurs empêche la conclusion de la vente

Dans ce cas une tierce partie virtuelle ou non, le *controller*, peut forcer la vente des actions de l'un ou l'autre des acteurs sans que ces derniers aient nécessairement donné leur autorisation aux *smart contracts* utilisés (décision judiciaire, etc).

La séquence des opérations est illustrée par le diagramme suivant :



Au cours de ce déroulement, chaque cas de figure est testé afin de s'assurer que les *smart contracts* gèrent les différentes possibilités comme ils le doivent et que chaque étape intermédiaire fonctionne correctement, de l'émission de l'avis d'achat ou vente au transfert de partitions et *tokens* de paiement.

Exécution des tests

Pour lancer l'exécution des tests présentés précédemment, il convient de se placer dans l'environnement ***brownie*** du projet depuis un terminal de commandes Python et de lancer la commande suivante pour tester chacune des clauses :

>>> brownie test -s

Ou pour tester une seule clause X en particulier :

>>> brownie test .\tests\test_clause_X.py -s

Le module ***brownie*** se charge alors de lancer l'exécution des tests dans l'ordre, cette étape peut être identifiée par l'affichage des lignes suivantes :

```
Brownie v1.12.2 - Python development framework for Ethereum

============================================= test session starts =============================================
platform win32 -- Python 3.6.12, pytest-6.0.1, py-1.10.0, pluggy-0.13.1 -- d:\documents_officiels\mission_uga\clausier\venv-clausier\python.exe
cachedir: .pytest_cache
hypothesis profile 'brownie-verbose' -> verbosity=2, deadline=None, max_examples=50, stateful_step_count=10, report_multiple_bugs=False, database=
DirectoryBasedExampleDatabase(WindowsPath('C:/Users/Thomas/.brownie/hypothesis'))
rootdir: D:\Documents_officiels\mission_uga\clausier\open-clausier
plugins: eth-brownie-1.12.2, hypothesis-5.41.3, forked-1.3.0, xdist-1.34.0, web3-5.11.1
collected 41 items

Launching 'ganache-cli.cmd --port 8545 --gasLimit 12000000 --accounts 10 --hardfork istanbul --mnemonic brownie'...
```

Puis lorsque l'exécution est terminée, *brownie* fournit un rapport du déroulement des tests qui est clos par la ligne de synthèse suivante :

```
================================== 1 failed, 40 passed in 220.84s (0:03:40) ==================================
Terminating local RPC client...
```

Perspectives

Cette batterie de tests fonctionnels est un outil efficace pour s'assurer que les scénarios typiques d'utilisation des différentes clauses de la librairie se déroulent correctement, elle gagnerait à être enrichie par des tests de sécurité. L'ajout aux différentes étapes de ces scénarios de tests vérifiant le bon fonctionnement des *requirements* (prérequis à l'exécution d'une fonction d'un *smart contract*) permettrait d'assurer la robustesse des différentes méthodes des *smart contracts* d'un point de vue fonctionnel.

Section 3. Création de la librairie

La plateforme qui accueille les *smart contracts* doit répondre à une double exigence qualitative : satisfaire les deux communautés d'informaticiens et de juristes. Telle était l'ambition du groupe de recherche (§1), conduisant à la réalisation d'une interface accessible depuis un site internet lié à un GitHub, c'est-à-dire un espace de programmation collaboratif pour les informaticiens (§2).

Paragraphe 1. Ambitions

Trois principales ambitions nous ont guidés.

- La première vient d'un constat : la nécessité pour les professionnels du droit et de la justice de saisir le tournant numérique qui se présente. L'outil technologique, qu'il s'agisse de la *blockchain* ou de l'intelligence artificielle, est souvent vecteur d'inquiétudes. C'est principalement la désintermédiation que permet la *blockchain* qui inquiète les professionnels du droit, jusqu'à craindre la disparition de leur profession. Sur ce point, Alexis Downe a parfaitement montré que le risque devait être pris en considération, mais qu'il doit être relativisé. Nous soulignons donc avec conviction que l'outil technologique, qu'il s'agisse d'une application de *blockchain*, d'un *smart contract* ou bien d'un logiciel d'intelligence artificielle, doit rester un complément des prestations intellectuelles humaines. Pour autant, les nouvelles technologies poussent au renouvellement des pratiques. Et en cela, elles doivent être accueillies. Comme l'a souligné Me Roguet, en sa qualité de bâtonnier de l'Ordre des avocats à Grenoble au moment où le projet a démarré, les avocats doivent envisager l'avenir de leur profession de manière « proactive » et nous espérons que la librairie de *smart contracts* leur fournira de nouveaux outils pour saisir ce tournant technologique. Rappelons que les *smart contracts* développés par l'équipe de recherche sont accessibles à tous et gratuitement[34]. Cette logique d'ouverture était essentielle pour les membres.

- La deuxième ambition de cette recherche était de produire un outil « utile » aux professionnels du droit et de la justice. En effet, il n'est pas question sous couvert de la « mode »

[34] Toutes les informations concernant l'accessibilité des clauses sur trouve sur le site internet du projet : https://smart-contracts.univ-grenoble-alpes.fr/

qui environne le secteur de la *blockchain* de voir des *smart contracts* déployés dans tous les domaines et dans tous les secteurs d'activité et pour l'intégralité des clauses d'un contrat. Me Guilhaudis l'a souligné dans l'entretien donné et tout au long de la recherche. Il a donc été indispensable pour l'équipe de rechercher :

1. si une *blockchain* était pertinente, Mme Hennebert a d'ailleurs rappelé dans sa présentation les principales caractéristiques de la *blockchain* ;
2. de vérifier auprès des professionnels membres de l'équipe s'il y avait un intérêt pratique à la traduction de ces clauses. Le *smart contract* est pertinent lorsqu'il s'agit de faire circuler des actifs « tokenisés ». Sur la notion de tokenisation, on peut renvoyer à l'entretien avec Rémy Ozcan et à la présentation de Mme Hennebert. Ceci explique que nous avons fait une sélection portant principalement sur des clauses issues des contrats d'affaires et financiers. Dans les perspectives de la recherche, d'autres clauses pourront être envisagées.

- La troisième ambition, et non la moindre, visait à proposer une traduction de clauses en *smart contracts* qui ne menacerait pas la sécurité juridique des parties, mais qui permettrait au contraire de renforcer les effets du contrat. Il ne faut pas perdre de vue que le *smart contract* n'est qu'un outillage au service de la force obligatoire du contrat. Les risques en matière de données et d'algorithmes rappelés par Abdoulaye Diallo et Sihem Amer-Yahia et les risques pour le professionnel du droit soulignés par Me Guilhaudis imposent d'avoir une approche raisonnable du *smart contract*. La présente réalisation impliquera que les professionnels avocats, juristes, conseils se saisissent des *smart contracts* finalisés par notre équipe et travaillent en collaboration avec des informaticien·nes afin d'adapter notre proposition générique à chaque situation particulière. Jusqu'à ce que les juristes puissent devenir un jour codeurs (est-ce souhaitable ? on peut en douter…), le potentiel des outils technologiques ne se développera que dans une complémentarité indispensable des disciplines du droit et de l'informatique.

Paragraphe 2. Réalisation

Nous voulions une plateforme qui puisse s'adresser à la fois à des juristes, qui ne sont pas forcément technophiles mais peuvent tout de même s'intéresser aux *smart contracts* sans posséder une culture du code informatique, et à des informaticiens ou des juristes très technophiles capables de coder des *smart contracts*. C'est donc un problème auquel nous avons

dû faire face puisque ces deux catégories de personnes n'utilisent pas les mêmes outils, n'ont pas la même culture, n'ont pas le même vocabulaire ni le même savoir-faire, etc. Le défi était donc de faire une seule et même plateforme pour ces deux populations.

Ainsi, l'outil que nous avons utilisé pour notre projet est une plateforme de partage de code : GitHub, qui permet de voir les codes des applications qui s'y trouvent et les modifications apportées à une application. Les parties en vert concernent ce qui a été rajouté et les parties en rouge ce qui a été supprimé. C'est quelque chose dont nous avions absolument besoin pour déployer des *smart contracts*, puisque cela permet de savoir « qui a fait quoi ». Mais ces fonctionnalités sont très difficiles à développer et assez inaccessibles pour quelqu'un de non avisé. Si cette plateforme semble adaptée dans l'ensemble, une seule chose ne l'est pas : elle est incompréhensible pour le profane. Sur GitHub, tout le monde peut contribuer, même des personnes que nous ne connaissons pas forcément.

C'est sur cette plateforme que nos clauses rédigées sont accessibles. En cliquant dessus, on trouvera d'abord une description de la clause et adossé à cette description le *smart contract* en lui-même et donc le code du *smart contract* avec le langage d'écriture et tout ce qui va lui permettre de fonctionner. Mais ici, seul le « juriste-geek » capable de créer son propre *smart contract* peut s'y retrouver. Il s'agissait donc d'envisager comment mettre le *smart contract* à disposition de toute la communauté des juristes.

Pour cela, nous avons fait le choix d'utiliser un site dédié au projet : « Opensmartcontract ». Dans le cas d'une personne souhaitant créer son propre *smart contract*, celle-ci pourra via le back-office de la plateforme mise en place avoir accès à la partie configuration et importer directement les clauses et toutes les informations provenant de GitHub. Ainsi, les clauses élaborées sont disponibles sur le site Opensmartcontract[35], ce qui signifie que pour chacune un développeur a créé le *smart contract*, l'a décrit et importé sur la plateforme.

Dans la situation d'une personne lambda consultant la page de présentation du site Opensmartcontract, elle aura accès à une description du projet. Il existe deux modes d'utilisation du site Internet, un mode consultation, qui permet de consulter les clauses présentes sur le site et de comprendre comment les utiliser, et un mode contribution, qui permet d'ajouter

[35] Le lien vers la plateforme dédiée à la Librairie de *smart contracts* est : https://opensmartcontract-front.netlify.app/

des *smart contracts* comme nous l'avons fait. Toute personne a donc la possibilité de découvrir les clauses présentes dans le clausier et de regarder les informations relatives aux différentes clauses. Chacun peut faire remonter des informations, ajouter des commentaires à une clause suite à sa découverte du clausier, comme par exemple des corrections ou des précisions sur les textes de loi s'agissant des juristes.

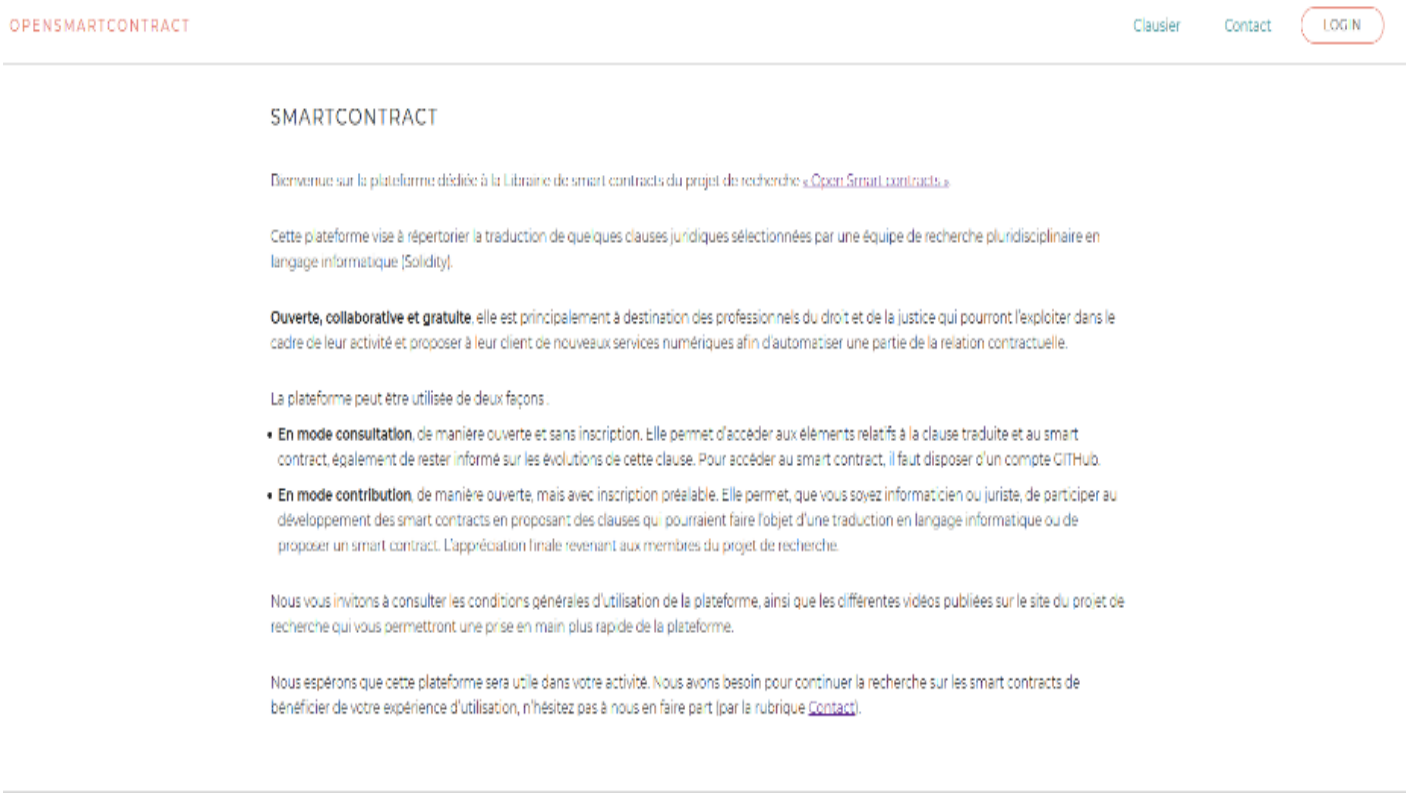
À partir de ces commentaires et retours, un système d'alerte géré à partir de GitHub est alors déclenché. La plateforme GitHub nous fait remonter directement les problèmes à régler en envoyant un message personnel au membre de l'équipe ayant rédigé le *smart contract*. Le développeur peut donc vérifier la clause qu'il a rédigée et la corriger si nécessaire. Le développeur peut également préciser, dans la page de rédaction du code, les raisons pour lesquelles il a apporté telle ou telle modification. Nous sommes ainsi en mesure de savoir qui a modifié quoi et quand. C'est indispensable si l'on veut que des milliers de personnes puissent collaborer pour créer des centaines de *smart contracts*. La plateforme Opensmartcontract quant à elle sera beaucoup plus lisible pour un juriste non habitué au code informatique.

L'administrateur, qui après avoir longuement travaillé sur une clause a le sentiment d'être arrivé à une forme de certitude, de confiance, sur la solidité de cette clause, pourra valider son *smart contract*. Le statut de ladite clause passera donc de "en cours de validation" à "certifié". L'avantage pour les visiteurs du site est qu'ils pourront savoir qu'une clause "en cours de validation" est une contribution externe d'un usager de notre site, qui est utilisable sans toutefois être complétement sûre, tandis qu'une clause "certifiée" est une clause vérifiée par les administrateurs du site Opensmartcontract, ce qui constitue une forme de garantie et de confiance.

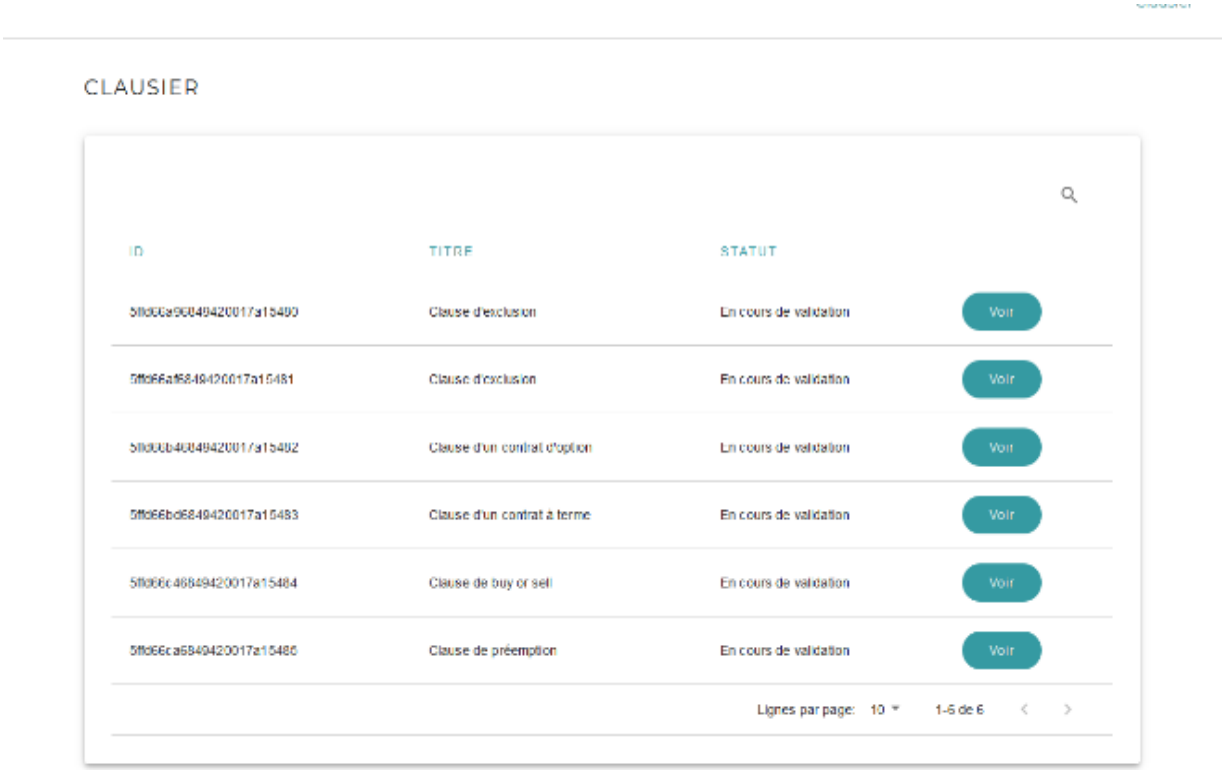
On voit comment on fait le lien avec cette plateforme entre les différentes communautés, d'un côté, la communauté des personnes qui souhaitent développer des *smart contracts* mais ne savent pas comment les mettre à la disposition de tous et, de l'autre côté, les personnes qui souhaitent utiliser des *smart contracts* mis à leur disposition mais ne savent ni les coder ni évaluer leur solidité.

Concernant la traduction en anglais du code, la localisation, comme on l'appelle en informatique, permet l'utilisation simultanée de plusieurs langues. C'est donc un aspect que notre site peut gérer sans souci. Sous chaque code, les "commentaires" ne sont pas interprétés

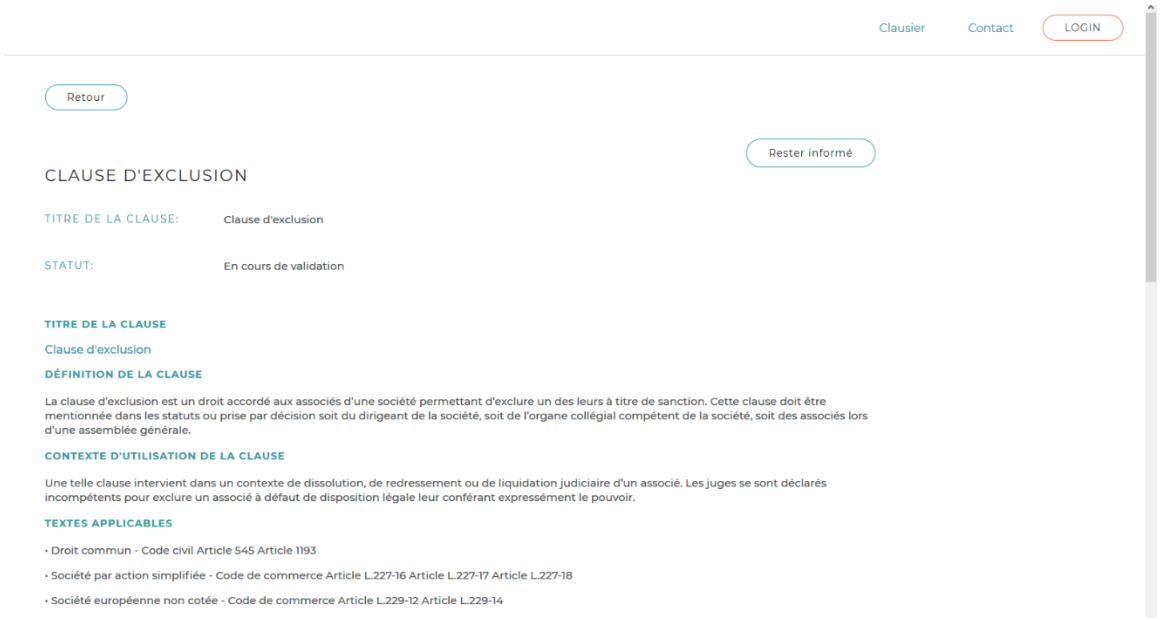
par la machine et peuvent donc s'écrire en français ou en anglais. La volonté de s'ouvrir à l'international pourrait inciter à tout uniformiser en anglais. Dans la réalité des faits, très peu de codes, mis à part ceux réalisés par les anglophones, sont uniformes au plan linguistique, il y a souvent un mélange. On remarque d'ailleurs qu'en dehors des commentaires, le code lui-même contient des termes à la fois anglais et français. Cela ne pose pas de problème en soi, il faut juste que les personnes qui développent ensemble un *smart contract* soient capables de le comprendre et de se mettre d'accord. Il est certain que si des anglophones venant sur le site voient les termes acheteur et vendeur et ne sont pas sûrs de ce qu'ils signifient, cela soulève une difficulté. Mais cela fait partie de la vie normale du logiciel. Le champ linguistique du code s'élargit au fur et à mesure des contributions, un Anglais proposant une traduction, un Chinois une autre, et ainsi de suite. Tout cela est géré directement sur GitHub, là où il nous eût été très difficile de mettre en place ce type de fonctionnalité directement sur notre site.



