Les Smart Contracts, les nouveaux contrats augmentés ?

Benjamin Jean, Primavera de Filippi

"Les Smart Contracts, les nouveaux contrats augmentés ?"

Conseils & Entreprises, la revue de l'ACE. Issue n.24

Benjamin Jean & Primavera De Filippi

Aussi importante que soit l'attention apportée à sa phase de conception, l'équilibre entier d'un contrat peut être remis en cause par la mauvaise exécution (volontaire ou par négligence) de ses clauses si consciencieusement négociées. Ce risque est proportionnel à la souplesse susceptible d'être mise en œuvre pour contracter (les contrats-cadres figurent certainement parmi les moins bien lotis) et est accru lorsque l'équilibre initial est déjà fragile (en matière de licences associées à l'usage d'une technologie, l'exécution diffère parfois tellement du cadre contractuel prévu qu'on se demande souvent si le contrat réel n'est pas dans une zone grise à mi-chemin entre ce qui a été écrit et l'usage constaté).

Avec la technologie Blockchain et les Smart Contracts, une réponse technique semble se dessiner pour réconcilier les parties au contrat, leurs conseils qui les assistent et ceux qui assurent la mise en oeuvre de ces engagements.

Particulièrement développés sur Ethereum[1], les Smart Contracts sont des logiciels exécutés de manière décentralisée sur la Blockchain dont les fonctionnalités se déclenchent par la réalisation de conditions prédéfinies (susceptibles de trouver leurs origines dans un contrat classique). Bénéficiant ainsi des forces de la blockchain que sont l'inaltérabilité et la vérifiabilité des transactions, ils trouvent tout leur intérêt dans le cadre de relations contractuelles à venir, entre individus et, plus encore, par objets interposés : la voiture connectée de Bruno « contractera » directement avec le téléphone de Joseph pour bénéficier de son abonnement musical ou – peut être – avec la voiture de Sandra qui lui prêtera un surplus d'énergie. Avec la démocratisation de l'IoT (Internet des objets) et puisqu'il faut se rappeler que tous les actes de la vie courante cache des contrats, c'est notre futur tout entier qui pourrait être un gigantesque terrain d'expérimentation du potentiel que présentent la technologie Blockchain et les Smart Contracts en automatisant à l'extrême les relations contractuelles et en créant des produits et services de plus en plus horizontaux.

---

[1] Ethereum est une blockchain publique permettant la création par les utilisateurs de contrats intelligents grâce à un langage Turing-complet. Ces contrats intelligents sont basés sur un protocole informatique permettant de vérifier ou de mettre en application un contrat mutuel, ils sont déployés et consultables publiquement dans la blockchain. Voir https://fr.wikipedia.org/wiki/Ethereum

Relativement technique, leur rédaction impose l'acquisition de nouvelles compétences que les juristes vont devoir acquérir s'ils souhaitent tirer profit de cette opportunité (ou a minima la comprendre). Cette compétence sera ensuite soit gérée directement (le Smart Contract pourrait être versé en annexe au contrat), soit obtenue en s'associant des tiers capables de traduire techniquement la volonté exprimée juridiquement par les conseils des parties. Interprofessionnalité, quand tu nous tiens...

Il s'agit ainsi d'outils qui peuvent se superposer au contrat tel un clone numérique, n'en assurer qu'une implémentation partielle ou encore s'étendre bien au-delà des provisions de la licence[2]. Dans toutes ces hypothèses, les Smart Contract ne remplacent pas les contrats, mais les renforcent. À ce titre, l'acceptation du contrat -- qu'il soit implicite ou explicite -- est un préalable légal nécessaire à son exécution et il est nécessaire de prévoir un processus par lequel le consentement ne puisse pas être remis en cause (les Smart Contrat opérant néanmoins de manière automatique, la sanction en cas d'absence de consentement serait classique et passerait par la réparation du préjudice subi). La question du consentement ne sera pas traitée dans cet article, étant néanmoins entendu qu'il s'agit d'un aspect important de l'adoption des Smart Contracts compte tenu de la nouveauté et de la complexité de l'outil d'une part et de l'utilisation fréquente de Smart Contracts rédigés par des tiers d'autre part.

Très en vogue aujourd'hui, les Smart Contract intéressent autant qu'ils font peur. Notre sentiment est que les Smart Contract sont le parfait complément au contrat, l'« augmentant » au sens qu'ils lui confèrent la force du numérique et des communautés.

I - L'ajout d'une dimension numérique

Le concept de « contrats intelligents » n'a pas attendu le phénomène Blockchain pour apparaître. De nombreux acteurs de la *Legal Tech*[3] sont sur ce marché depuis plusieurs années déjà[4], appliquant les technologies numériques sur les contrats afin de participer à leur rédaction, leur évaluation ou encore le suivi de leur exécution.