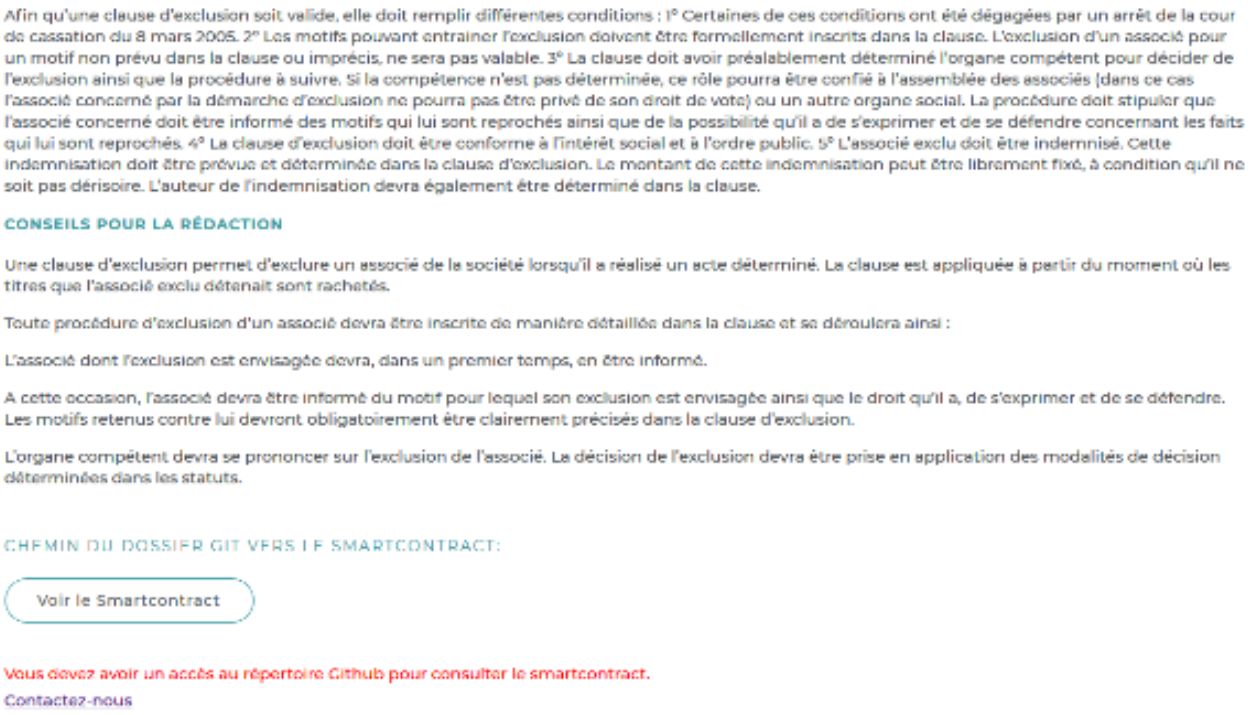
Capture d'écran 7 : Page d'accueil de la plateforme OPENSMARTCONTRACT



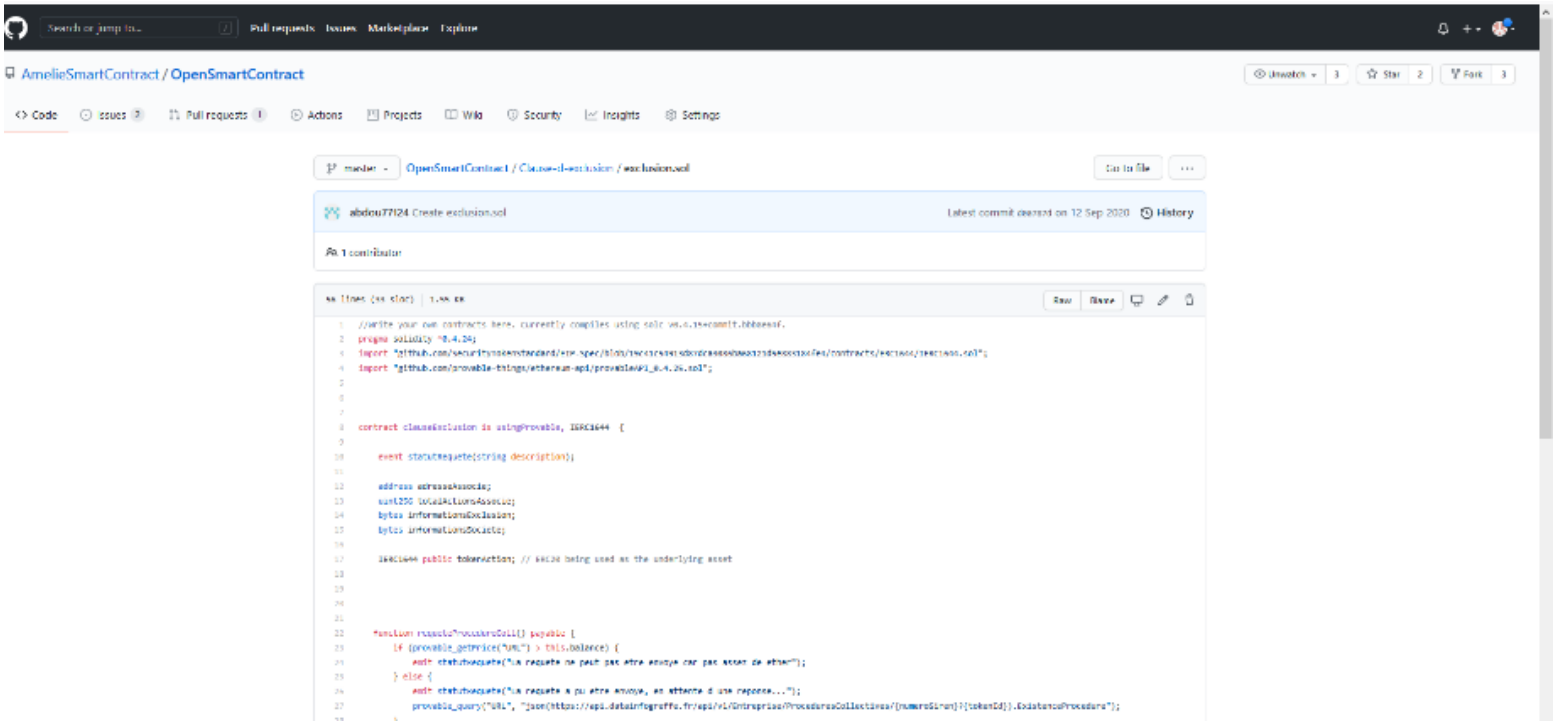
Capture d'écran 8 : Le clausier, avec les différentes clauses



Capture d'écran 9 : Présentation de la Clause d'Exclusion



Capture d'écran 10 : Précision pour l'application de la Clause d'Exclusion



Capture d'écran 11 : Visualisation du smart contract pour la clause d'exclusion

Section 4. Encadrement juridique de l'utilisation de la librairie

La mise en ligne de cette librairie a impliqué la rédaction de conditions générales d'utilisation (§1) et d'une réflexion sur la conformité au Règlement général sur la protection des données (RGDP) (§2) ainsi que la mise en place d'une licence libre pour l'utilisation des *smart contracts* (§3). L'ensemble de ce cadre juridique était sous-tendu par une réflexion sur la responsabilité liée à cette programmation.

Paragraphe 1 : Conditions juridiques d'utilisation de la librairie

Préalables à la rédaction des conditions générales d'utilisation (CGU). La plateforme propose, à des fins de recherche scientifique, des clauses juridiques et leur traduction en langage informatique (Solidity 6.0) dans des *smart contracts*. Cette traduction informatique peut permettre à l'utilisateur, sous certaines conditions, d'automatiser l'exécution d'une clause juridique sur une *blockchain*. Ouverte, collaborative et gratuite, la plateforme Opensmartcontract est à destination des professionnels du droit et de la justice (juristes et avocats).

Description des services. La plateforme Opensmartcontract peut être utilisée de deux façons :

- **en mode consultation**, de manière ouverte et sans inscription. Il permet d'accéder au clausier comportant des clauses juridiques et à des traductions génériques proposées en langage informatique dans des *smart contracts*. La plateforme permet aussi d'être informé sur les évolutions de cette clause.
- **en mode contribution**, de manière ouverte, mais avec inscription préalable. Il permet de participer à l'enrichissement du clausier en proposant des clauses juridiques qui pourraient faire l'objet d'une traduction en langage informatique. Il est également possible de proposer des *smart contracts*.

Création d'un compte sur Opensmartcontract – Clôture. Pour accéder aux services de la plateforme en tant que contributeur, l'utilisateur doit créer un compte en renseignant son nom, son adresse électronique et un mot de passe.
L'utilisateur dispose d'un compte nominatif.

Chaque utilisateur ne peut créer qu'un seul compte. Pour cela, il garantit que ses informations personnelles sont exactes, sincères et à jour. Tous les champs présents dans le formulaire de création de compte doivent être renseignés. Tout défaut de réponse aura pour conséquence la non-création du compte.
Avant d'accéder pour la première fois aux services de la plateforme en tant que contributeur, l'utilisateur devra dans tous les cas, accepter, en cochant les cases prévues à cet effet :

- les conditions générales d'utilisation
- la politique d'utilisation des données personnelles.

Les identifiants de connexion de l'utilisateur sont strictement personnels et confidentiels. L'utilisateur s'engage à les conserver secrets et à ne pas les divulguer sous quelque forme que ce soit et il lui est interdit de les céder à des tiers.

L'utilisateur conserve la possibilité de mettre à jour ses données personnelles à tout moment depuis son compte.
L'utilisateur s'engage à informer le fournisseur par tout moyen et dans les meilleurs délais, en cas de :

- vol ou perte d'identifiant ou mot de passe suite à un piratage ;
- perte du tout ;
- piratage d'un ou plusieurs terminaux permettant l'accès au compte de l'utilisateur.

L'utilisateur a la possibilité de demander la suppression de son compte.
Le fournisseur se réserve par ailleurs, la possibilité, après l'envoi d'une notification d'inactivité par courrier électronique restée sans réponse de l'utilisateur pendant 30 jours, de supprimer un compte, si le compte présente une durée d'inactivité égale ou supérieure à un an.

En cas de clôture du compte, les données de l'utilisateur sont supprimées ou anonymisées à des fins statistiques.

Création et modification des *smart contracts* depuis GitHub. GitHub est une plateforme web de partage open sources de codes, appartenant à Microsoft, très utilisée par les développeurs informatiques. Le fournisseur a choisi d'utiliser GitHub pour gérer les tickets des contributeurs, ainsi que les modifications et évolutions des *smart contracts* associés aux clauses juridiques disponibles sur la plateforme. Pour accéder à l'espace « OpenSmartContract » de la plateforme GitHub, le contributeur doit au préalable créer un compte sur GitHub. Les utilisateurs de la

plateforme GitHub peuvent librement proposer de nouveaux *smart contracts*, ainsi que des modifications et/ou des évolutions des *smart contracts* existants. Toutefois, le fournisseur décide seul de leur publication sur la plateforme Opensmartcontract.

Services délivrés par le fournisseur à l'utilisateur. Dans le cadre de la plateforme Opensmartcontract, le fournisseur rend à l'utilisateur les services suivants.

- Le droit à l'accès et à l'utilisation des services de la plateforme (droit de licence) consenti à l'utilisateur dans les conditions prévues à l'article 9 « Propriété intellectuelle » des présentes CGU

- L'hébergement de la plateforme et des données personnelles qu'elle contient

L'utilisateur est informé que le fournisseur sous-traite l'hébergement de la plateforme à un prestataire tiers. L'hébergement est réalisé sur des serveurs mutualisés situés dans l'Union européenne.

- Maintenance

L'utilisateur est informé que le fournisseur sous-traite la maintenance de la plateforme à un prestataire tiers. Le fournisseur ne garantit pas l'absence totale de dysfonctionnement.

- La disponibilité des services

Le fournisseur mettra en œuvre tous les efforts matériellement raisonnables pour assurer la disponibilité des services de la plateforme, sauf pendant les périodes de maintenance.
Le fournisseur pourra interrompre les services occasionnellement pour des travaux de maintenance et/ou d'amélioration.

- La sécurité

Le fournisseur propose les services de la plateforme, dans des conditions de sécurité conformes à la réglementation française en vigueur.

Services non fournis à l'utilisateur et points de vigilance particuliers. Le fournisseur entend attirer l'attention de l'utilisateur sur les points suivants :

- la plateforme s'adresse à des professionnels du droit. Par conséquent, l'utilisateur est censé parfaitement connaître le domaine juridique pour lequel il envisage l'application pratique d'une clause juridique relevant du clausier et sa traduction informatique en *smart contract* ;
- aucun conseil, aucune validation d'aucune sorte, n'est délivré par le fournisseur à l'utilisateur, que ce soit au plan juridique ou informatique ;
- l'utilisateur juge seul de l'opportunité d'utiliser une clause traduite en *smart contract* proposée sur la plateforme pour son compte ou pour le compte d'un client. Par ailleurs, il décide seul de la manière d'utiliser ces éléments ;
- l'utilisateur assume seul et en totalité les éventuelles conséquences préjudiciables de ses choix, sans aucune possibilité de recours contre le fournisseur.

Avant toute application pratique d'un *smart contract*, l'utilisateur doit systématiquement veiller à :

- adapter la clause et le *smart contract* proposés sur la plateforme à son besoin spécifique, le fournisseur ne proposant sur la plateforme que des versions génériques ;
- s'assurer que la traduction de la clause juridique en langage informatique est exacte, complète, et adaptée à ses besoins ou à ceux de son client ;
- tester le bon fonctionnement du *smart contract* ;
- se rapprocher d'un informaticien pour la mise en œuvre du *smart contract* sur une *blockchain*, étant précisé que l'utilisateur décide seul du type de *blockchain* qu'il entend utiliser.

Obligations générales de l'utilisateur. L'utilisateur s'engage à utiliser la plateforme dans les conditions suivantes :

- ne pas l'utiliser aux fins d'entraver ou altérer son fonctionnement, notamment en l'encombrant, volontairement ou involontairement, par le transfert intempestif de contenus, en dehors des cas d'utilisation prévus ;
- ne pas extraire, copier, dupliquer, des éléments et graphismes, sur lesquels seul le fournisseur dispose des droits de propriété intellectuelle ;
- ne pas introduire de fichiers/programmes malveillants, ou contenant des virus informatiques ;

- ne pas stocker, transmettre du contenu non autorisé, qui serait illégal ou qui pourrait être constitutif d'incitation à la réalisation de crimes et délits, de diffamations et injures, d'atteinte à la vie privée, ou encore d'actes mettant en péril des mineurs ;
- ne pas stocker, transmettre du contenu qui violerait le droit à l'image, tout droit de propriété intellectuelle ou tout autre droit appartenant à autrui.

L'utilisateur s'engage à ne pas utiliser la plateforme notamment pour :

- effectuer de la publicité non autorisée ;
- transmettre ou envoyer tout contenu qui serait illégal ou qui pourrait être constitutif d'incitation à la réalisation de crimes et délits, de diffamations et injures, d'atteinte à la vie privée, ou encore d'actes mettant en péril des mineurs ;
- transmettre tout contenu qui violerait le droit au respect de la vie privée, le droit à l'image, tout droit de propriété intellectuelle notamment tout brevet, marque déposée, droit d'auteur, secret de fabrication ou tout autre droit appartenant à autrui ;
- transmettre tout contenu contenant des virus informatiques ou tout autre code, dossier ou programme conçus pour interrompre, détruire ou limiter la fonctionnalité de tout logiciel, ordinateur, ou outil de télécommunication sans que cette énumération ne soit limitative ;
- violer, intentionnellement ou non, toute loi ou réglementation nationale ou internationale.

L'utilisateur déclare disposer des droits nécessaires au transfert, à la communication et à la diffusion des contenus via la plateforme, et garantit le fournisseur de toute action contentieuse qui pourrait être dirigée contre lui à l'occasion de ces activités (en ce compris les honoraires d'avocat et tous autres frais de procédure).

Propriété intellectuelle. Concernant les droits de propriété intellectuelle relatifs aux éléments composant la plateforme, le fournisseur détient les droits de propriété intellectuelle lui permettant de concéder à l'utilisateur le droit d'accéder à la plateforme, conformément aux présentes.

L'utilisateur reconnaît que les présentes ne lui confèrent aucun droit de propriété sur les éléments composant la plateforme (tant au plan logiciel que graphique). La mise à disposition de ces éléments dans les conditions prévues aux présentes ne saurait être analysée comme la cession d'un quelconque droit de propriété intellectuelle, au sens du Code français de la propriété intellectuelle.

Ainsi, le droit d'accès et d'utilisation qui est concédé à l'utilisateur au titre des présentes est exclusif de tout transfert de droits de propriété intellectuelle. Ceci concerne en particulier les droits visés aux articles L. 122-1, L. 122-6, L. 122-7 du Code de la propriété intellectuelle.

L'utilisateur est informé que certains éléments composant la plateforme sont susceptibles d'utiliser ou de reproduire des modules ou bibliothèques dites « libres » ou « open source ».
Les licences relatives à ces modules ou bibliothèques (tenues à la disposition de l'utilisateur sur demande écrite) peuvent contenir des exclusions pures et simples de toutes garanties. Dans ce cas, l'utilisateur accepte que le fournisseur ne puisse lui conférer plus de garanties qu'il n'en tient lui-même des licences de ces modules ou bibliothèques.

Concernant, les droits de propriété intellectuelle relatifs au clausier et aux *smart contracts*, le fournisseur accorde à tous les utilisateurs un droit de licence perpétuel, mondial, non exclusif, transférable, sous-licenciable dans les mêmes conditions, pour utiliser, modifier, repartager, copier, représenter publiquement ses créations à des fins personnelles ou commerciales. Lorsque l'utilisateur décide de publier une création qu'il a faite, seul ou avec d'autres, sur la plateforme Opensmartcontract, il accorde alors à tous les utilisateurs un droit de licence perpétuel, mondial, non exclusif, transférable, sous-licenciable dans les mêmes conditions, pour utiliser, modifier, repartager, copier, représenter publiquement cette création à des fins personnelles ou commerciales.

Ces règles sont également applicables à toutes les modifications, contributions, mises à jour ou encore toutes améliorations d'œuvres existantes relatives au clausier et aux *smart contracts*.

Le contenu que l'utilisateur choisit de publier, partager ou importer sur la plateforme Opensmartcontract ou GitHub peut être protégé par des droits de propriété intellectuelle de tiers.
L'utilisateur s'engage à respecter scrupuleusement les droits de propriété intellectuelle des tiers et à s'assurer au préalable qu'il dispose de tous les droits nécessaires.
L'utilisateur s'engage à garantir le fournisseur contre toute éventuelle action et condamnation en responsabilité et/ou en contrefaçon. Dans un tel cas, l'utilisateur devra rembourser tous les dommages et intérêts, amendes, frais et dépenses, que le fournisseur pourrait être amené à verser à un tiers à ce titre.

Exclusion de responsabilité. Le fournisseur exclut toute responsabilité pour les services qu'il propose aux utilisateurs depuis la plateforme Opensmartcontract ainsi que depuis le site GitHub.

Paragraphe 2 : Le Règlement général sur la protection des données personnelles

Données que le fournisseur collecte en tant que responsable de traitement. Lors de la création du compte pour devenir contributeur, les données à renseigner sont : nom, prénom, courriel, profession, société, adresse postale. Lorsqu'un utilisateur ouvre un ticket, sont renseignés : nom, prénom, courriel, contenu du ticket. Lorsqu'un utilisateur souhaite être tenu informé des actualités relatives à une clause figurant dans le clausier, il renseigne : nom, prénom, courriel. Lorsqu'un utilisateur utilise la page de contact, il renseigne : nom, prénom, courriel, profession, objet de la demande, contenu de la demande.

Collecte. Les données sont collectées directement par le fournisseur. La finalité des traitements est la recherche scientifique. Les données personnelles faisant l'objet d'un traitement sont conservées par le fournisseur pendant une durée n'excédant pas celle nécessaire aux finalités pour lesquelles elles sont enregistrées. Le fournisseur collecte et traite les données personnelles uniquement lorsque ces traitements reposent sur une base légale. Cette base légale est l'intérêt légitime.

Utilisation. Sauf lorsque cela est nécessaire à la réalisation de nos services, le fournisseur ne partage pas les données avec des tiers.
L'utilisateur est cependant informé que pour la réalisation de certains services, le fournisseur fait appel à des prestataires externes (notamment pour l'hébergement et la maintenance de la plateforme). Ces prestataires peuvent avoir accès à certaines données dans la stricte limite de ce qui est nécessaire et uniquement afin d'assurer la mission qui leur est confiée par le fournisseur.

Le fournisseur met en œuvre les mesures techniques et organisationnelles appropriées afin de garantir un niveau de sécurité adapté au risque.

Le fournisseur prend des mesures afin de garantir que toute personne physique agissant sous leur autorité ou sous celle du sous-traitant, qui a accès à des données à caractère personnel, ne les traite pas, excepté sur instruction du fournisseur, à moins d'y être obligée.

Le fournisseur ne réalise aucun transfert de données personnelles en dehors de l'Union européenne.

Les traitements ne prévoient pas de prise de décision entièrement automatisée.

La personne concernée par un traitement peut définir des directives relatives à la conservation, à l'effacement et à la communication de ses données personnelles après son décès. Ces directives peuvent être générales ou particulières.

La personne concernée par un traitement bénéficie également d'un droit d'accès, d'opposition, de rectification, de suppression et, à certaines conditions, de portabilité de ses données personnelles. La personne concernée a le droit de retirer son consentement à tout moment si le consentement constitue la base légale du traitement.

La demande devra indiquer les nom et prénom, adresse courriel ou postale, de la personne concernée, et être signée et accompagnée d'un justificatif d'identité en cours de validité.

Paragraphe 3. La licence CECILL et responsabilité des *smart contracts*

Licence CeCILL. La participation de Christine HENNEBERT au projet nous a permis d'avoir connaissance de la Licence CeCILL[36]. Ce contrat est une licence de logiciel libre issue d'une concertation entre ses auteurs (CEA, CNRS, INRIA) afin que le respect de deux grands principes préside à sa rédaction :
* d'une part, le respect des principes de diffusion des logiciels libres : accès au code source, droits étendus conférés aux utilisateurs ;
* d'autre part, la désignation d'un droit applicable, le droit français, auquel elle est conforme, tant au regard du droit de la responsabilité civile que du droit de la propriété intellectuelle et de la protection qu'il offre aux auteurs et titulaires des droits patrimoniaux sur un logiciel.

[36] Détails sur cette licence dans l'Annexe 3. Contrat de licence de logiciel libre CeCILL-B, voir p. 205.

En termes de responsabilité, il existe une même exonération : le logiciel est livré « *as is* ».

Réflexions sur la responsabilité de programmation des librairies de *smart contracts*. La responsabilité est une question vaste qui se réfère aux personnes physiques auxquelles est imputée la responsabilité de comportement algorithmique, ainsi que de la révélation de données et d'historiques de transactions.

Plusieurs acteurs interviennent dans le développement et le déploiement d'un *smart contract* :

- **l'organisation** mettant à disposition la *blockchain*, le langage de programmation, le compilateur, le mécanisme de consensus et généralement toute l'infrastructure originelle nécessaire au déploiement du *smart contract* ;
- **les mineurs** chargés de valider les transactions et donc d'exécuter les instructions du *smart contract* sur leurs ordinateurs ;
- **des personnes tierces** qui auraient développé des libraires et des logiciels annexes nécessaires aux développements et déploiement du *smart contract* désiré par le client ;

(tous ces acteurs peuvent changer d'un contrat à l'autre)

- **un prestataire** à qui un client aurait commandé la réalisation d'un *smart contract* et qui serait responsable d'utiliser l'infrastructure fournie par l'organisation pour développer et déployer le *smart contract* selon les spécifications du client ;
- **l'utilisateur du *smart contract***, qui peut aussi être client ou non (il peut avoir développé lui-même le *smart contract*).

Les licences jouent un rôle prépondérant dans la responsabilité. La faisabilité d'un *smart contract* est dépendante des licences sous lesquelles les librairies existantes sont mises à disposition des développeurs. Les licences libres et open source sont deux types de licences sous lesquelles on peut ranger un nombre pléthorique de licences de logiciels. La plupart des langages de programmation sont distribués sous ces types de licence pour faciliter leur adoption, les langages de développement des *smart contracts* n'y font pas exception. D'emblée, il convient de dire que tous les contrats de licence de ces programmes sont très permissifs et permettent une utilisation satisfaisante pour la plupart des cas d'usage. La différence se joue surtout au niveau de la responsabilité des éditeurs du langage.

Conclusion.

Pour clôturer ce rapport, il convient de revenir rapidement sur le déroulement de la recherche (Paragraphe 1) avant d'effectuer une présentation détaillée de la plateforme accueillant le clausier (Paragraphe 2). Pour conclure, nous évoquerons les potentialités du développement des *smart contracts* dans le domaine social (Paragraphe 3).

Paragraphe 1 – Synthèse du déroulement de la recherche

D'importants travaux de recherches préparatoires. L'équipe a procédé à un important travail de réflexion et de recherche avant de se lancer dans la réalisation de *smart contracts*. En effet, il était important de comprendre et d'appréhender les nombreux défis techniques et juridiques qui s'imposaient au projet. Il fallait faire des choix : le choix du langage informatique pour la traduction des clauses ; le choix des clauses elles-mêmes et de leur nombre. Et pour faire ces choix, il était important de définir les enjeux du projet, ses objectifs ainsi que la vision directrice qui guiderait l'équipe. Ont été privilégiées la sécurité, la validité juridique ainsi que l'utilité. Ce sont ces mots d'ordre qui ont orienté les choix de l'équipe. Ainsi, après avoir eu recours à des tableaux comparatifs et des analyses poussées des diverses possibilités, le groupe de recherche a fait ses choix. Le projet se borne donc à la traduction de clauses juridiques en langage informatique Solidity. Plusieurs clauses ont été désignées comme susceptibles d'une traduction ultérieure. Seulement cinq ont été traduites dans le cadre du projet. Toutes se rapportent au droit des affaires. C'est avec cette base solide que l'équipe a entamé la construction du clausier.

La constitution du clausier. Une fois les clauses sélectionnées, l'équipe a pris soin de définir chacune d'entre elles et d'en dresser le régime juridique. Le but était que l'utilisateur arrivant sur la plateforme puisse trouver une explication à propos de chaque clause et de son application. Ce travail a été réalisé en français et en anglais pour s'inscrire dans une démarche internationale. L'équipe a ensuite ajouté une écriture en pseudo-code afin de préserver la longévité du projet et permettre d'éventuelles traductions des clauses dans d'autres langages informatiques que celui choisi. Enfin, la constitution du clausier s'est achevée par la réalisation de la traduction des clauses en langage informatique pour en faire des *smart contracts* pouvant se déployer sur

la *blockchain*. Une fois le clausier construit, il fallait le rendre accessible gratuitement et facilement. Ainsi, l'équipe de recherche a opté pour l'utilisation d'une plateforme en ligne afin d'héberger le clausier. De cette façon, les travaux de l'équipe sont désormais accessibles facilement.

Paragraphe 2 – Présentation de la plateforme et du site internet

Création de la plateforme en ligne. L'aboutissement du projet fût la création d'une plateforme en ligne afin de rendre les travaux de recherche accessibles et de former ainsi la première librairie européenne ouverte de *smart contracts*. Cette plateforme se trouve sur le site « Opensmartcontract » accessible via ce lien : https://opensmartcontract-front.netlify.app/.
Cette plateforme répond à la double exigence posée par l'équipe de recherche, à savoir satisfaire à la fois la communauté des juristes et celle des informaticiens. En effet, la plateforme propose une interface pédagogique et collaborative qui convient aux professionnels du droit souhaitant découvrir ou utiliser des *smart contracts*, mais également aux informaticiens souhaitant développer des *smart contracts*. Pour accéder à la librairie, il faut aller sur le site « Opensmartcontract »[37]. La page d'accueil du site présente le projet et indique qu'il existe deux modes d'utilisation : un mode consultation, qui permet de consulter les clauses présentes sur le site et de comprendre comment les utiliser, et un mode contribution, qui permet d'ajouter des *smart contracts* ou de proposer des modifications. Ensuite lorsque l'utilisateur clique sur l'onglet clausier, il découvre l'ensemble des clauses dont la traduction est disponible. Il peut alors sélectionner la clause qui l'intéresse et obtenir toutes les informations nécessaires à sa compréhension telle qu'une présentation détaillée de la clause avec sa définition, le contexte dans lequel elle doit être utilisée, les textes de lois liés à cette clause, son régime juridique ainsi que des conseils pour sa rédaction. Enfin, en cliquant sur « voir le Smartcontract » l'utilisateur pourra visualiser le *smart contract* transcrit en langage Solidity.

Collaboration et partage. L'équipe de recherche a porté une attention particulière à l'aspect collaboratif de la librairie. Tout le monde peut consulter les clauses mais également faire des commentaires, des propositions d'amélioration, proposer la traduction de nouvelles clauses, suggérer des améliorations techniques s'agissant des informaticiens ou bien indiquer des précisions sur les textes de loi s'agissant des juristes. La plateforme est en libre accès ce qui

[37] https://opensmartcontract-front.netlify.app/

implique une volonté de la part de l'équipe de recherche de partager son travail avec le plus grand nombre. Le but est de permettre à tous de pouvoir utiliser les modèles de *smart contracts* présents sur la plateforme. C'est aussi la raison pour laquelle la librairie est traduite en français et en anglais. En effet, il est même possible de mélanger les langues lorsqu'on souhaite inscrire des écritures sur la plateforme, que ce soit pour les commentaires ou dans le codage des clauses. Ainsi, le champ linguistique du code est amené à s'élargir au fur et à mesure des contributions. L'ambition portée par l'équipe de recherche d'une ouverture européenne voire internationale fut un succès car le projet a suscité un intérêt certain pour des chercheurs et professionnels européens et même au-delà des frontières de l'Europe. En outre, cette ambition de partage s'observe également à travers le choix d'une plateforme open source et d'une licence de logiciel libre. En effet, en faisant le choix d'utiliser la licence CeCILL, l'équipe permet aux futurs utilisateurs de la librairie de posséder des droits étendus et d'avoir accès au code source de la plateforme. Ainsi, le projet s'inscrit dans une démarche publique et transparente. Cette licence a aussi l'avantage d'ancrer la plateforme dans le droit français et ainsi de définir les règles de responsabilité civile et de droit de la propriété intellectuelle applicables. À ce titre, il faut rappeler que ce qui se trouve sur la plateforme est livrée « *as is* ». Dès lors, l'équipe de recherche s'exonère de toute responsabilité quant à l'utilisation qui sera faite des *smart contracts* proposés sur la plateforme.

Paragraphe 3 – Le développement des *smart contracts* dans le domaine social

La librairie créée, vient le temps du bilan et de la réflexion sur les nombreuses perspectives ouvertes par le projet. L'équipe de recherche a fait le choix de se concentrer sur l'aspect économique et commercial du *smart contract*. Ce choix était pertinent au regard des lignes directrices du projet : utilité, validité juridique et sécurité des parties. Toutefois l'équipe s'est également interrogée au cours de ses recherches sur la perspective de *social smart contracts*. Des *smart contracts* qui s'inscriraient dans le domaine social avec d'autres enjeux que ceux financiers. C'est par la présentation de ces différentes pistes de réflexion qu'il convient de conclure ce rapport afin d'observer le potentiel de développement infini qui entoure le *smart contract*.

Les smart contracts *dans le domaine de l'éducation*

Positions institutionnelles. Les parlements français et européen soulignent les promesses de la *blockchain* dans le domaine de l'éducation notamment pour la délivrance et la vérification des qualifications universitaires. En effet, différents projets d'élaboration d'une *blockchain* pour l'enseignement supérieur se développent. C'est ce qu'illustre le rapport d'information parlementaire (2018) [p.67] « Dans le domaine de l'enseignement supérieur, il en va ainsi du projet d'établir une grande *blockchain* distribuée au sein de laquelle seraient inscrits les diplômes universitaires. Selon les explications fournies par Mme Perrine de Coëtlogon, expert numérique et animatrice à l'université de Lille d'un groupe de travail #Blockchain4EDU, il s'agirait de permettre à tout étudiant de produire ses titres – voire de justifier de compétences spécifiques évaluées par les universités – à partir des informations contenues dans un espace de confiance. De fait, dans le cadre de leur visite à la Station F[38], le plus grand campus de start-up du monde situé à Paris, vos rapporteurs ont pu rencontrer les représentants de start-up qui, à l'instar de BCDiploma[39], développent des protocoles poursuivant cet objectif »[40]. Le rapport ajoute qu'« Un tel usage paraît de nature à conforter le mouvement de dématérialisation et de simplification des procédures et pourrait – sous réserve de progrès techniques – participer à la réalisation de projets aussi ambitieux qu' "Erasmus without paper", avec notamment pour perspectives : l'établissement d'une carte d'étudiant européen, qui éviterait de renouveler certaines démarches dans le cadre d'une inscription ; la création d'un dépôt de logiciels et de ressources universitaires accessibles à l'ensemble des membres d'un réseau d'établissements de l'enseignement supérieur à l'échelle de l'Europe ». On constate alors des perspectives d'utilisation de la *blockchain* et des *smart contracts* dans un objectif de vérification des qualifications universitaires, de certification cryptée dans l'enseignement (par exemple, les « *blockcerts* »), et la mise en place de mécanismes de transferts de crédits automatisés. Les parlementaires soutenant ces technologies et leur développement rappellent néanmoins qu'il convient d'accroître la connaissance et la compréhension de la technologie des registres distribués, afin que cette technologie soit jugée digne de confiance.

Position de l'équipe de recherche. La prudence est de mise. Certes, réaliser des *smart contracts* dans le domaine de l'éducation et déposer des clauses standards sur la plateforme seraient techniquement et juridiquement réalisables. Mais est-ce réellement souhaitable ? Pour

[38] https://stationf.co/
[39] https://www.bcdiploma.com/fr
[40] L. DE LA RAUDIERE et J.-M. MIS, préc., note 1, p. 67.

quelle utilité réelle ? L'équipe de recherche attachée à ce principe consistant à n'utiliser la technologie *blockchain* que dans les situations les plus pertinentes, juge que l'éducation n'est pas un domaine pertinent pour le déploiement de *smart contracts* sur la *blockchain*.

Explication sur la non-pertinence de la *blockchain* pour l'enregistrement des diplômes. Comme Julien Gossa l'avait déjà exprimé à l'occasion du séminaire international, il est sceptique quant à la pertinence de la *blockchain* dans ce domaine. Pour le cadre de ce rapport, il explique les raisons de ce scepticisme.

L'unique objet de la technologie *blockchain* est de créer un consensus entre des acteurs qui ne se font pas confiance. Par consensus, on entend une forme de vérité partagée, qui ne correspond pas forcément à une vérité objective, absolue. Pour toutes les autres caractéristiques d'une *blockchain* (intégrité, transparence, non répudiabilité, etc.) des solutions technologiques plus simples, plus efficaces et moins coûteuses existent. Incidemment, la technologie *blockchain* n'est utile qu'en l'absence de tout tiers de confiance. Dès lors qu'un tiers de confiance existe, il existe également des solutions technologiques plus simples, plus efficaces et moins coûteuses pour rendre le même service. Dès lors, on peut comprendre pourquoi la technologie *blockchain* n'a pas vocation à servir le domaine de l'éducation. Bien que de prime abord, on puisse facilement imaginer une *blockchain* permettant de stocker et retrouver les diplômes de tous les étudiants du monde, la création de cette *blockchain* soulève de nombreuses questions. Qui va en supporter le coût ? Et comment imposer son utilisation ?

Concernant le coût de la *blockchain*. Pour créer un consensus entre acteurs sans confiance, la *blockchain* adopte une approche structurellement très coûteuse : la preuve de travail[41]. Cette preuve de travail a un coût financier pour les mineurs, acteurs chargés de créer le consensus, sans prendre parti dans les transactions. Il faut donc une motivation forte pour que ces mineurs existent. Dans le cadre de cryptomonnaies, la motivation qui a été trouvée est l'attribution d'une part de cette cryptomonnaie aux mineurs. Ainsi, miner rapporte éventuellement plus que cela ne coûte, ce qui a fait le succès du BitCoin.

Dans le contexte de la gestion de diplôme, il est difficile d'identifier une motivation pour des mineurs. À défaut d'une rémunération pécuniaire, il faut éliminer des acteurs indépendants des transactions. Il reste donc : les diplômants, les diplômés et les recruteurs. Il est peu probable que les diplômés et recruteurs soient prêts à payer pour un tel service, d'autant qu'il s'agit d'un investissement, matériel et énergétique, conséquent et à long terme. Les diplômants, notamment

[41]Ou Pow (*Proof of Work*).

les universités, auraient les moyens de miner une telle *blockchain* mais désirent-il investir là-dedans ?

Concernant l'obligation d'utiliser la *blockchain*. Si les diplômants, et notamment les universités, ont les moyens techniques de créer une *blockchain* d'attestation des diplômes, elles ont également les moyens de fournir exactement le même service pour un coût bien moindre. On peut penser par exemple aux projets GEANT[42], qui interconnectent physiquement toutes les universités d'Europe, ou à Eduroam[43] qui permet à tous les personnels et étudiants des établissements partenaires de s'authentifier dans n'importe quel autre. Ces projets montrent que le monde de l'éducation sait se coordonner pour fournir des services internationaux de très grande envergure, sur des problèmes technologiques beaucoup plus complexes que la certification de diplômes sur le plan technique. Dès lors, il apparaît évident pour l'équipe de recherche que la sphère de l'éducation et de l'enseignement supérieur n'est pas une priorité pour le développement de *smart contracts*. Dans ce domaine il existe des solutions techniques plus simples et moins coûteuses[44] pour répondre aux enjeux posés.

Toutefois, si l'on veut mener la réflexion jusqu'au bout, il faut envisager d'autres hypothèses où la *blockchain* pourrait tendre à trouver une utilité en matière d'éducation. Certains soutiennent par exemple qu'en cas de situation politique instable ou de conflit, la *blockchain* pourrait être utile afin de sauvegarder les attestations et diplômes des étudiants même en cas de destruction matérielle des locaux. Cela permettrait aussi aux personnes migrantes fuyant un conflit de conserver leurs attestations. Cependant pour participer à une *blockchain*, il faut *a minima* une connexion internet. Or, dès lors qu'une connexion internet existe, la sauvegarde des données se fera beaucoup plus facilement sur un *cloud*, par exemple, que par une *blockchain*. Ainsi, les données étant délocalisées, elles seront protégées des instabilités locales, et les diplômés migrants pourront accéder à leurs attestations simplement en se rappelant leurs identifiants. Ainsi, même dans cette situation on constate que la technologie *blockchain* ne présente pas l'intérêt requis pour être mise en place.