Les Smart Contract vont néanmoins plus loin encore, techniquement et culturellement, et il est important pour les juristes d'en saisir l'entière portée pour pleinement tirer profit de la dimension logicielle des Smart Contract.

---

[2] À cet égard, le parallèle est facile à faire avec les Mesures Techniques de Protection (DRM), même si le bénéfice des Smart Contrat est bien plus élevé puisqu'ils s'étendent de la conclusion du contrat jusqu'à son execution.

[3] Est défini comme acteur de la LegalTech toute organisation qui propose, fournit et/ou développe des technologies au service du Droit ou de l'accès à la Justice.

[4] Marché en perpétuel croissance, notamment propulsé par l'arrivée de l'Open Data sur le marché du droit. L'écosystème est particulièrement présent au sein de la communauté « Open Law » et disposera de son propre salon professionnel en novembre 2016.

A - Des contrats intelligents

Les technologies numériques ont permis à plusieurs reprises d'accompagner les professionnels et consommateurs dans leur pratique entourant les contrats. Ainsi, si l'introduction du numérique offre en elle-même des atouts considérables (en termes d'archivage, classement, recherche de texte ou de collaboration entre individus) c'est lorsque les logiques de programmation et d'écriture sont combinées que le potentiel se révèle.

La cohabitation des logiques informatiques et juridiques est ainsi assez naturelle même si elle met en présence deux typologies de « codes » qui ont des caractéristiques différentes. Formaliser une relation contractuelle au sein d'un système informatique nécessite ainsi d'avoir une juste compréhension de leurs forces et faiblesses :

- Le « code juridique », écrit en un langage naturel, intrinsèquement souple et ambigu. Il repose sur l'idée que la loi doit toujours être appréciée par un juge afin de déterminer, au cas par cas, comment celle-ci doit être appliqué au cas en question. Dans certains cas, le juge peut même décider d'ignorer les paroles de la loi, lorsque, compte tenu des faits, l'application aveugle des règles de droit irait effectivement à l'encontre de l'intention originale du législateur.
- Le « code informatique », écrit en un langage strict et formel, et s'appliquant uniquement aux cas qui ont été spécifiquement pris en compte. Par opposition à la loi, le code informatique ne possède pas la souplesse nécessaire pour couvrir des situations imprévues qui pourraient émerger dans le futur. D'autant plus que, plus une règle a été formalisée, plus il sera facile pour un attaquant de la contourner.

C'est ce qui explique les difficultés, encore aujourd'hui, qu'il y a à interpréter le langage naturel dans une logique informatique : trop de zone d'interprétations, de recouvrements et parfois de références externes non normalisées viennent embrouiller la lecture que pourrait en faire un logiciel[5]. La tendance a ainsi plutôt été de proposer des langages de programmation relativement simples à prendre en main et suffisamment souples pour répondre aux grandes catégories de contrats que nous connaissons.

Une fois cette passerelle ouverte entre les deux codes, tous les outils et toutes les pratiques utiles au développement peuvent être transposés à ce nouveau territoire : les logiques de test et de qualité logicielles qui permettent de réduire drastiquement les bugs et codes morts, les logiques d'optimisation, d'API, d'assistance au développement[6].

---

[5] Des solutions techniques émergent néanmoins aujourd'hui à base de Big Data et de Machine Learning. Elles restent néanmoins spécifique à un métier et elle ne confère pas les avantages d'un code qui serait nativement interprétable par une machine.

[6] Tel le logiciel Scratch (https://scratch.mit.edu/) qui permet à des enfants de programmer des logiciels simples. Développé par le groupe de recherche Lifelong Kindergarten auprès du laboratoire Média du MIT, Scratch est un langage de programmation destiné à faciliter la création d'histoires

Les Smart Contracts d'Ethereum, dans cette conception d'un « contrat augmenté », reposant sur un langage de programmation adapté à la rédaction de contrats, bénéficiant à ce titre des outils d'assistance au développement, qui soit commun à tous les utilisateurs de la blockchain concernés. La logique de décentralisation sous-jacente et les technologies utilisées favorisent l'interopérabilité entre les différents Smart Contracts, ce qui permet de combiner facilement leur effet pour produire des Smart Contracts de plus en plus complexes.