C'est le même constat qui est opéré lorsque l'on remarque qu'il existe d'une inadéquation fondamentale entre l'objet initial d'une *blockchain* – éliminer tout tiers de confiance – et la nature même des diplômes – une attestation délivrée par un tiers de confiance. Pour utiliser la

[42] https://geant3plus.archive.geant.net/Pages/default.aspx
[43] https://www.eduroam.org/
[44] https://www.letudiant.fr/educpros/actualite/diplomes-attestes-numeriquement-printemps-2019.html

blockchain dans l'éducation, il convient donc d'identifier un cas d'usage sans tiers de confiance, ce qui n'est pas chose aisée dans un domaine très marqué par les références et l'autorité.

Enfin, de manière exploratoire, on peut envisager le cas de la certification de compétences. Il faut éliminer la certification de compétence par les pairs, qui implique une confiance et existe déjà avec des technologies plus simples et moins coûteuses, comme les réseaux sociaux professionnels par exemple. Reste alors la certification de compétence par les tests.

Il faut donc imaginer une communauté ayant un fort intérêt à attester des compétences hors de tout tiers de confiance, donc notamment d'organisme de certification ou de tests de recrutement employeur. Ces compétences devront être purement intellectuelles pour que les tests puissent être en ligne, et se posent aussi le problème de l'authenticité de celui qui passe le test. Ainsi, ce type de contexte n'est envisageable que dans la perspective d'une « ubérisation » si totale qu'elle est difficilement concevable à l'heure actuelle.

En conclusion, développer la technologie *blockchain* dans le domaine de l'éducation en matière d'attestations et de diplômes ne semble pour lors pas pertinente.

Les* smart contracts *dans le domaine de la vie publique

Propositions institutionnelles. Sur l'importance stratégique de la *blockchain* pour les infrastructures publiques, le Parlement européen insiste sur le potentiel d'amélioration de l'efficacité que représente la technologie des registres distribués pour les services et la gestion dans le secteur public, notamment en vue de la mise en œuvre du plan d'action pour l'e-gouvernement en faisant particulièrement référence à l'adoption, à l'échelle européenne, du principe numérique de la transmission unique d'informations, réduisant ainsi davantage la charge administrative pour les citoyens, les entreprises et les administrations publiques. Le Parlement européen souligne également le potentiel de la technologie des registres distribués en matière de décentralisation de la gouvernance et d'amélioration de la capacité des citoyens à rendre les gouvernements responsables. Il invite la Commission à étudier l'amélioration des services publics traditionnels, notamment la numérisation et la décentralisation des registres publics, le cadastre, l'octroi de licences, les certificats à l'usage des citoyens (par exemple, les certificats de naissance ou de mariage) et la gestion des migrations, en particulier par le développement de cas d'utilisations et de pilotes concrets. Il demande aussi à la Commission d'explorer les applications de la technologie des registres distribués qui améliorent les

processus relatifs à la vie privée et à la confidentialité des échanges de données, ainsi que l'accès aux services d'administration en ligne avec une identité numérique décentralisée. S'ajoute à cela la demande à la Commission d'évaluer la sécurité et l'efficacité des systèmes de vote électronique, y compris ceux qui emploient les technologies des registres distribués, tant pour le secteur privé que pour le secteur public.

En ce qui concerne la France, la Mission parlementaire[45] d'information commune sur les chaînes de blocs (*blockchains*) a souligné que « le canton [de Genève] a recours depuis 2017 à une *blockchain* afin d'assurer la certification des données contenues dans le registre du commerce et la délivrance d'actes (65) certifiés de façon automatique » et que « sur la base de résultats très satisfaisants, il est désormais envisagé d'employer la technologie afin d' "industrialiser" la délivrance de nombreux actes administratifs (actes de naissance, de décès, etc.) ». « Le canton entend par ailleurs – avec l'accord de la Confédération – développer l'usage de la signature électronique et permettre l'obtention d'une identité numérique afin de rendre possible des échanges entre les résidents et ses services. Dans le cadre d'une autre *blockchain*, il envisage enfin de donner aux détenteurs de droits à bâtir sur des terrains constructibles en zones industrielles la capacité de les échanger directement »[46]. La proposition n° 6 de la Mission est de créer au sein de la DINSIC[47] un groupe de travail transversal chargé d'une mission d'évaluation des conditions du développement de la technologie des *blockchains* dans la vie économique et sociale et de son usage par les collectivités publiques.

Propositions dans le cadre du projet de recherche. Le professeur Burkhard Schafer a réalisé une note de synthèse sur la question du vote sur la *blockchain*[48]. Le but de son étude est d'établir si les dispositions du Code électoral sont susceptibles d'être conduites par un *smart contract* sur un registre distribué, tel que la *blockchain*. La *blockchain* a été proposée comme un outil pour le vote en ligne afin de répondre aux exigences d'anonymisation, de défaut de confiance et de publicité qui assureraient l'intégrité du vote et de son dépouillement. Après avoir exercé une veille sur des propositions telles que FollowMyvote, il semble cependant que la fonctionnalité du système ne requiert pas l'utilisation de *smart contract*. En effet, mettre en place un vote en ligne sur la base de *smart contracts* aurait pour conséquence, dans de nombreux cas, de polluer certains principes centraux de la loi électorale, car elle lierait nécessairement le

[45] N°1501 Assemblée Nationale, Rapport d'Information par la mission d'information commune sur les chaînes de blocs (*blockchains*).
[46] L. DE LA RAUDIERE et J.-M. MIS, préc., note 1, p. 68.
[47] Direction interministérielle du numérique et du système d'information et de communication de l'État.
[48] La traduction du texte déposé en annexe a été réalisée par Amélie Favreau.

vote à une incitation externe, comme un paiement. L'anonymat du vote est censé garantir le fait qu'on ne peut pas savoir qui a voté quoi et si la personne a voté ou non. Or avec ce système de technologie *blockchain* on produit une preuve du vote réalisé ou non puisqu'il s'inscrit dans la *blockchain*. Ainsi en observant plusieurs des expériences de vote en ligne supportée par un registre distribué, il apparaît que l'emploie du *smart contract* dans un processus de vote en ligne doit rester prudent car il peut s'avérer contraire aux valeurs électorales.

Une exception possible apparaît toutefois. Il s'agit du vote par procuration, dans lequel sont en relation des « votants primaires » et la personne détenant leur procuration. Cette situation pourrait être mise en œuvre par le contrat, avec un avantage possible : le « votant primaire » pourrait vérifier que le vote réalisé correspond au vote qu'il avait souhaité. Ce mécanisme de vote par procuration est développé FollowMyvote notamment dans un contexte de vote des actionnaires dans des sociétés commerciales. Actuellement, plusieurs expérimentations sont en cours dans diverses sociétés. La société Broadridge Financial Solutions a quant à elle enregistré un brevet (http://patents.com/us-9967238.html) pour activer les votes par procuration dans les assemblées générales, lorsque les décisions sont susceptibles d'être déléguées.

Cependant, ce n'est pas simple de traduire ces scénarios de la sphère commerciale aux élections dans la vie publique, parce que le vote en ligne minimise le besoin de vote par procuration. Il demeure que le vote par procuration ou par mandat est l'un des scénarios dans lequel on serait susceptible de retrouver des *smart contracts*, particulièrement si l'autorisation était conditionnelle sous la forme suivante : la personne détenant une procuration peut choisir entre A, B, et C, mais ne peut pas voter pour D. Si toutefois X se réalise (par exemple en cas de déficit présenté par une société commerciale) alors les choix ne seraient que A et B et en excluant C. On en revient de nouveau à l'optique que le *smart contract* se prête davantage aux enjeux du monde des affaires.

Une autre étape de la réflexion menée par l'équipe conduit à envisager d'autres mécanismes de vote politique. Il y a quelques temps, un candidat potentiel a enflammé les journaux. En effet, dans les votes du Parlement britannique (et dans de nombreux parlements construits sur le modèle de Westminster) il est possible « d'appairer » deux députés de partis opposés, afin que leur absence ne change pas le résultat. Dans la pratique parlementaire, cette technique du « pairing » est un arrangement informel entre le gouvernement et les partis d'opposition par lequel un membre du parlement accepte ou est désigné pour s'abstenir de voter à la chambre ou

bien est prié d'en être absent pendant que l'autre parti a aussi besoin qu'un membre soit absent de la chambre pour d'autres engagements, des raisons de maladie ou de voyage.

La confiance sur laquelle est basée ce système a été éprouvée en juillet 2018, quand une membre libérale démocrate du Parlement Jo Swinson, lors de son congé maternité, a été « appairée » avec le président conservateur Brandon Lewis. Malgré l'arrangement en place, Lewis s'est seulement abstenu lors des sept premières sessions, mais a voté avec le gouvernement à l'occasion de deux votes, qui étaient clos. Un *smart contract* aurait permis d'imposer cette obligation, mais la même remarque que pour le vote par procuration ci-dessus vaut également ici : si le vote en ligne avait été possible, « l'appariement » n'aurait pas été nécessaire.

Ainsi, tant les votes par procuration que les votes par « pairing » apparaissent comme étant de bons cas d'études pour les *smart contracts* dans un système de vote.

Plus éloigné des dispositions électorales existantes, l'on pourrait considérer l'intérêt de la *blockchain* dans le cas des droits de vote « conditionnels » – par exemple ceux permis dans des situations électorales spécifiques (dans le cadre d'une démocratie directe), situation où un votant doit avoir à montrer un degré minimum d'engagement. Pour attester de la faculté à voter, il pourrait être envisagé de réaliser un test, par exemple en écoutant des émissions des deux partis en cause, ou avec quelques indicateurs qui pourraient être vérifiés. Ceci servirait à éviter le « vote par ignorance ».

En conclusion, et plus ambitieusement, il existe un certain nombre de paradoxes que le système de vote génère. Un exemple marquant est le « Paradoxe de Condorcet », qui est un problème discursif, ou le Théorème d'Arrow. Quelques conséquences négatives peuvent être résolues à travers plusieurs étapes de la procédure. Bien que ce ne soit pas immédiatement évident pour les *smart contracts*, ils pourraient contribuer à cela, particulièrement sur le paradoxe discursif qui se présente sous la forme d'une « procédure de prémisses et de conclusions » qui traite le vote comme un argument « si-alors » et pourrait être présentée par un *smart contract*. Cela nécessiterait toutefois des recherches ultérieures considérables dans un milieu complexe (choix sociaux), sans garantie de succès.

En conclusion, les situations les plus favorables pour voir l'émergence de *smart contract* sont les systèmes de votes conditionnels par procuration, qui permettent de transférer la capacité de vote à une autre personne, possiblement sous certaines contraintes, et le vote par « *pairing* ».

Ajoutant à la conclusion du professeur Schafer, l'équipe souhaite aussi retenir l'utilité soulignée du *smart contract* dans le processus précédant le vote pour éviter les « votes par ignorance ».

Perspectives pour la recherche. Quelques exemples de *smart contracts* ont déjà été construits pour réaliser des votes sur la *blockchain*[49]. Il sera nécessaire d'effectuer une recherche plus approfondie pour pouvoir les implémenter sur la plateforme si l'équipe de recherche le décide.

Également, quatre exemples sont encore à étudier en matière de vote par la *blockchain*. Le premier est celui abouti de la Sierra Leone, qui est parvenue à réaliser un vote par la *blockchain* le 7 mars 2018 pour des élections présidentielles[50]. C'est la société suisse Agora qui a réalisé cette première mondiale et a signé un Whitepaper pour la décrire[51]. Le second est celui de la Thaïlande[52]. En novembre 2018, la Thaïlande a utilisé la *blockchain* lors des élections primaires du parti démocrate. Sur plus de 120 000 votants, le Premier ministre Abhisit Vejjajiva a remporté l'élection. D'autres initiatives sont à étudier en Inde, en Russie ou dans les Émirats arabes unis, une initiative gouvernementale sur la *blockchain* à l'horizon de 2021. Enfin, plus général, l'exemple de l'Estonie est à approfondir. Dans le domaine de la vie publique, l'Estonie a déjà 99% de ses services publics accessibles en ligne. Or, cette dématérialisation a été possible grâce à la mise en place d'un système informatique décentralisé qui utilise la *blockchain* : X-Road en 2011. Pour authentifier et sécuriser des données, la signature KSI (*Keyless Signature Infrastructure*) empêche l'État ou n'importe quelle organisation de modifier une donnée. De plus, les autorités estoniennes ont créé, via X-Road, un programme de e-residency permettant à n'importe quelle personne, n'ayant pas la nationalité estonienne, de bénéficier d'une identité numérique, facilitant ainsi les investissements.

L'approfondissement de cette recherche s'avère particulièrement pertinent depuis que deux collègues, M. Dino Santaniello, avocat au Laos et M. Buhmindr Butr-Indr, assistant professorat en Thaïlande ont rejoint le projet de recherche. Dino Santaniello soulignait la possibilité de se mettre en rapport avec les autorités du Laos pour mettre en œuvre cette proposition. Il soulignait tant l'intérêt de la *blockchain* dans le processus de vote qu'en tant que registre pour le droit des sociétés. Il est prématuré de dire si de telles clauses pour des raisons de sensibilité du sujet doivent figurer sur la plateforme. L'équipe de recherche préfère rester prudente. Il demeure

[49] Decentralized Voting: A Self-tallying Voting System Using a Smart contract on the Ethereum Blockchain: 19th International Conference, Dubai, United Arab Emirates, November 12-15, 2018, Proceedings, Part I, November 2018.

[50] https://www.weforum.org/agenda/2018/03/the-world-s-first-blockchain-powered-elections-just-happened-in-sierra-leone

[51] https://www.agora.vote/

[52] https://bitcoinmagazine.com/articles/thailand-uses-blockchain-supported-electronic-voting-system-primaries/

qu'une recherche sur les *smart contracts* ne pouvait pas faire l'économie d'un sujet si prometteur.

Les smart contracts *et la santé*

Propositions institutionnelles. Sur le secteur des soins et de la santé, le Parlement européen insiste sur le potentiel de la technologie des registres distribués pour ce qui est de l'amélioration de l'efficacité des données et des rapports d'essais cliniques dans le secteur de la santé. Il reconnaît le potentiel d'amélioration de l'efficacité du secteur de la santé par l'interopérabilité des données électroniques sur la santé, la vérification de l'identité et l'amélioration de la distribution des médicaments. Le Parlement note également que la technologie des registres distribués pourrait permettre aux citoyens de contrôler leurs données de santé, de bénéficier de la transparence à leur sujet, et de choisir les données à partager, y compris en ce qui concerne leur utilisation par les compagnies d'assurance et l'écosystème des soins de santé au sens large. De plus, ces technologies doivent par leur application protéger la confidentialité des données sensibles sur la santé. Ainsi, le Parlement européen invite la Commission à explorer les cas d'utilisation fondés sur la technologie des registres distribués dans la gestion des systèmes de santé et à identifier des cas de référence et des exigences qui permettent des entrées de données de qualité et l'interopérabilité entre les différentes technologies des registres distribués, en fonction du système, du type d'institutions et de leur processus de travail.

Position du groupe de travail. C'est une dimension sur laquelle le groupe de travail perçoit intuitivement un grand intérêt. La recherche reste à réaliser. Le groupe souligne notamment l'intérêt de clauses comme celles liées aux essais cliniques ou à l'accès aux données médicales dans le cadre de contrats d'assurance.

Perspectives de recherche. Les perspectives de recherche sur les *smart contracts* dans le domaine de la santé sont grandes. Ainsi, l'équipe n'a pas eu le temps nécessaire pour formuler des propositions de clauses en lien avec le domaine de la santé à implémenter sur la plateforme. L'équipe soutient que le travail de réflexion doit se poursuivre. Progressivement, cette réflexion fera naître dans ce domaine vaste et prometteur un travail de sélection de clauses qui seraient susceptibles d'être traduites en *smart contract*. Ces clauses pourront être implémentées ultérieurement sur l'architecture existante. Pour

conclure, l'équipe de recherche entrevoit une réelle utilité ouvrir le développement la technologie des *smart contract* dans le domaine de la santé.

BIBLIOGRAPHIE

Articles

Bruno ANCEL, « Les smart contracts : révolution sociétale ou nouvelle boîte de Pandore ? Regard comparatiste », *Communifcation Commerce Électronique*, 1er juillet 2018, n°7, p. 15 à 18.

Ombline ANCELIN et Florent BARBU, « Le droit de la concurrence face aux évolutions du numérique », *Décideurs Juridiques et Financiers*, 1er février 2019, n°17, p. 50 à 51.

Mehdi BALI, « La prise de sûreté sur crypto-monnaie : le cas du Bitcoin », *Revue de Droit Bancaire et Financier*, 1er novembre 2018, n°6, p. 26 à 31.

Alice BARBET-MASSIN et Géraldine GOFFAUX CALLEBAUT, « Blockchain et marché de l'art », *AJ Contrats d'affaires - Concurrence - Distribution*, 1er juillet 2019, n°7, p. 324 à 328.

Eric BARBRY, « Smart contracts... Aspects juridiques ! », *Annales des Mines - Réalités industrielles 2017/3*, août 2017, p. 77 à 80.

Boris BARRAUD, « Les blockchains et le droit », *Revue Lamy droit de l'immatériel*, 1er avril 2018, n°147, p. 48 à 61.

Leslie BENSOUSSAN, « Le smart contract : enjeux juridiques et pratiques », *Revue de Droit Bancaire et Financier*, 1er mars 2019, n°2, p. 111 à 112.

Rodolphe BIGOT, « L'assurance, le droit et le digital : un mauvais remake du "bon, la brute et le truand" ? », *Revue Générale du Droit des Assurances*, 1er janvier 2018, n°1, p. 8 à 22.

Stéphane BLEMUS, « Law and Blockchain : a legal perspective on current regulatory trends worldwide », *RTDF*, 1er octobre 2017, n°4, p. 34 à 49.

Valérie BOCCARA, « Le monde et l'avenir nous appartiennent », *Les Petites Affiches*, 2 avril 2018, n°66, p. 4 à 8.

Céline BONDARD, Grégory CHENU, Hubert DE VAUPLANE, Sylvie DUFOURNAUD et Franck GUIADER, « Quelques utilisations actuelles de cet outil en droit des affaires », *JCP E*, 27 juillet 2017, n°30-34, p. 50 à 55.

Philip BOUCHER, « How blockchain technology could change our lives ? », *Scientific Foresight Unit*, European Parliamentary Research Service, s. d.

Virginie BOUNOT, « La blockchain, outil de gouvernance et de traçabilité du processus créatif (R&D) », *Revue Lamy Droit des affaires*, 1er septembre 2019, n°151, p. 41 à 48.
Jean-Michel BRUGIÈRE, Vincent FAUCHOUX et Alexandre QUIQUEREZ, « Actualité du droit civil du numérique », *Revue Lamy Droit Civil*, 1er septembre 2018, n°162, p. 36 à 41.
Marine CALVO, « De la standardisation au smart contract », *Décideurs Juridiques et Financiers*, 1er novembre 2019, n°25, p. 26 à 27.
Ermanno CALZOLAIO, « Intelligence artificielle et décisions de justice : dans un perspective comparatiste », *Revue Lamy Droit Civil*, 1er novembre 2019, n°175, p. 40 à 44.
Éric CAPRIOLI, Benoît CHARPENTIER, Valérie CHAVANNE, Jérôme DE LABRIFFE, Dominic O'KANE, Christophe ROQUILLY, Arnaud TOUATI et Edouard VIGUIER, « Blockchain et smart contracts : enjeux technologiques, juridiques et business », *Cahiers de droit de l'entreprise*, 1er mars 2017, n°2, p. 9 à 18.
Garance CATTALANO, « Smart contracts et droit des contrats », *AJ Contrats d'affaires - Concurrence - Distribution*, 1er juillet 2019, n°7, p. 321 à 324.
Gaël CHANTEPIE, « De la nature contractuelle des contrats-types », *Revue des contrats*, n°3, 1er juillet 2009, p. 1233.
Gaël CHANTEPIE, « Le droit en algorithmes ou la fin de la norme délibérée ? », *Dalloz IP/IT*, 1er octobre 2017, p. 522 à 526.
Sandrine CHASSAGNARD-PINET, « Les usages des algorithmes en droit : prédire ou dire le droit ? », *Dalloz IP/IT*, 1er octobre 2017, p. 496 à 499.
Mélanie CLÉMENT-FONTAINE, « Le Smart contract et le droit des contrats : dans l'univers de la mode », *Dalloz IP/IT*, 1er octobre 2018, n°10, p. 540 à 543.
Yaël COHEN-HADRIA, « Blockchain : révolution ou évolution ? La pratique qui bouscule les habitudes et l'univers juridique », *Dalloz IP/IT*, 1er novembre 2016, n°11, p. 537 à 542.
Jérôme DEROULEZ, « Blockchain et preuve », *Dalloz Avocats*, 1er février 2017, p. 58 à 62.
Stéphanie DE SILGUY, « Les blockchains, la nouvelle révolution numérique », *Revue Lamy Droit Civil*, 1er mai 2016, n°138, p. 39 à 40.
Hubert DE VAUPLANE, « Blockchain and Conflict of Laws », *RTDF*, 1er octobre 2017, n°4, p. 50 à 52.
Hubert DE VAUPLANE, « Blockchain, cryptomonnaies, finance et droit : état des lieux », *Revue Lamy Droit des affaires*, 1er septembre 2018, n°140 supplt, p. 4 à 13.
Nathalie DEVILLIER, « Jouer dans le "bac à sable" réglementaire pour réguler l'innovation disruptive : le cas de la technologie de la chaîne de blocs », *RTDCom*, Octobre-Décembre 2017, p. 1037 à 1049.