B - Des contrats autonomes

Une fois rédigés et mis à disposition sous le format idoine, les Smart Contracts seront accessibles en tout point de la Blockchain. Ils vont ensuite « vivre leur propre vie » : de nouvelles relations contractuelles seront générées à chaque nouvelle acceptation de ses termes par un nouvel utilisateur conformément au Smart Contract[7] et cela aussi longtemps que le Smart Contract disposera de « fonds » lui permettant de s'exécuter[8]. Ainsi, le concepteur -- et plus encore l'utilisateur -- d'un Smart Contract devra être vigilant quant aux prérogatives qu'il s'est ménagées dans le contrat : il est parfaitement envisageable qu'il soit lui-même incapable de corriger, débloquer un programme défaillant (voir notamment la faille exploitée sur TheDAO pour récupérer plusieurs millions à l'insu des autres utilisateurs[9]) voire terminer son propre Smart Contract s'il n'a prévu de fonction idoine (dans ce dernier cas le Smart Contract existera toujours).

Il est possible pour toute personne le désirant de s'inspirer d'un Smart Contract existant, voire de le répliquer, pour le publier à son tour avec les mêmes effets. Afin de donner confiance à cette base immatérielle croissante, il est nécessaire de définir pour ces contrats un régime qui permet à tout utilisateur -- commercial ou non -- d'en bénéficier pour y contribuer à son tour. L'Open Source est la solution naturelle.

Sur un tout autre terrain, parce que pouvant être à l'origine de dommages, en particulier financiers, les Smart Contracts sont en théorie des « faits juridiques » susceptibles d'alimenter une action en responsabilité. Ces derniers se heurtent néanmoins à certaines considérations pratiques potentiellement problématiques. D'une part, étant donné la pseudonymité de la plupart des plateformes blockchain, il est difficile d'établir l'identité des personnes responsables d'avoir déployé un smart contrat sur la blockchain. D'autant plus que, même si un "compte" était identifié comme spécifiquement responsable, les smart contrats pouvant être exécutés de manière autonome, ils continueront à opérer indépendamment de ce qu'il arrive à leurs créateurs. Ainsi, même s'il est théoriquement possible d'incriminer les créateurs d'un Smart

---

interactives, de dessins animés, de jeux, de compositions musicales, de simulations numériques et leurs partage sur le Web.

[7] Il est possible pour un Smart Contract de limiter le bénéficiaires potentiels du Smart Contract : seuls ces derniers seront en capacité d'activer ce Smart Contract.

[8] Lors de la publication d'un Smart Contract sur Ethereum, il est nécessaire de lui affecter un certain nombre de crédits. Ce dispositif permet d'alléger Ethereum du poids d'ancien Smart Contracts qui ne seraient plus opérationnels.

[9] Cf infra.

Contract qui est à l'origine d'un tort, cela ne servira pas à arrêter le Smart Contract, qui sera exécuté à chaque fois que quelqu'un effectuera une transaction vers ce contrat. D'autre part, puisque l'exécution des Smart Contracts se fait de manière décentralisée, il n'y a pas d'opérateur central qui peut être jugé responsable de leur exécution. On pourrait dire alors qu'il existe une responsabilité distribuée, entre tous les nœuds actifs du réseau, qui participent tous en partie à perpétuer le tort. Or, d'une manière générale, le caractère décentralisé de la communauté blockchain en fait une nébuleuse difficile à attraire devant les tribunaux[10].

C - Des contrats Open Source

À l'instar d'Internet qui ne s'est développement qu'à partir de protocole Open Source et décentralisé, le succès de la Blockchain repose uniquement sur des technologies libres et Open Source. Le protocole Blockchain d'origine et toutes ses implémentations -- dont Ethereum -- sont eux-mêmes sous licence GNU General Public License v2[11]. Il est même probable que cette dimension juridique de la Blockchain soit l'une des composantes essentielles de son succès -- ce qui amène à penser que tout autre choix aurait grandement limité l'adoption de la technologie et que des projets de Blockchain fermés (on parle généralement de « permissioned blockchain ») mésestiment certainement l'une des caractéristiques non techniques les plus fondamentales (les enjeux de sécurité et d'évaluation par les pairs étant d'autant plus important que les opérations sont stratégiques).

On parle d'Open Source en matière de logiciel lorsqu'une licence Open Source (c'est à dire répondant un certain nombre de critères de non-exclusivité définis par l'*Open Source Initiative*[12]) est apposée sur un code logiciel afin d'en permettre le développement collaboratif et ouvert. Par extension, cette logique a été portée sur tout type de création potentiellement soumise à un quelconque droit de propriété intellectuelle (droit d'auteur, brevet, droit *sui generis* des bases de données, etc.)[13] et s'étend à tous les aspects collaboratifs du projet (et non plus seulement la dimension juridique). Les avantages du modèle sont multiples et les plus évidents sont la transparence (toute personne peut prendre connaissance du code source), la mutualisation (la logique Open Source favorise la réutilisation et le perfectionnement plutôt que le redeveloppement) et la pérennité (le caractère incitatif repose sur le bénéfice partagé qu'assure la licence).