Thibault DOUVILLE, « Blockchain et protection des données à caractère personnel », *AJ Contrats d'affaires - Concurrence - Distribution*, 1er juillet 2019, n°7, p. 316 à 320.
Sébastien DRILLON, « La révolution Blockchain : la redéfinition des tiers de confiance », *Revue Trimestrielle de Droit Commercial (RTD Com)*, 1er octobre 2016, n°4, p. 893 à 900.
Gregorio ESTALLO PUYUELO et Paula TOBAJAS VINUESA, « Blockchain 2.0- Smart contract: ¿Contratos inteligentes? », 4 août 2017.
Amélie FAVREAU, « L'avenir de la propriété intellectuelle sur la blockchain », *Propriétés Intellectuelles*, 1er avril 2018, n°67, p. 11 à 19.
Amélie FAVREAU, « Présentation du projet de recherche sur les smart contracts », *Dalloz IP/IT*, 1er janvier 2019, n°1, p. 33 à 34.
Max GANADO et Steve TENDON, « Legal Personnality for Blockchains, DAOs and Smart contracts », *RTDF*, 1er janvier 2018, n°1, p. 39 à 47.
Emmanuel GAUDET et Valeria STARIKOVA, « Blockchain : panacée ou miroir aux alouettes ? », *Revue Banque*, 1er novembre 2018, n°825, p. 64 à 66.
Fabien GILLIOZ, « Du contrat intelligent au contrat juridique intelligent », *Dalloz IP/IT*, 1er janvier 2019, n°1, p. 16 à 21.
Philippe GINESTIÉ, « La robotisation des contrats - par les juristes eux-mêmes - sera leur prochain eldorado », *Dalloz IP/IT*, 1er octobre 2017, p. 527 à 535.
Lêmy GODEFROY, « Le code algorithmique au service du droit », *Recueil Dalloz Sirey*, 12 avril 2018, n°14, p. 734 à 740.
Lêmy GODEFROY, « La gouvernementalité de Blockchains publiques », *Dalloz IP/IT*, 1er septembre 2019, n°9, p. 497 à 502.
Julien GOSSA, « Les Blockchains et Smart contracts pour les juristes », *Dalloz IP/IT*, 1er juillet 2018, n°7, p. 393 à 397.
Mélanie GOUPY et Gilles KOLIFRATH, « Blockchain : les enjeux en droit français », *Revue internationale des services financiers (RISF)*, 1er octobre 2017, n°2017/4, p. 19 à 24.
Luc GRYNBAUM, « Assurance dans le secteur Santé et Blockchain », *Journal de Droit de la Santé et de l'Assurance Maladie (JDSAM)*, 1er janvier 2018, n°18, p. 67 à 68.
Gaëtan GUERLIN, « Considérations sur les smart contracts », *Dalloz IP/IT*, 1er octobre 2017, n°10, p. 512 à 516.
Elise GUILHAUDIS, « Comprendre la Blockchain à travers l'étude d'un cas pratique : le covoiturage "Blockcar" », *Revue Lamy droit de l'immatériel*, 1er décembre 2017, n°143, p. 53 à 71.

Olivier HIELLE, « La technologie Blockchain : une révolution aux nombreux problèmes juridiques », *Dalloz Actualité*, 31 mai 2016.
Dimitri HOUTCIEFF, « La réactivité en droit contemporain des contrats : des réactions unilatérales au smart contract », *La Gazette du Palais*, 19 juin 2019, p. 9 à 14.
Delphine IWEINS, « La blockchain bouscule le monde des affaires », *Option Droit & Affaires*, 1er février 2017, n°339, p. 5.
Benjamin JEAN et Primavera De FILIPPI, « Les Smart contracts, les nouveaux contrats augmentés ? », *Conseils et Entreprises, La revue de l'ACE*, septembre 2016.
Laurence JOLY, « La Blockchain est-elle une révolution pour la propriété intellectuelle ? », *Dalloz IP/IT*, 1er octobre 2018, n°10, p. 536 à 539.
Markus KAULARTZ et Falco KREIS, « Smart contracts and dispute resolution-A chance to raise efficiency ? », *Bulletin de l'Association Suisse de l'Arbitrage*, 1er juin 2019, n°2/2019, p. 336 à 357.
Rikka KOULU, « Blockchains and online dispute resolution : smart contracts as an alternative to enforcement », *Script'Ed*, 1er Mai 2016, n°13-1, p. 40 à 69.
Mathias LATINA, « Les professions réglementées, des experts du nouveau monde numérique », *Cahiers de droit de l'entreprise*, 1er mai 2018, n°3, p. 52 à 55.
Xavier LAVAYSSIÈRE, « L'émergence d'un ordre numérique », *AJ Contrats d'affaires - Concurrence - Distribution*, 1er juillet 2019, n°7, p. 328 à 332.
Legeais, Dominique. « Fascicule 534 : BLOCKCHAIN », *JurisClasseur Commercial*, 7 mars 2017.
Dominique LEGEAIS, « Regards sur une opération juridique non identifiée : les ICOs », *Dalloz IP/IT*, 1er février 2018, n°2, p. 113 à 117.
Dominique LEGEAIS, « Blockchain et crypto-actifs : état des lieux », *RTDCom*, 1er juillet 2018, n°3, p. 754 à 758.
Thomas LEGLER, « Arbitration of Intellectual property Disputes », *Bulletin de l'Association Suisse de l'Arbitrage*, 1er juin 2019, n°2/2019, p. 289 à 305.
Lawrence LESSIG, « Code is Law – On Liberty in Cyberspace », *Harvard Magazine*, janvier 2000.
Karen LEVY, « Book-Smart, Not Street-Smart: Blockchain-Based Smart contracts and The Social Workings of Law », *Engaging Science, Technolog, and Society*, 2017.
Jan LIEDER, « Blockchain and transfer or shares », *RTDF*, 1er avril 2018, n°2, p. 49 à 56.
Nicolas LOUVET, « Les apports de la blockchain et des actifs numériques au secteur financier », *Dalloz IP/IT*, 1er octobre 2019, n°10, p. 546 à 548.

Maris MALAURIE-VIGNAL, « Blockchain et propriété intellectuelle », *Propriété industrielle*, 1er octobre 2018, n°10, p. 6 à 8.
Enguerrand MARIQUE, « Les smart contracts en Belgique : une destruction utopique du besoin de confiance », *Dalloz IP/IT*, 1er janvier 2019, n°1, p. 22 à 26.
Gaëlle MARRAUD DES GROTTES, « La blockchain, une technologie stratégique pour la France », *Revue Lamy droit de l'immatériel*, 1er janvier 2019, n°155, p. 42 à 44.
Winston MAXWELL et Gauthier VANNIEUWENHUYSE, « Robots Replacing Arbitrators : Smart contract Arbitration », *ICC Dispute Resolution Bulletin*, 1er janvier 2018, n°1, p. 24 à 33.
Mustapha MEKKI, « Droits(s) et algorithmes : de la blockchain à la justice prédictive », *Dalloz Etudiant*, 6 juin 2017.
Mustapha MEKKI, « Les mystères de la blockchain », *Recueil Dalloz Sirey*, 2 novembre 2017, n°37, p. 2160 à 2169.
Mustapha MEKKI, « Le contrat, objet des smart contracts (partie 1) », *Dalloz IP/IT*, 1er juillet 2018, n°7, p. 409 à 417.
Mustapha MEKKI, « Blockchain, smart contracts et notariat : servir ou asservir ? », *JCP N Semaine Juridique*, 6 juillet 2018, n°27, p. 8 à 11.
Mustapha MEKKI, « Le smart contract, objet du droit (partie 2) », *Dalloz IP/IT*, 1er janvier 2019, n°1, p. 27 à 32.
Mustapha MEKKI, « L'intelligence artificielle et le notariat », *JCP N*, 4 janvier 2019, n°1, p. 36 à 48.
Mustapha MEKKI, « Blockchains : entre mystères et fantasmes », *Dalloz IP/IT*, 1er juillet 2019, n°7, p. 415 à 416.
Philippe MEYER, « Blockchain dans la banque : quels développements pour l'avenir ? », *Revue Banque*, 1er juin 2017, n°809, p. 58 à 60.
Jean-Michel MIS, « Les technologies de rupture à l'aune du droit », *Dalloz IP/IT*, 1er juillet 2019, n°7, p. 425 à 428.
Julien MOIROUX, « Commande publique et technologie blockchain : un avenir, mais quel avenir ? », *JCP A*, 17 juillet 2017, n°28, p. 20 à 25.
Julien MOREAU et Olivier POINDRON, « Recommandations juridiques de l'ISDA sur l'utilisation des smart contracts dans le domaine des dérivés », *Option Finance*, 23 avril 2019, n°1507, p. 55.
Emmanuel NETTER, « Blockchain et professions réglementées », *Cahiers de droit de l'entreprise*, 1er mai 2018, n°3, p. 48 à 51.

Bruno PACCIONI et Simon POLROT, « Blockchain : un cadre juridique en chantier », *Option Finance*, 19 septembre 2016, n°1381, p. 48 à 49.
RLDC, « Tous les contrats ne peuvent pas être des smart contracts », *Revue Lamy Droit Civil*, 1er avril 2017, n°147, p. 39 à 42.
Jean-Christophe RODA, « Smart contracts, Dumb Contracts ? », *Dalloz IP/IT*, 1er juillet 2018, n°7, p. 397 à 402.
Burkhard SCHAFER, « Smart social contracts ? Jurisprudental reflections in blockchain enables e-voting », *Dalloz IP/IT*, 1er juillet 2018, n°7, p. 403 à 408.
Mickael SIGDA, « La blockchain outil de l'immobilier de demain ? », *Opérations immobilières*, 1er octobre 2018, n°109, p. 36 à 37.
Jakub J. SZCZERBOWSKI, « Place of smart contracts in civil law. A few comments on form and interpretation », *The 12th Annual International Scientific Conference NEW TRENDS* 2017, 9 novembre 2017.
Eva THÉOCHARIDI, « La conclusion des smart contracts : révolution ou simple adaptation », *Revue Lamy Droit des affaires*, 1er juin 2018, n°138, p. 28 à 38.
Eva THEOCHARIDI, « La conclusion des smart contracts : révolution ou simple adaptation ? », *Revue Lamy Droit Civil*, 1er juillet 2018, n°161, p. 48 à 55.
Xavier VAMPARYS, « Blockchain et droit des sociétés », *JCP E*, 26 avril 2018, n°17, p. 23 à 27.
Xavier VAMPARYS, « Blockchain : quelques réflexions sur la confiance 2.0 », *JCP E Semaine Juridique*, 11 octobre 2018, n°41, p. 47 à 49.
Gauthier VANNIEUWENHUYSE, « Arbitration and New Technologies : Mutual Benefits », *Journal of international arbitration*, 1er février 2018, n°35-1, p. 119 à 129.
Thibault VERBIEST, « Quelle valeur juridique pour les smart contracts ? », *Revue Lamy Droit des affaires*, 1er septembre 2017, n°129, p. 35 à 37.
Mark VERSTRAETE, « The Stakes of Smart contracts », *Loyola University Chicago Law Journal*, 17 mai 2018.
Loland VINGIANO-VIRICEL, « Quel usage de la donnée en assurance ? », *Revue Générale du Droit des Assurances*, 1er août 2019, n°8, p. 47 à 53.
Bélinda WALTZ-TERACOL, « Blockchain et assurance : entre mythe et désillusion », *Revue Générale du Droit des Assurances*, 1er novembre 2019, n°11, p. 5 à 12.
Kevin WERBACH et Nicolas CORNELL. « Contracts Ex Machina », *Duke Law journal*, 2017.

Célia ZOLYNSKI, « Blockchain et smart contracts : premiers regards sur une technologie disruptive », *Revue de Droit Bancaire et Financier*, 1er janvier 2017, n°1, p. 85 à 88.
Célia ZOLYNSKI, « La Blockchain : la fin de l'ubérisation ? », *Dalloz IP/IT*, Juillet-Août 2017, p. 385 à 387.
Contributeur invité, « Smart contracts for the layman: Pros and cons of the new way of building relationships », *TechBullion*, 31 mai 2018.

Rapports et contributions diverses

Bruno DONDERO, « Les smart contracts », *Actes du colloque du Master 2 Droit Privé Général et du laboratoire de Droit Civil - Paris II*, 21 avril 2017.
Conseil supérieur de la propriété littéraire et artistique, « Rapport de la mission sur l'état des lieux de la blockchain et ses effets potentiels pour la propriété littéraire et artistique », 13 février 2018.

Mémoires

Aurélie BAYLE, « Analyse prospective des smart contracts en droit français » (année universitaire 2016-2017).

ANNEXES

Annexe 1. Valorisation de la recherche

Séminaires et conférences

Séminaire international. La première action menée a été de réaliser un **séminaire international**, afin de lancer officiellement le projet de recherche. Nous souhaitions qu'il soit à l'image de notre recherche, à savoir inscrit dans une démarche d'ouverture. Ainsi, il a été l'occasion de faire intervenir des professionnels extérieurs au projet et/ou internationaux. L'une des conséquences a été de susciter l'intérêt de Maître Fabien Gillioz qui a par la suite rejoint l'équipe de recherche avec M. Yaniv Benhamou. Le séminaire international s'est tenu **le vendredi 16 mars 2018**, dont le programme est annexé au présent rapport (Annexe 2).

D'un point de vue scientifique, le séminaire a été l'occasion de rappeler que le « *smart contract* » est un contrat « autoexécuté » et non un contrat « autoformé ». Il a été souligné à quel point l'expression « *smart contract* » pouvait être un « faux-ami ». Il s'agit donc d'un outil, qui plus est facultatif, employé au moment de l'exécution du contrat dans le but de le « renforcer » et de l'« augmenter ». La recherche ne portera donc pas sur les questions relatives à une formation automatique du contrat. En effet, au-delà de sa mise en œuvre très hypothétique, elle mobilise d'autres techniques, comme celle de l'intelligence artificielle et sollicite les dispositions sur la formation du contrat, principalement celles relatives au consentement, qui ne sont pas abordées dans le projet. Le séminaire nous a également permis d'orienter les groupes de travail.

WEB CONFÉRENCE : PRÉSENTATION DE LA PREMIÈRE LIBRAIRIE EUROPÉENNE DE *SMART CONTRACTS* À DESTINATION DES PROFESSIONNELS DU DROIT ET DE LA JUSTICE

LE 18 SEPTEMBRE 2020

Pouvoir exécuter automatiquement une clause de préemption sur des actifs immatériels, une clause « *buy or sell* » ou un contrat à terme, tel sont les potentialités que nous offrent les *smart contracts*.

Déployés sur une *blockchain* afin de bénéficier de ses caractéristiques (transparence, sécurité, validation des transactions par un consensus distribué), les *smart contracts* sont une ressource

informatique à disposition des professionnels du droit et de la justice. Ils contribuent à une nouvelle forme d'écriture du droit. Le *smart contract* est la traduction en langage formel de clauses issues de contrats traditionnels.

Le projet de recherche « *Smart contracts* » a reçu le soutien de la Mission de recherche Droit et Justice en mars 2018 (pour en savoir plus : https://smart-contracts.univ-grenoble-alpes.fr/). Il s'est clôturé par une webconférence le vendredi 18 septembre 2020.

Une équipe internationale de professionnels (avocats, juristes, entrepreneurs) et de chercheurs en droit et en informatique ont présenté la première plateforme européenne en open source répertoriant des *smart contracts* sous la forme d'une librairie. Ce séminaire a été l'occasion de présenter les résultats de recherche qui ont environné la réalisation de la plateforme et enrichi la connaissance sur les *smart contracts*.
Qu'est-ce qu'un *smart contract* ? Quels potentiels présentent les *smart contracts* déployés sur une *blockchain* pour les professionnels du droit et de la justice ?
Plus spécifiquement, quelles sont les clauses que l'équipe a choisi de traduire ? Toute clause d'un contrat peut-elle être traduite en langage formel, en *smart contract* ? Peut-on se passer de l'écriture du droit en langage naturel ? Et, d'un point de vue technique, dans le paysage des langages propres aux *smart contracts*, lequel choisir ? Existe-t-il une méthodologie pour opérer cette nouvelle forme d'écriture du droit ?

Les membres du projet de recherche se sont donné pour objectif de fournir un cadre juridique au *smart contract*. Notre droit possède-t-il les règles nécessaires à cet encadrement ou doit-on envisager l'élaboration de dispositions spécifiques ? De nouvelles pratiques entraînent de nouvelles responsabilités, quelles sont-elles ?

Enfin, face aux développements à venir de cette technologie, quels doivent être les principes fondamentaux de l'écriture du droit en langage formel ? Comment protéger notre droit des sirènes libérales venues des États-Unis ? Existe-t-il une éthique du *smart contract* ?

Publications

« LES *SMART CONTRACTS* SUR LA *BLOCKCHAIN* ». *DALLOZ IP/IT*

LE 10 JUILLET 2018

Première publication du groupe de recherche : *Dalloz IP/IT*, n°7-8, Juillet-Août 2018, p. 392, Dossier : « Les *smart contracts* sur la *blockchain* ».

Le 16 mars 2018 s'est tenu le séminaire de lancement d'une recherche conduite par une équipe franco-britannique et pluridisciplinaire de vingt-deux membres sur les « *smart contracts* » et soutenue par la Mission de recherche Droit et Justice (2018-2020). Il a été organisé à l'université Grenoble-Alpes, en collaboration avec le barreau de Grenoble, par le Centre universitaire d'enseignement et de recherche en propriété intellectuelle et le Centre de recherches juridiques (CUERPI CRJ). Les interventions de professionnels et d'universitaires de quatre nationalités différentes, membres ou associés à l'équipe de recherche sont présentées dans ce dossier. Il sera publié en deux parties sous la direction d'Amélie Favreau, maîtresse de conférences HDR en droit privé (CRJ), responsable scientifique du projet de recherche sur les « *Smart contracts* ».

Avec les contributions de :

J. GOSSA, Les *blockchains* et *smart contracts* pour les juristes, Maître de conférences informatique à l'université de Strasbourg

J.-C. RODA, Smart Contracts, Dumb Contracts, Professeur agrégé des Facultés de droit, à l'université Jean-Moulin, Lyon III.

B. SCHAFER, Smart social contract. Jurisprudential reflections in blockchain enabled e-voting, Professeur en informatique juridique à la Faculté de droit de l'université d'Edimbourg.

M. MEKKI, Le contrat, objet des *smart contracts* (Partie 1), Professeur agrégé des Facultés de droit à l'université Paris 13.

« LES *SMART CONTRACTS* SUR LA *BLOCKCHAIN* – APPROCHE DE DROIT COMPARE ». *DALLOZ IP/IT*

Janvier 2019

Le 16 mars 2018 s'est tenu le séminaire de lancement d'une recherche conduite par une équipe franco-britannique et pluridisciplinaire de vingt-deux membres sur les « *smart contracts* » à l'université Grenoble-Alpes. Ce dossier est l'occasion d'en présenter la seconde partie. Ce dossier inscrit la démarche internationale de l'équipe de recherche, à travers la vision finlandaise de la *blockchain* dans les droits de propriété intellectuelle et les points de vue belge, suisse et français sur les *smart contracts*. Ce dossier sera conclu par une présentation du projet de recherche.

Avec les contributions de :

Rosa Maria Ballardini, Senior Lecturer in IP law, University of Lapland et **Olli Pitkänen**, Research Director, IPR University Center, Hanken School of Economics : Balancing Exclusive Rights and Access to Technologies: Blockchain and Intellectual Property Rights

Fabien Gillioz, Du contrat intelligent au contrat *juridique* intelligent, Avocat et associé en l'Étude Ochsner & Associés à Genève en Suisse

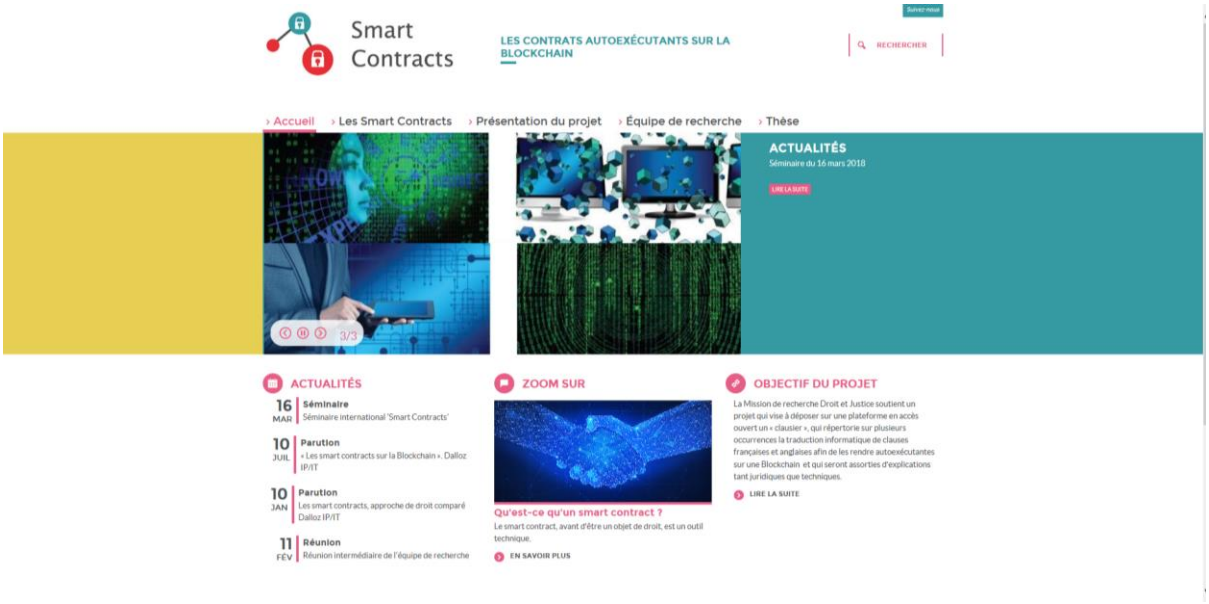
Enguerrand Marrique, Les *smart contracts* en Belgique : une destruction utopique du besoin de confiance, Doctorant FSR et chercheur à la faculté de droit de l'université catholique de Louvain, Belgique

Mustapha Mekki, Le *smart contract*, objet du droit (Partie 2), Professeur agrégé des Facultés de droit à l'université Paris 13.

Amélie FAVREAU, Présentation du projet de recherche sur les *Smart contracts***,** Maître de conférences HDR en droit privé, à l'université Grenoble-Alpes, en Délégation CNRS, Chercheur invitée au SCRIPT, Université d'Édimbourg (Royaume-Uni).

Site Internet

Réalisation du site internet. Toujours animés de la même intention de visibilité de la recherche, nous avons fait réaliser un site internet, qui est hébergé par l'université Grenoble-Alpes, à l'adresse suivante : https://smart-contracts.univ-grenoble-alpes.fr/.



D'une manière générale, la recherche sur la *blockchain* est foisonnante. Il est donc indispensable de faire connaître nos différentes actions et l'avancée de notre réflexion. Le site internet que nous avons réalisé est en ce sens, nous l'espérons, une jolie « vitrine » du projet.

L'entreprise a toutefois été longue et complexe. En effet, après avoir déterminé quel devaient être le contenu et l'organisation du site, nous avons fait face aux limites de nos compétences en qualité de webmestre et au manque de disponibilité que l'entreprise nécessitait. Nous avons décidé de faire appel à une étudiante de 1re année du DUT Carrières Juridiques, **Mlle Margaux Gressard** (IUT2, université Grenoble-Alpes), qui possédait les qualités requises et a suivi une formation pour le logiciel Drupal. Pour le financement de son stage, nous avons candidaté à un récent dispositif mis en place par l'université Grenoble-Alpes, les « Stages d'excellence », qui permettent aux meilleurs étudiants de première et deuxième année de découvrir les métiers de la recherche à l'université, quelle que soit la discipline. La plus grande partie de la programmation a été réalisée pendant l'été 2018. Pour la mise en ligne du site internet qui a eu lieu le 9 janvier 2019, il a été nécessaire de procéder à quelques ajustements qui ont nécessité

à nouveau l'intervention de Margaux Gressard, que nous avons pu financer grâce aux vacations budgétées dans le projet (Partie III). Un graphiste de l'université Grenoble-Alpes a réalisé un logo, que nous utilisons dans nos documents de communication sur le projet.

Nous avons eu d'excellents retours sur l'ergonomie et le contenu du site Internet. Depuis le mois de janvier, nous mesurons à quel point il est un excellent moyen de communication et de présentation du projet de recherche lorsque nous entrons en contact avec de nouvelles personnes intéressées par la recherche. Nous envisageons de réaliser une version en anglais.

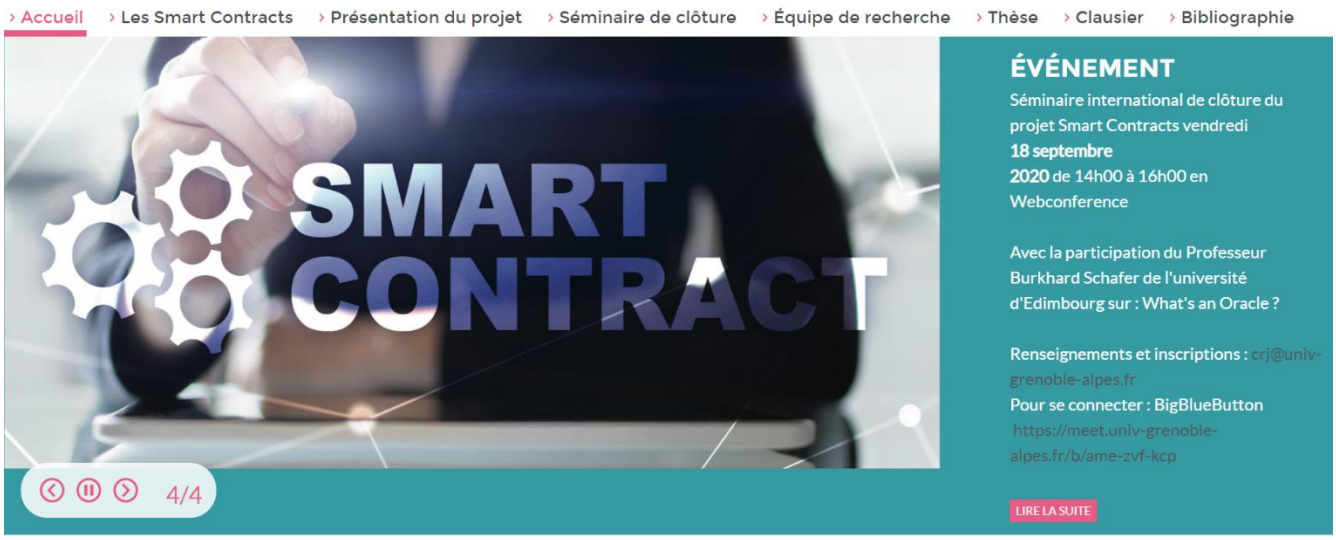
Blockchain Summit. Avec l'accord préalable de la Mission Droit et Justice, nous avons assisté au *Blockchain Summit*, qui s'est tenu les 30 et 31 janvier 2019 à Paris. Cette manifestation a été l'occasion d'établir un état de l'art et de la réflexion sur la technologie *blockchain*. Les avancées technologiques sont particulièrement rapides dans le domaine et implique un suivi assidu et rigoureux. Au-delà, la réalisation du clausier nécessite d'avoir une prise de contact avec les différents acteurs du marché de la *blockchain*. En effet, l'évolution du projet de recherche nous conduira à devoir sélectionner la *blockchain* la plus pertinente et possiblement lancer un appel à manifestation d'intérêt (Partie III). L'entreprise Crypto4all rencontrée à cette occasion nous semble avoir un potentiel intéressant.

Crypto4all est une société française d'ingénierie spécialisée dans la technologie *blockchain*. Depuis 2015, Crypto4all accompagne des entreprises qui souhaitent développer des produits et/ou services basés sur la technologie *blockchain* dans des secteurs extrêmement variés tels que finance, immobilier, automobile, paiement électronique, pharmaceutique, agriculture. Ils conçoivent l'architecture de la solution et développent la solution conformément au cahier des charges et aux spécifications techniques définies avec leur client. Ils sont agnostiques d'un point de vue technologique, c'est-à-dire qu'ils sélectionnent le type de *blockchain* le plus adapté au(x) besoin(s) et la plus à même d'assurer la pérennité de la solution.

Les *smart contracts* constituent aujourd'hui l'un des principaux éléments sur lesquels reposent un bon nombre de projets basés sur la technologie *blockchain* et sont donc une composante essentielle des solutions qu'ils conçoivent. Ils réalisent également des audits de sécurité.

Communication sur le projet. Le projet a fait l'objet de différents signalements sur des plateformes de recensement. Nous l'avons intégré à l'initiative MAP de l'EU Blockchain Observatory, https://www.eublockchainforum.eu/initiative-map qui répertorie tous les projets *blockchains* dans le monde. Également, nous avons inscrit le projet sur le site de la Direction générale des entreprises qui recense tous les projets français et leur niveau de maturité,

https://www.entreprises.gouv.fr/numerique/recensement-des-projets-reposant-sur-la-technologie-blockchain.



2/4

ACTUALITÉS

2020

18 SEP — **Séminaire**
Web conférence : Présentation de la première librairie européenne de smart contracts à destination des professionnels du droit et de la justice

2019

11 FÉV — **Réunion**
Réunion intermédiaire de l'équipe de recherche

10 JAN — **Parution**
Les smart contracts, approche de droit comparé Dalloz IP/IT

ZOOM SUR

Qu'est-ce qu'un smart contract ?

Le smart contract, avant d'être un objet de droit, est un outil technique.

EN SAVOIR PLUS

OBJECTIF DU PROJET

La Mission de recherche Droit et Justice soutient un projet qui vise à déposer sur une plateforme en accès ouvert un « clausier », qui répertorie sur plusieurs occurrences la traduction informatique de clauses françaises et anglaises afin de les rendre autoexécutantes sur une Blockchain et qui seront assorties d'explications tant juridiques que techniques.

LIRE LA SUITE

ZOOM SUR

Séminaire de clôture

Web conférence vendredi 18 septembre 2020 : présentation de la première librairie européenne de smart contracts à destination des professionnels du droit et de la justice

EN SAVOIR PLUS

Avenir de la recherche

Thèse en cours

Le sujet est : « **L'écriture du droit des contrats autoexécutants sur la blockchain. Analyse et perspective** », sous la codirection de Mme Sihem Amer-Yahia, directrice de recherche au CNRS, membre du Laboratoire d'informatique de Grenoble (LIG) et Mme Amélie Favreau, maîtresse de conférences en droit privé, HDR, membre du Centre de recherches juridiques (CRJ).

Le sujet consiste à analyser la démarche de traduction du droit en code dans les *smart contracts* et à contribuer à l'établissement de standards et de bonnes pratiques.

Le travail doctoral se concentre autour de trois principaux objectifs scientifiques : améliorer la connaissance en droit et en informatique des contrats autoexécutants sur la *blockchain*, contribuer à la performance des contrats autoexécutants et participer à une nouvelle écriture du droit à destination des professions judiciaires et juridiques. Ces objectifs s'inscrivent dans une démarche de recherche fondamentale, qui ambitionne de répondre à l'excellence internationale.

Le doctorant bénéficie d'un contrat doctoral à l'université Grenoble-Alpes.

Abdoulaye DIALLO est un juriste spécialisé en droit des affaires et en droit des activités numériques ; passionné par l'informatique, il sait aussi développer dans plusieurs langages de programmation, dont Solidity, un langage d'écriture des *smart contracts*. Aujourd'hui il est doctorant au sein de l'université Grenoble Alpes et il travaille sur le sujet de l'écriture du droit dans les *smart contracts*.

Annexe 2. Composition de l'équipe et constitution de groupes de travail

Composition de l'équipe

La composition initiale de l'équipe de recherche. L'équipe est initialement constituée de 22 membres avec :

- **Des enseignants et chercheur de l'UGA et du CNRS Grenoble** : AMER-YAHIA Sihem, Directrice de recherche CNRS (CNRS, LIG) ; BENINNI Aïda Maîtresse de conférences (UGA, CESICE), BOISSON Alexis, Maître de conférences (UGA, CRJ), BRUGUIERE Jean-Michel, Professeur (UGA, CRJ), FAVREAU Amélie, Maîtresse de conférences (UGA, CRJ), MRAOUAHI Sabrina, Maîtresse de conférences (UGA, CRJ), RENAUD Pascale, Maîtresse de conférences (UGA, CRJ), SASSOLAS Delphine, Maîtresse de conférences (UGA, CRJ), TREFIGNY Pascale, Professeure (UGA, CRJ), VIDELIN Jean-Christophe, Maître de conférences (UGA, CRJ), VILAY Alexandre, Doctorant (UGA, CRJ),

- **Des enseignants et chercheur hors UGA** : DEPINCÉ Malo, Maître de conférences (Université de Montpellier, CDCM), FATHISALOUT Motaharet, Maître de conférences (USMB, CDPPOC), GOSSA Julien, Maître de conférences (Université Strasbourg, ICUBE), LE BOURG Johann, Maître de conférences (USMB, CDPPOC)

- **Des enseignants et chercheur Université d'Édimbourg (*SCRIPT*)** : Burkhard SCHAFER Professeur (Université d'Édimbourg, *SCRIPT*), JONDET Nicolas Enseignant-Chercheur (Université d'Édimbourg, *SCRIPT*).

- **Professionnels du droit, de la justice et du numérique** : BOUSSAID Emeric, Avocat inscrit au barreau de Chambéry (PAST IUT2), CHARBONNEL Lionel, Avocat inscrit au barreau de Marseille, JEAN Benjamin, Consultant Inno3 et Président Open Law, TAMBOURG Olivier, Avocat inscrit au barreau de Grenoble (PAST IUT2), ZORN Caroline, Avocate inscrite au barreau de Strasbourg.

Évolution de l'équipe de recherche. L'implication de chacun sur un projet long est fonction des nouvelles charges administratives et scientifiques. La composition de notre équipe de recherche a de ce fait beaucoup évolué tout en conservant une dynamique forte qui a permis la réalisation de la librairie.

Grâce à un travail de communication et de valorisation du projet de la part des membres de l'équipe initiale, nous avons pu rallier à notre entreprise des forces et des compétences précieuses tant professionnelles qu'universitaires. Elles s'inscrivent pleinement dans la trajectoire pluridisciplinaire et internationale de notre équipe. Notre équipe de recherche compte maintenant 9 membres associés nationaux et internationaux : 5 professionnels, 4 chercheurs. Notre projet de recherche est donc aujourd'hui conduit entre la France, le Royaume-Uni, la Suisse, la Thaïlande, le Laos.

- Maître Élise Guilhaudis, Avocate inscrite au barreau de Grenoble et Gérante de la Legaltech NUMETIK Avocats,
- M. Bhumindr Butr-Indr, Maître de conférences à la Faculté de droit de Thammassat (Bangkok, Thaïlande)
- Maître Dino Santaniello, Consultant, responsable du cabinet juridique Tilleke and Gibbins
- Maître Fabien Gillioz, Avocat associé, Cabinet Ochsner et Associé, Genève (Suisse).
- Mme Guhlan Pinar, Doctorante, Université Grenoble-Alpes.
- M. Yaniv Benhamou, Avocat, Maître de conférences à Genève (Suisse)
- Christine Hennebert, Ingénieure de recherche, CEA, Grenoble
- Thomas Féraud, CentraleSupélec, Université Paris Saclay, France

Évolution de l'équipe de recherche : doctorant. Le 25 octobre 2018, nous avons obtenu le soutien de l'université Grenoble-Alpes pour un contrat doctoral. Le sujet de la thèse est directement en lien avec le projet de recherche soutenu par la Mission Droit et Justice. Il s'inscrit en complément. Il s'agit de « **L'écriture du droit des contrats autoexécutants sur la blockchain. Analyse et perspective** ». La thèse présentant les doubles aspects de droit et d'informatique est réalisée sous la codirection de Mme Sihem Amer-Yahia, directrice de recherche au CNRS, membre du Laboratoire d'informatique de Grenoble (LIG) et Mme Amélie Favreau, maîtresse de conférences en droit privé, HDR, membre du Centre de Recherches Juridiques (CRJ). Le candidat recruté est Monsieur Abdoulaye DIALLO. Il a été recruté pour

ses qualités de juristes, ses très bons résultats obtenus en Master 2 Droit du numérique (Université Paris 5) en attestent et sur les recommandations de la codirectrice du Master, Mme Caroline LE GOFFIC. Il possède également de solides compétences en informatique. Il sait notamment développer dans plusieurs langages de programmation, dont Solidity, un langage d'écriture des *smart contracts*.

Evolution de l'équipe de recherche : stagiaires. Les forces vives de l'équipe ont été renforcées par l'arrivée de stagiaires. Margaux GRESSARD, étudiante en 1re année du DUT Carrières juridiques à l'IUT2 de Grenoble a réalisé le site internet du projet de recherche. Nathan ROUGIER, étudiant en 2e année de DUT Informatique à IUT2 de Grenoble a contribué à la rédaction des *smart contracts*. Julien FONTRIER, étudiant en 2e année de DUT Carrières juridique à IUT2 de Grenoble a participé à la présentation des clauses sur la librairie, a constitué une bibliographie exhaustive sur le sujet et a enrichi le site internet du projet de recherche et enfin a participé à l'organisation et à la valorisation du séminaire de clôture (18 septembre 2020). Enfin, Anna COMPANY, étudiante en 3ème année de Licence de Droit à l'Université Grenoble Alpes a collaboré à l'écriture de ce rapport.

Quel impact pour la plateforme de *smart contract* ? Outre l'apport de compétences indispensables pour cette recherche de chercheurs ancrés dans la pratique du droit, l'étude d'autres systèmes juridiques viendra enrichir cette réflexion. En effet, le clausier tel que nous l'avions envisagé était trilingue avec la présentation de clauses présélectionnées françaises, britanniques et traduites en langage informatique pour assurer leur autoexécution sur une blockchain. L'intérêt de l'ouverture internationale de la recherche est de pouvoir implémenter de nouvelles stipulations issues de systèmes étrangers. En droit suisse, notamment, la recherche sur les *smart contracts* est particulièrement avancée.

Le dynamisme de l'équipe de recherche et l'intérêt du sujet nous ont conduits à associer de nouveaux membres au projet de recherche pour bénéficier de leur expertise. Celui-ci compte maintenant 30 membres, dont 9 professionnels du droit, 6 chercheurs internationaux (Royaume-Uni, Suisse, Thaïlande, Laos), 3 doctorants.

Constitution des groupes de travail

Réunion des groupes de travail. Il avait été initialement avancé par les membres de l'équipe de travailler sur « Slack.com ». Il s'agit d'un outil collaboratif qui offre à chacun un espace de travail partagé où les conversations sont organisées et accessibles. Il est ainsi très avantageux de pouvoir consulter les archives des conversations, des décisions et des séances de travail. Pour autant, il n'a pas été adopté par les membres du projet de recherche, qui ont préféré des solutions plus « classiques » de communication et d'échange à travers des documents électroniques partagés (Google doc). Il est enfin à noter que M. Abdoulaye Diallo, depuis le début de ses études doctorales, contribue à l'ensemble des groupes de travail du projet de recherche et les enrichit de sa réflexion et de ses recherches.

Groupe de travail 1. Sur l'outil informatique smart contract

Ce groupe de travail est composé de :

- **Sihem AMER-YAHIA (coordinatrice), DR CNRS, Université Grenoble-Alpes, LIG (Informatique),**
- Abdoulaye DIALLO, Doctorant UGA (nouvelles technologies, informatique),
- Amélie FAVREAU, MCF, Université Grenoble-Alpes (Propriété intellectuelle et nouvelles technologies),
- Julien GOSSA, MCF, Université Strasbourg, ICube (Informatics),
- Benjamin JEAN, Consultant, Inno3, Open Law Association (Propriété intellectuelle et nouvelles technologies).
- Nicolas JONDET, Chercheur, Université Édimbourg, *SCRIPT* (Propriété intellectuelle et nouvelles technologies)
- Thomas FERAUD, CentraleSupélec, Université Paris Saclay, France

Le registre distribué de la *blockchain* serait par sa lourdeur, sa lenteur et sa consommation énergétique un moyen à écarter au profit d'autres technologies, qui répondraient aussi bien voire mieux aux attentes des utilisateurs. L'un des objectifs du groupe est donc de déterminer quand

la technologie *blockchain* présente un véritable bénéfice. Le séminaire a notamment mis en évidence que la *blockchain* possède un intérêt en l'absence ou en cas de défaillance ou de défiance d'un organisme centralisateur (un État, une banque, etc.).
En effet, si la technologie *blockchain* est utilisée pour répliquer sans plus-value le rôle d'un organe centralisateur que l'on pourrait qualifier de « sain », le *smart contract* sera nécessairement désavoué. Cette question est centrale dans le projet de recherche pour effectuer le travail de sélection nécessaire aux clauses qui figureront dans le clausier.
Le groupe ambitionne enfin de déterminer le langage le plus approprié pour réaliser la plateforme.

Groupe de travail 2. Sur les economical smart contracts

Ce groupe de travail est composé de :

- **Alexis BOISSON (coordinateur), MCF, Université Grenoble-Alpes CUERPI- CRJ (Propriété intellectuelle et nouvelles technologies)**
- Aïda BENNINI, MCF, Université Grenoble-Alpes CESISE, (Droit des affaires)
- Émeric BOUISSAID, Avocat, MCF associé Université Grenoble-Alpes (Droit commercial)
- Jean-Michel BRUGUIERE, Professeur, Université Grenoble-Alpes (Propriété intellectuelle)
- Lionel CHARBONNEL, Avocat, Docteur en droit, Marseille (Droit de la Consommation, des assurances)
- Abdoulaye DIALLO, Doctorant UGA (nouvelles technologies, informatique),
- Amélie FAVREAU, MCF, Université Grenoble-Alpes (Propriété intellectuelle et nouvelles technologies)
- Olivier TAMBOURG, Avocat, MCF associé Université Grenoble-Alpes (Droit des affaires)
- Pascale TREFIGNY, Professeur, Université Grenoble-Alpes (Propriété intellectuelle).
- Alexandre VILAY, Doctorant, Université Grenoble-Alpes CUERPI-CRJ (Propriété intellectuelle et nouvelles technologies)

Et des membres associés au projet :

- M. Yaniv BENHAMOU, Avocat, Maître de conférences à Genève (Suisse)
- Maître Fabien GILLIOZ, Avocat associé, Cabinet Ochsner et Associé, Genève (Suisse).
- Maître Élise GUILHAUDIS, Avocate inscrite au barreau de Grenoble et Gérante de la Legaltech NUMETIK Avocats,
- Mme Guhlan PINAR, Doctorante, Université Grenoble-Alpes.

Il s'agit de la vision la plus usuelle de la technologie du *smart contract* : mettre en œuvre l'exécution automatique d'un contrat, le plus souvent liée à la réalisation de la contrepartie financière du contrat qui a été convenue par les parties. Même si elles ne sont encore qu'au stade de l'expérimentation, de nombreuses applications de *smart contracts* se profilent dans les domaines de l'assurance, de la location, des sociétés (pactes d'actionnaires), de l'immobilier, etc. Les problématiques traditionnelles sur le développement des *smart contracts*, soulignées lors du dépôt de ce projet de recherche, sont liées à la technique contractuelle mise en perspective d'une programmation informatique. Ce sont alors les questions de codage de notions à contenu variable, de révocation unilatérale du contrat en cas d'inexécution contractuelle (sauf à considérer que le *smart contract* n'a jamais vocation à s'inexécuter), etc. Au-delà, sa forte propension à s'installer dans les contrats d'adhésion doit toutefois interroger sur la philosophie sous-jacente. En effet, il a été souligné lors du séminaire que le vocabulaire employé par les promoteurs anglo-saxons du *smart contract* est proche de celui de l'École de Chicago, qui soutient une vision libérale de l'économie. Les algorithmes ne sont pas neutres. L'intérêt de constituer un groupe de travail sur cette thématique économique du *smart contract* permettra au-delà de la stricte identification des clauses susceptibles d'être programmées informatiquement de mesurer leur impact dans l'équilibre contractuel. La démarche portant initialement sur la technique contractuelle sera doublée très utilement de cette mise en perspective philosophique et économique.

Groupe de travail 3. Sur les social smart contracts

Ce groupe de travail est composé de :

- **Nicolas JONDET (coordinateur), Chercheur, Université d'Édimbourg, SCRIPT (Propriété intellectuelle et nouvelles technologies)**

- Sihem AMER-YAHIA, DR CNRS, Université Grenoble-Alpes, LIG (Informatique)
- Abdoulaye DIALLO, Doctorant UGA (nouvelles technologies, informatique),
- Amélie FAVREAU, MCF, Université Grenoble-Alpes (Propriété intellectuelle et nouvelles technologies)
- Julien GOSSA, MCF, Université Strasbourg, ICube (Informatics)
- Benjamin JEAN, Consultant, Inno3, Open Law Association (Propriété intellectuelle et nouvelles technologies)
- Burkhard SCHAFER, Professeur, Université Édimbourg, SCRIPT (Propriété intellectuelle et nouvelles technologies)
- Caroline ZORN, Avocat, Docteur en droit, Enseignante Université Strasbourg (Droit de la santé et droit des données personnelles)

Et des membres associés au projet :

- M. Bhumindr BUTR-INDR, Maître de conférences à la Faculté de droit de Thammassat (Bangkok, Thaïlande)
- Maître Dino SANTANIELLO, Consultant, responsable du cabinet juridique Tilleke and Gibbins

L'exemple développé lors du séminaire était celui du vote et de l'utilité d'employer les *smart contracts* et la *blockchain* dans ces domaines d'exercice démocratique. Le consensus distribué permet de ne pas lier l'exécution d'une action à sa validation par une personne de confiance. Une analyse comparée des systèmes de vote pourrait mettre en évidence les situations dans lesquelles les *smart contracts* présenteront un intérêt. La recherche sur les *smart contracts* élargit l'analyse aux domaines de la santé et de l'éducation. Ils sont en effet d'une particulière acuité sur la *blockchain* et soulèvent des réflexions communes. D'une part leur utilité est liée à celle de l'organe centralisateur qui implique de resserrer leur emploi sur des champs pertinents. Pour l'éducation, par exemple, il s'agirait de mettre en œuvre les *smart contracts* dans un processus de certification des compétences des étudiants et pour le domaine de la santé, le *smart contract* serait convaincant dans la mise en œuvre d'un registre certifié et distribué notamment sur les volontés relatives à la fin de vie. D'autre part, ces *social smart contracts* (démocratie, éducation, santé et possiblement d'autres) soulèvent la question de la protection des données à caractère personnel, liée à la « pseudonymisation » des informations personnelles sur la *blockchain* et non à leur « anonymisation ». Enfin, ils impliquent de lier le présent projet de

recherche à d'autres actuellement en cours (par exemple #Blockchain4Edu, qui comprend des membres communs) pour mettre en synergie les réflexions engagées.

Groupe de travail 4. Sur l'encadrement juridique des smart contracts

Ce groupe de travail est composé de :

- **Amélie FAVREAU (coordinatrice), MCF, Université Grenoble-Alpes (Propriété intellectuelle et nouvelles technologies)**
- Maître Élise GUILHAUDIS, Avocate inscrite au barreau de Grenoble et Gérante de la Legaltech NUMETIK Avocats,

Une fois les clauses déterminées à figurer sur la plateforme, il sera nécessaire de mener une réflexion sur la « contractualisation du *smart contract* », en encadrant sa mise en œuvre et son fonctionnement sans freiner son essor. Les parties à ce contrat devront être déterminées (informaticiens, avocats, utilisateurs du *smart contract*, etc.) ainsi que les clauses qui assureront cette protection contractuelle (relatives à la force majeure, des clauses d'imprévision ou limitatives de responsabilité, clauses pénales, attributives de compétence ou encore celles relatives au règlement amiable des litiges, etc.). Seront aussi déterminantes les conditions générales d'utilisation de la plateforme sur laquelle seront déposés les *smart contracts*.

Annexe 3. CONTRAT DE LICENCE DE LOGICIEL LIBRE CeCILL-B

English version : http://www.cecill.info/licences/Licence_CeCILL-B_V1-en.txt

Avertissement

Ce contrat est une licence de logiciel libre issue d'une concertation entre ses auteurs afin que le respect de deux grands principes préside à sa rédaction :

* d'une part, le respect des principes de diffusion des logiciels libres ; accès au code source, droits étendus conférés aux utilisateurs ;
* d'autre part, la désignation d'un droit applicable, le droit français, auquel elle est conforme, tant au regard du droit de la responsabilité civile que du droit de la propriété intellectuelle et de la protection qu'il offre aux auteurs et titulaires des droits patrimoniaux sur un logiciel.

Les auteurs de la licence CeCILL-B (pour Ce[a] C[nrs] I[nria] L[ogiciel] L[ibre]) sont :

Commissariat à l'énergie atomique - CEA, établissement public de recherche à caractère scientifique, technique et industriel, dont le siège est situé 25 rue Leblanc, immeuble Le Ponant D, 75015 Paris.

Centre national de la recherche scientifique - CNRS, établissement public à caractère scientifique et technologique, dont le siège est situé 3 rue Michel-Ange, 75794 Paris cedex 16.

Institut national de recherche en informatique et en automatique - INRIA, établissement public à caractère scientifique et technologique, dont le siège est situé Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay cedex.

Préambule

Ce contrat est une licence de logiciel libre dont l'objectif est de conférer aux utilisateurs une très large liberté de modification et de redistribution du logiciel régi par cette licence.

L'exercice de cette liberté est assorti d'une obligation forte de citation à la charge de ceux qui distribueraient un logiciel incorporant un logiciel régi par la présente licence afin d'assurer que les contributions de tous soient correctement identifiées et reconnues.

L'accessibilité au code source et les droits de copie, de modification et de redistribution qui découlent de ce contrat ont pour contrepartie de n'offrir aux utilisateurs qu'une garantie limitée et de ne faire peser sur l'auteur du logiciel, le titulaire des droits patrimoniaux et les concédants successifs qu'une responsabilité restreinte.

À cet égard l'attention de l'utilisateur est attirée sur les risques associés au chargement, à l'utilisation, à la modification et/ou au développement et à la reproduction du logiciel par l'utilisateur étant donné sa spécificité de logiciel libre, qui peut le rendre complexe à manipuler et qui le réserve donc à des développeurs ou des professionnels avertis possédant des connaissances informatiques approfondies. Les utilisateurs sont donc invités à charger et tester l'adéquation du logiciel à leurs besoins dans des conditions permettant d'assurer la sécurité de leurs systèmes et/ou de leurs données et, plus généralement, à l'utiliser et l'exploiter dans les mêmes conditions de sécurité. Ce contrat peut être reproduit et diffusé librement, sous réserve de le conserver en l'état, sans ajout ni suppression de clauses.

Ce contrat est susceptible de s'appliquer à tout logiciel dont le titulaire des droits patrimoniaux décide de soumettre l'exploitation aux dispositions qu'il contient.

Article 1 - DÉFINITIONS

Dans ce contrat, les termes suivants, lorsqu'ils seront écrits avec une lettre capitale, auront la signification suivante:

Contrat : désigne le présent contrat de licence, ses éventuelles versions postérieures et annexes.

Logiciel : désigne le logiciel sous sa forme de Code Objet et/ou de Code Source et le cas échéant sa documentation, dans leur état au moment de l'acceptation du Contrat par le Licencié.

Logiciel Initial : désigne le Logiciel sous sa forme de Code Source et éventuellement de Code Objet et le cas échéant sa documentation, dans leur état au moment de leur première diffusion sous les termes du Contrat.

Logiciel Modifié : désigne le Logiciel modifié par au moins une Contribution.

Code Source : désigne l'ensemble des instructions et des lignes de programme du Logiciel et auquel l'accès est nécessaire en vue de modifier le Logiciel.

Code Objet : désigne les fichiers binaires issus de la compilation du Code Source.

Titulaire : désigne le ou les détenteurs des droits patrimoniaux d'auteur sur le Logiciel Initial.

Licencié : désigne le ou les utilisateurs du Logiciel ayant accepté le Contrat.

Contributeur : désigne le Licencié auteur d'au moins une Contribution.

Concédant : désigne le Titulaire ou toute personne physique ou morale distribuant le Logiciel sous le Contrat.

Contribution : désigne l'ensemble des modifications, corrections, traductions, adaptations et/ou nouvelles fonctionnalités intégrées dans le Logiciel par tout Contributeur, ainsi que tout Module Interne.

Module : désigne un ensemble de fichiers sources y compris leur documentation qui permet de réaliser des fonctionnalités ou services supplémentaires à ceux fournis par le Logiciel.

Module Externe : désigne tout Module, non dérivé du Logiciel, tel que ce Module et le Logiciel s'exécutent dans des espaces d'adressage différents, l'un appelant l'autre au moment de leur exécution.

Module Interne : désigne tout Module lié au Logiciel de telle sorte qu'ils s'exécutent dans le même espace d'adressage.

Parties : désigne collectivement le Licencié et le Concédant.

Ces termes s'entendent au singulier comme au pluriel.

Article 2 - OBJET

Le Contrat a pour objet la concession par le Concédant au Licencié d'une licence non exclusive, cessible et mondiale du Logiciel telle que définie ci-après à l'article 5 pour toute la durée de protection des droits portant sur ce Logiciel.

Article 3 - ACCEPTATION

3.1 L'acceptation par le Licencié des termes du Contrat est réputée acquise du fait du premier des faits suivants :

* (i) le chargement du Logiciel par tout moyen notamment par téléchargement à partir d'un serveur distant ou par chargement à partir d'un support physique ;

* (ii) le premier exercice par le Licencié de l'un quelconque des droits concédés par le Contrat.3.2 Un exemplaire du Contrat, contenant notamment un avertissement relatif aux spécificités du Logiciel, à la restriction de garantie et à la limitation à un usage par des utilisateurs expérimentés a été mis à disposition du Licencié préalablement à son acceptation telle que définie à l'article 3.1 ci dessus et le Licencié reconnaît en avoir pris connaissance.

Article 4 - ENTRÉE EN VIGUEUR ET DURÉE

4.1 ENTRÉE EN VIGUEUR

Le Contrat entre en vigueur à la date de son acceptation par le Licencié telle que définie en 3.1.

4.2 DURÉE

Le Contrat produira ses effets pendant toute la durée légale de protection des droits patrimoniaux portant sur le Logiciel.

Article 5 - ÉTENDUE DES DROITS CONCÉDÉS

Le Concédant concède au Licencié, qui accepte, les droits suivants sur le Logiciel pour toutes destinations et pour la durée du Contrat dans les conditions ci-après détaillées.

Par ailleurs, si le Concédant détient ou venait à détenir un ou plusieurs brevets d'invention protégeant tout ou partie des fonctionnalités du Logiciel ou de ses composants, il s'engage à ne pas opposer les éventuels droits conférés par ces brevets aux Licenciés successifs qui utiliseraient, exploiteraient ou modifieraient le Logiciel. En cas de cession de ces brevets, le Concédant s'engage à faire reprendre les obligations du présent alinéa aux cessionnaires.

5.1 DROIT D'UTILISATION

Le Licencié est autorisé à utiliser le Logiciel, sans restriction quant aux domaines d'application, étant ci-après précisé que cela comporte :

1. la reproduction permanente ou provisoire du Logiciel en tout ou partie par tout moyen et sous toute forme.
2. le chargement, l'affichage, l'exécution, ou le stockage du Logiciel sur tout support.
3. la possibilité d'en observer, d'en étudier, ou d'en tester le fonctionnement afin de déterminer les idées et principes qui sont à la base de n'importe quel élément de ce Logiciel ; et ceci, lorsque le Licencié effectue toute opération de chargement, d'affichage, d'exécution, de transmission ou de stockage du Logiciel qu'il est en droit d'effectuer en vertu du Contrat.

5.2 DROIT D'APPORTER DES CONTRIBUTIONS

Le droit d'apporter des Contributions comporte le droit de traduire, d'adapter, d'arranger ou d'apporter toute autre modification au Logiciel et le droit de reproduire le Logiciel en résultant.

Le Licencié est autorisé à apporter toute Contribution au Logiciel sous réserve de mentionner, de façon explicite, son nom en tant qu'auteur de cette Contribution et la date de création de celle-ci.

5.3 DROIT DE DISTRIBUTION

Le droit de distribution comporte notamment le droit de diffuser, de transmettre et de communiquer le Logiciel au public sur tout support et par tout moyen ainsi que le droit de mettre sur le marché à titre onéreux ou gratuit, un ou des exemplaires du Logiciel par tout procédé.

Le Licencié est autorisé à distribuer des copies du Logiciel, modifié ou non, à des tiers dans les conditions ci-après détaillées.

5.3.1 DISTRIBUTION DU LOGICIEL SANS MODIFICATION

Le Licencié est autorisé à distribuer des copies conformes du Logiciel, sous forme de Code Source ou de Code Objet, à condition que cette distribution respecte les dispositions du Contrat dans leur totalité et soit accompagnée :

1. d'un exemplaire du Contrat,
2. d'un avertissement relatif à la restriction de garantie et de responsabilité du Concédant telle que prévue aux articles 8 et 9,

et que, dans le cas où seul le Code Objet du Logiciel est redistribué, le Licencié permette un accès effectif au Code Source complet du Logiciel pendant au moins toute la durée de sa distribution du Logiciel, étant entendu que le coût additionnel d'acquisition du Code Source ne devra pas excéder le simple coût de transfert des données.

5.3.2 DISTRIBUTION DU LOGICIEL MODIFIÉ

Lorsque le Licencié apporte une Contribution au Logiciel, le Logiciel Modifié peut être distribué sous un contrat de licence autre que le présent Contrat sous réserve du respect des dispositions de l'article 5.3.4.

5.3.3 DISTRIBUTION DES MODULES EXTERNES

Lorsque le Licencié a développé un Module Externe les conditions du Contrat ne s'appliquent pas à ce Module Externe, qui peut être distribué sous un contrat de licence différent.

5.3.4 CITATIONS

Le Licencié qui distribue un Logiciel Modifié s'engage expressément :

1. à indiquer dans sa documentation qu'il a été réalisé à partir du Logiciel régi par le Contrat, en reproduisant les mentions de propriété intellectuelle du Logiciel,
2. à faire en sorte que l'utilisation du Logiciel, ses mentions de propriété intellectuelle et le fait qu'il est régi par le Contrat soient indiqués dans un texte facilement accessible depuis l'interface du Logiciel Modifié,

3. à mentionner, sur un site Web librement accessible décrivant le Logiciel Modifié, et pendant au moins toute la durée de sa distribution, qu'il a été réalisé à partir du Logiciel régi par le Contrat, en reproduisant les mentions de propriété intellectuelle du Logiciel,
4. lorsqu'il le distribue à un tiers susceptible de distribuer lui-même un Logiciel Modifié, sans avoir à en distribuer le code source, à faire ses meilleurs efforts pour que les obligations du présent article 5.3.4 soient reprises par le dit tiers.

Lorsque le Logiciel modifié ou non est distribué avec un Module Externe qui a été conçu pour l'utiliser, le Licencié doit soumettre le dit Module Externe aux obligations précédentes.

5.3.5 COMPATIBILITÉ AVEC LES LICENCES CeCILL et CeCILL-C

Lorsqu'un Logiciel Modifié contient une Contribution soumise au contrat de licence CeCILL, les stipulations prévues à l'article 5.3.4 sont facultatives.

Un Logiciel Modifié peut être distribué sous le contrat de licence CeCILL-C. Les stipulations prévues à l'article 5.3.4 sont alors facultatives.

Article 6 - PROPRIÉTÉ INTELLECTUELLE

6.1 SUR LE LOGICIEL INITIAL

Le Titulaire est détenteur des droits patrimoniaux sur le Logiciel Initial. Toute utilisation du Logiciel Initial est soumise au respect des conditions dans lesquelles le Titulaire a choisi de diffuser son oeuvre et nul autre n'a la faculté de modifier les conditions de diffusion de ce Logiciel Initial.

Le Titulaire s'engage à ce que le Logiciel Initial reste au moins régi par le Contrat et ce, pour la durée visée à l'article 4.2.

6.2 SUR LES CONTRIBUTIONS

Le Licencié qui a développé une Contribution est titulaire sur celle-ci des droits de propriété intellectuelle dans les conditions définies par la législation applicable.

6.3 SUR LES MODULES EXTERNES

Le Licencié qui a développé un Module Externe est titulaire sur celui-ci des droits de propriété intellectuelle dans les conditions définies par la législation applicable et reste libre du choix du contrat régissant sa diffusion.

6.4 DISPOSITIONS COMMUNES

Le Licencié s'engage expressément :

1. à ne pas supprimer ou modifier de quelque manière que ce soit les mentions de propriété intellectuelle apposées sur le Logiciel ;
2. à reproduire à l'identique lesdites mentions de propriété intellectuelle sur les copies du Logiciel modifié ou non.

Le Licencié s'engage à ne pas porter atteinte, directement ou indirectement, aux droits de propriété intellectuelle du Titulaire et/ou des Contributeurs sur le Logiciel et à prendre, le cas échéant, à l'égard de son personnel toutes les mesures nécessaires pour assurer le respect des dits droits de propriété intellectuelle du Titulaire et/ou des Contributeurs.

Article 7 - SERVICES ASSOCIES

7.1 Le Contrat n'oblige en aucun cas le Concédant à la réalisation de prestations d'assistance technique ou de maintenance du Logiciel.

Cependant le Concédant reste libre de proposer ce type de services. Les termes et conditions d'une telle assistance technique et/ou d'une telle maintenance seront alors déterminés dans un acte séparé. Ces actes de maintenance et/ou assistance technique n'engageront que la seule responsabilité du Concédant qui les propose.

7.2 De même, tout Concédant est libre de proposer, sous sa seule responsabilité, à ses licenciés une garantie, qui n'engagera que lui, lors de la redistribution du Logiciel et/ou du Logiciel Modifié et ce, dans les conditions qu'il souhaite. Cette garantie et les modalités financières de son application feront l'objet d'un acte séparé entre le Concédant et le Licencié.

Article 8 - RESPONSABILITÉ

8.1 Sous réserve des dispositions de l'article 8.2, le Licencié a la faculté, sous réserve de prouver la faute du Concédant concerné, de solliciter la réparation du préjudice direct qu'il subirait du fait du Logiciel et dont il apportera la preuve.

8.2 La responsabilité du Concédant est limitée aux engagements pris en application du Contrat et ne saurait être engagée en raison notamment : (i) des dommages dus à l'inexécution, totale ou partielle, de ses obligations par le Licencié, (ii) des dommages directs ou indirects découlant de l'utilisation ou des performances du Logiciel subis par le Licencié et (iii) plus généralement d'un quelconque dommage indirect. En particulier, les Parties conviennent expressément que tout préjudice financier ou commercial (par exemple perte de données, perte de bénéfices, perte d'exploitation, perte de clientèle ou de commandes, manque à gagner, trouble commercial quelconque) ou toute action dirigée contre le Licencié par un tiers, constitue un dommage indirect et n'ouvre pas droit à réparation par le Concédant.

Article 9 - GARANTIE

9.1 Le Licencié reconnaît que l'état actuel des connaissances scientifiques et techniques au moment de la mise en circulation du Logiciel ne permet pas d'en tester et d'en vérifier toutes les utilisations ni de détecter l'existence d'éventuels défauts. L'attention du Licencié a été attirée sur ce point sur les risques associés au chargement, à l'utilisation, la modification et/ou au développement et à la reproduction du Logiciel qui sont réservés à des utilisateurs avertis.

Il relève de la responsabilité du Licencié de contrôler, par tous moyens, l'adéquation du produit à ses besoins, son bon fonctionnement et de s'assurer qu'il ne causera pas de dommages aux personnes et aux biens.

9.2 Le Concédant déclare de bonne foi être en droit de concéder l'ensemble des droits attachés au Logiciel (comprenant notamment les droits visés à l'article 5).

9.3 Le Licencié reconnaît que le Logiciel est fourni "en l'état" par le Concédant sans autre garantie, expresse ou tacite, que celle prévue à l'article 9.2 et notamment sans aucune garantie sur sa valeur commerciale, son caractère sécurisé, innovant ou pertinent.

En particulier, le Concédant ne garantit pas que le Logiciel est exempt d'erreur, qu'il fonctionnera sans interruption, qu'il sera compatible avec l'équipement du Licencié et sa configuration logicielle ni qu'il remplira les besoins du Licencié.

9.4 Le Concédant ne garantit pas, de manière expresse ou tacite, que le Logiciel ne porte pas atteinte à un quelconque droit de propriété intellectuelle d'un tiers portant sur un brevet, un logiciel ou sur tout autre droit de propriété. Ainsi, le Concédant exclut toute garantie au profit du Licencié contre les actions en contrefaçon qui pourraient être diligentées au titre de l'utilisation, de la modification, et de la redistribution du Logiciel. Néanmoins, si de telles actions sont exercées contre le Licencié, le Concédant lui apportera son aide technique et juridique pour sa défense. Cette aide technique et juridique est déterminée au cas par cas entre le Concédant concerné et le Licencié dans le cadre d'un protocole d'accord. Le Concédant dégage toute responsabilité quant à l'utilisation de la dénomination du Logiciel par le Licencié. Aucune garantie n'est apportée quant à l'existence de droits antérieurs sur le nom du Logiciel et sur l'existence d'une marque.

Article 10 - RÉSILIATION

10.1 En cas de manquement par le Licencié aux obligations mises à sa charge par le Contrat, le Concédant pourra résilier de plein droit le Contrat trente (30) jours après notification adressée au Licencié et restée sans effet.

10.2 Le Licencié dont le Contrat est résilié n'est plus autorisé à utiliser, modifier ou distribuer le Logiciel. Cependant, toutes les licences qu'il aura concédées antérieurement à la résiliation du Contrat resteront valides sous réserve qu'elles aient été effectuées en conformité avec le Contrat.

Article 11 - DISPOSITIONS DIVERSES

11.1 CAUSE EXTÉRIEURE

Aucune des Parties ne sera responsable d'un retard ou d'une défaillance d'exécution du Contrat qui serait dû à un cas de force majeure, un cas fortuit ou une cause extérieure, telle que, notamment, le mauvais fonctionnement ou les interruptions du réseau électrique ou de télécommunication, la paralysie du réseau liée à une attaque informatique, l'intervention des

autorités gouvernementales, les catastrophes naturelles, les dégâts des eaux, les tremblements de terre, le feu, les explosions, les grèves et les conflits sociaux, l'état de guerre...

11.2 Le fait, par l'une ou l'autre des Parties, d'omettre en une ou plusieurs occasions de se prévaloir d'une ou plusieurs dispositions du Contrat, ne pourra en aucun cas impliquer renonciation par la Partie intéressée à s'en prévaloir ultérieurement.

11.3 Le Contrat annule et remplace toute convention antérieure, écrite ou orale, entre les Parties sur le même objet et constitue l'accord entier entre les Parties sur cet objet. Aucune addition ou modification aux termes du Contrat n'aura d'effet à l'égard des Parties à moins d'être faite par écrit et signée par leurs représentants dûment habilités.

11.4 Dans l'hypothèse où une ou plusieurs des dispositions du Contrat s'avèrerait contraire à une loi ou à un texte applicable, existants ou futurs, cette loi ou ce texte prévaudrait, et les Parties feraient les amendements nécessaires pour se conformer à cette loi ou à ce texte. Toutes les autres dispositions resteront en vigueur. De même, la nullité, pour quelque raison que ce soit, d'une des dispositions du Contrat ne saurait entraîner la nullité de l'ensemble du Contrat.

11.5 LANGUE

Le Contrat est rédigé en langue française et en langue anglaise, ces deux versions faisant également foi.

Article 12 - NOUVELLES VERSIONS DU CONTRAT

12.1 Toute personne est autorisée à copier et distribuer des copies de ce Contrat.

12.2 Afin d'en préserver la cohérence, le texte du Contrat est protégé et ne peut être modifié que par les auteurs de la licence, lesquels se réservent le droit de publier périodiquement des mises à jour ou de nouvelles versions du Contrat, qui posséderont chacune un numéro distinct. Ces versions ultérieures seront susceptibles de prendre en compte de nouvelles problématiques rencontrées par les logiciels libres.

12.3 Tout Logiciel diffusé sous une version donnée du Contrat ne pourra faire l'objet d'une diffusion ultérieure que sous la même version du Contrat ou une version postérieure.

Article 13 - LOI APPLICABLE ET COMPÉTENCE TERRITORIALE

13.1 Le Contrat est régi par la loi française. Les Parties conviennent de tenter de régler à l'amiable les différends ou litiges qui viendraient à se produire par suite ou à l'occasion du Contrat.

13.2 À défaut d'accord amiable dans un délai de deux (2) mois à compter de leur survenance et sauf situation relevant d'une procédure d'urgence, les différends ou litiges seront portés par la Partie la plus diligente devant les Tribunaux compétents de Paris.

Version 1.0 du 2006-09-05.

Lexique :

API (*Application Programming Interface*) :

Une interface de programmation d'application formant un ensemble normalisé de classes, méthodes, fonctions et constantes qui sert de façade pat laquelle un logiciel offre des services à d'autres logiciels.

Attaque des 51 % :

Attaque qui cible les *blockchains Proof of Work* ou *Proof of Stake*. Cette attaque est réalisée par une entité ou une organisation qui parvient à obtenir la majorité du taux de hachage du réseau, entrainant la perturbation de celui-ci. Le but de cette attaque est de bloquer les validations des transactions ou bien de créer une double dépense, c'est-à-dire réitérer une transaction et ainsi modifier l'historique de la *blockchain*.

***Blockchain* :**

Technologie de stockage et de transmission d'information décentralisée et sécurisée. Il s'agit d'une suite de transactions organisées en blocs qui relie le bloc « genèse » au bloc le plus récent formant ainsi une chaîne dont les données sont vérifiées à intervalles de temps réguliers. Cette chaîne constitue le registre de la *blockchain* dont les données ne peuvent être modifiées.
Son utilité s'observe au travers de sa capacité à maintenir en mémoire tous les échanges effectués entre les utilisateurs depuis sa création grâce à son système d'enregistrement en continu des données produites.

Contrat d'*escrow* :

Contrat de séquestre informatique consistant à confier à un tiers des éléments essentiels (tels que les codes sources, bases de données, etc.) destinés à la réalisation du contrat.

***Delegate Proof of Stake* (DPoS) :**

Mécanisme de consensus doté d'un système de vote par lequel les utilisateurs de la plateforme votent pour élire des représentants chargés de valider les transactions sur la *blockchain*. Ce

système permet d'atteindre un équilibre entre les exigences de sécurité, décentralisation et de scalabilité

GitHub :

Service/Plateforme web d'hébergement et de gestion de développement de logiciels ainsi que de partage open sources de codes.

Jeton ou *Token* :

Représentation numérique d'un actif ou d'un droit qui peut être transférable entre deux parties sur une *blockchain*.

Langage *Domaine* spécifique :

Langage de programmation dont les spécifications sont conçues pour répondre aux contraintes d'un domaine d'application précis.

Licence BSD (Berkeley Software Distribution License) :

Licence libre utilisée pour la distribution de logiciels. Cette licence permet de réutiliser tout ou partie du logiciel pour lequel elle est accordée sans restriction, qu'il soit intégré dans un logiciel libre ou propriétaire.

Licence CeCILL :

Licence de logiciel libre issue d'une concertation entre ses auteurs (CEA, CNRS, INRIA).

Licence GNU/GPL :

Licence publique générale qui fixe les conditions légales de distribution d'un logiciel libre du projet GNU. Cette licence a depuis été adoptée, en tant que document définissant le mode d'utilisation, donc d'usage et de diffusion, par de nombreux auteurs de logiciels libres, en dehors des projets GNU.

Lightning Network (LN) :

Réseau décentralisé construit comme une application de deuxième couche adossée à la *blockchain* Bitcoin. Ce réseau permet un paiement de pair-à-pair instantané et promet ainsi de résoudre les problèmes de scalabilité de la *blockchain* Bitcoin.

***Liquid proof of stake* :**

Un mécanisme de consensus qui permet aux détenteurs de jetons de prêter leurs droits de validation des transactions à d'autres utilisateurs sans renoncer à la propriété de leurs jetons.

Minage :

Opération qui consiste à valider des blocs sur une *blockchain Proof of Work* en contrepartie de récompenses. Ce procédé permet la sécurisation de la *blockchain*.

Mineurs :

Personne vérifiant les transactions et opérations effectuées par les utilisateurs sur un réseau. Il les inscrit ensuite sur la *blockchain* (registre public).

Module Python :

Programme Python qui contient des définitions et des instructions formant ainsi un programme que l'on est amené à réutiliser souvent.

Nœud :

Un élément du réseau de pair-à-pair qui coopère au fonctionnement du système de *blockchain* en validant l'accrochage des nouveaux blocs à la chaîne. Il s'agit de matériel informatique connecté au réseau, ordinateur, groupe d'ordinateurs, serveurs, téléphones, équipés de fonctionnalités lui offrant une capacité de calcul et de mémorisation spécifique pour opérer cette validation.

Oracle :

Tiers de confiance capable d'analyser le comportement observé du programme exécuté, et de décider si oui ou non une défaillance est présente.

« *Pairing* » :

Arrangement informel entre le gouvernement et les partis d'opposition par lequel un membre du parlement accepte ou est désigné pour s'abstenir de voter à la chambre ou bien est prié d'en être absent pendant que l'autre parti a aussi besoin qu'un membre soit absent de la chambre pour d'autres engagements, des raisons de maladie ou de voyage.

Programmation fonctionnelle :

Programmation de type déclaratif qui considère le calcul en tant qu'évaluation de fonctions mathématiques. Le langage fonctionnelle dont la syntaxe et les caractéristiques encouragent la programmation fonctionnelle met en avant l'application de fonctions.

Programmation orientée objet (POO) :

Programmation qui consiste en la définition et l'interaction de briques logicielles appelées objets. Un objet représente un concept, une idée ou toute entité du monde physique. Il possède une structure interne et un comportement, et il sait interagir avec ses pairs. Les langages de programmation orientés objet permettent donc de retranscrire les éléments du réel sous forme virtuelle.

***Proof of Stake* (PoS) :**

Un mécanisme de consensus constituant une alternative à la *Proof of Work* dans lequel la personne qui valide les transactions sur la *blockchain* doit fournir une garantie afin d'être sélectionnée pour valider une transaction. Cela signifie qu'il doit garantir un dépôt (ou une mise) qui est à risque s'il agit de manière malveillante. Ainsi, si le validateur agit de manière malveillante, sa mise est perdue. Ce mécanisme de consensus améliore la décentralisation, la sécurité et la scalabilité sur de la *blockchain*.

***Proof of Work* (PoW) :**

Un algorithme de consensus utilisé pour sécuriser les transactions sur la *blockchain* et éviter les doubles transactions. Avec le mécanisme de *PoW*, les mineurs sont en concurrence pour effectuer des transactions sur le réseau en échange d'une récompense pour leur vitesse et leur précision.

Python :

Langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

***Smart contract* :**

Protocole informatique qui s'occupe de la vérification, la négociation ou l'exécution d'un contrat. Ces protocoles sont régis par un code informatique enregistré dans une *blockchain* dont l'exécution est déclenchée par une transaction et dont le résultat est soumis au consensus des nœuds validateurs avant enregistrement dans un nouveau bloc. Ce système permet une exécution automatique des contrats et ainsi d'assurer leur force obligatoire, empêchant toute falsification.

Source ganache.cli :

La version en ligne de la commande Ganache. Elle fait partie de la suite d'outil de développement Ethereum. Cette commande permet de simuler le comportement complet d'un client et ainsi de rendre le développement d'applications Ethereum plus rapide, plus facile et plus sûr.

Source brownie :

Environnement de développement et de test de programme basé sur la *blockchain* Ethereum.

***Time-stamping* :**

Mécanisme, aussi appelé horodatage, qui associe une date et une heure à une donnée informatique ou un événement. Permet de préciser l'instant où une opération a été effectuée.

***Turing complete* :**

Un système informatique formel est dit complet au sens de Turing ou *Turing-complete* s'il possède un pouvoir expressif au moins équivalent à celui des machines de Turing.

Dictionnaire des acronymes

API : *Application Programming Interface* (interface de programmation d'application)

CGU : Conditions générales d'utilisation

CNIL : Commission nationale de l'informatique et des libertés

DINSIC : Direction interministérielle du numérique et du système d'information et de communication de l'État

PoW : *Proof of Work*

PoS : *Proof of Stake*

RGPD : Règlement général sur la protection des données

SNE : Syndicat national de l'édition

Références

Sites internet :

Binance Academy : https://academy.binance.com/fr

Bybit Learn : https://learn.bybitglobal.com

CeCILL, licence française de logiciel libre : https://cecill.info/licences.fr.html

Cryptonaute : https://cryptonaute.fr

Economie.gouv.fr : https://www.economie.gouv.fr

Generix group : https://www.generixgroup.com

Github : https://github.com

Journal du Net : https://www.journaldunet.fr

Vie Publique : https://www.vie-publique.fr

Wikipedia : https://fr.wikipedia.org

Rapports :

Sophie COUTOR, Christine HENNEBERT, Mourad FAHER, Ministère de l'Intérieur, « Blockchain et identification numérique - Restitution des ateliers du groupe de travail "blockchain et identité" (BCID) » *Rapport Vie Publique*, octobre 2020.

Table des matières :

Le constat de l'impact des nouvelles technologies sur les professions du droit est parfois alarmant. Il est vrai que les changements à venir dans l'exercice du droit par l'automatisation de certaines opérations interrogent. Conscients des risques que peut représenter cette technologie, les membres de l'équipe de recherche ambitionnaient de faire profiter les professionnels du droit des potentialités des smart contracts.

Cette recherche s'est ainsi donnée pour objectif de déposer sur une plateforme en accès ouvert un « clausier », qui répertorie sur plusieurs occurrences la traduction informatique de clauses françaises et anglaises afin de les rendre autoexécutantes sur une blockchain, ce qui a supposé de faire cohabiter un triple langage : juridique, naturel et informatique.

Issue d'une collaboration entre juristes franco-britanniques et informaticiens, chacun détenant un savoir-faire indispensable à cette nouvelle forme d'écriture du droit, cette réalisation s'est opérée en trois étapes. La première consistait à circonscrire les défis et à évaluer les solutions pour les dépasser. Une fois cette étape franchie venait un travail de sélection entre les stipulations contractuelles et les différents langages informatiques. Il s'agissait de faire la part des nombreuses stipulations ne pouvant être traduites en contrat autoexécutant et parmi les autres stipulations de retenir les plus utiles et pertinentes pour les professionnels du droit au sein des deux systèmes juridiques étudiés. La dernière étape a consisté en un indispensable travail d'explication sur la sélection opérée et la mise en œuvre de l'autoexécution, tant technique que juridique, de sorte à favoriser l'accessibilité et l'intelligibilité du clausier trilingue et en vue de contribuer à l'élaboration d'un cadre en droit et en informatique pour l'essor des *smart contracts*.

Site dédié à la recherche : https://smart-contracts.univ-grenoble-alpes.fr/

Plateforme dédiée à la librairie de smart contracts : https://opensmartcontract-front.netlify.app/

Amélie Favreau, maîtresse de conférences HDR à l'université Grenoble Alpes

UGA Université Grenoble Alpes
CRJA Université Grenoble Alpes
Institut des Études et de la Recherche sur le Droit et la Justice