En tant que développement logiciel, les Smart Contracts peuvent être qualifiés d'œuvres soumises aux droits d'auteur dès lors que la contribution de leur concepteur est suffisamment

---

10 Cf infra.

11 La licence GNU General Public License est une licence à réciprocité (le logiciel et toute modification apportée doivent rester soumis à cette même licence) publiée par la *Free Software Foundation* et le *Software Freedom Law Center*. Elle garantit à tous les utilisateurs et contributeurs que le code et ses dérivées resteront soumis à ce même régime inclusif.

12 Voir l'*Open Source Definition* : https://opensource.org/osd-annotated qui s'approche de la Free Software Definition -- qui adopte le point de vue de l'utilisateur.

13 B. Jean, L'évolution des licences libres et open source : critères, finalités et complétude ?, dans « l'Histoire du Libre. Du code échangé aux licences partagées », coll. Framabook, ed. Framasoft 2013

importante pour révéler un apport personnel[14]. C'est une réalité néanmoins encore peu prise en compte par les auteurs de Smart Contracts qui, pourtant majoritairement dans une logique de partage et d'emprunt, n'assurent que rarement un tel cadre contractuel. Cela se justifie par divers facteurs qui passent par l'ignorance, par la volonté délibérée de ne pas se plier à un modèle en place ou encore par l'absence de sentiment d'appartenance sur les Smart Contracts en tant que tels.

Néanmoins, il y a fort à parier que le développement d'Ethereum et des Smart Contracts s'accompagne d'une réelle réflexion sur le sujet et le choix de l'Open Source présente de nombreux avantages :

- en termes de mutualisation. Encore une fois : la pratique existe déjà, il s'agirait ici de lui associer le cadre sous-jacent[15] ;

- en termes de transparence pour s'assurer d'une bonne compréhension du code exécuté. Notons néanmoins que le code source du contrat n'est visible que par ses développeurs, ceux utilisateurs qui contractent accèdent seulement au "bytecode". Enregistré sur la Blockchain (et accessible à tous), c'est ce code qui permettra, à l'image d'un binaire en matière de logiciel[16], son exécution sur Etherium ;

- en termes de qualité : afin de perfectionner les Smart Contracts existants en produire des plus en plus complexes pour répondre à la diversité des usages à venir.

C'est là tout l'intérêt d'initiatives visant à créer et maintenir une bibliothèque Open Source de Smart Contracts ouverte et accessible à tous[17].

---

[14] Au même titre que les contrats eux-mêmes, cette originalité pourrait être remise en cause [Ref]. Par ailleurs, l'auteur du Smart Contract n'étant pas nécessairement le même que celui du contrat, un copropriété pourra potentiellement se dessiner selon la contribution de chacun.

[15] Un parallèle peut être fait avec la communauté Github à sa naissance qui, dans un mouvement qui avait été présenté comme le Post Open Source Software, refusait délibérément de se préoccuper des enjeux de licence et de propriété intellectuelle. Cette "mode" est passée en raison de la difficulté de voire réutiliser le code ainsi produit et l'absence de licence Open Source est un critère notant négativement les projets de la plate-forme

[16] En pratique, le code généré est de type assembleur et permettrait, dans le cadre d'un désassemblage, de reproduire un code source équivalent au code source initial.

[17] Voir notamment les ateliers menés par Open Law, Cellabz et les Bricodeurs autour des Smart Contracts afin de réunir juristes, designers et développeurs sur un temps d'exploration pratique. Des exercices permettront aux participants de concevoir et expérimenter les Smart Contracts sur Ethereum et permettront l'alimentation d'une bibliothèque de Smart Contract Open Source. Figurant dans la liste des communs qui promeut et soutient l'association, ce projet bénéficie à ce titre d'une pérennité accrue et de dispositif particulier tels que le statut de Contributeur rémunéré au commun qui permet l'indemnisation en droit d'auteur de contributeurs au projet qui accepterait la diffusion Open Source de leurs Smart Contracts.

La logique d'Open Source induit de réfléchir immédiatement en termes de puissance communautaire, les millions d'utilisateurs de la Blockchain étant sans aucun doute sa plus grande richesse.

II - L'ajout d'une dimension communautaire

Par son fonctionnement décentralisé et l'emprunt à de nombreuses technologies et pratiques de l'Open Source, la Blockchain est un projet éminemment communautaire qui tire toute sa force et son succès de l'engouement qu'il a su générer.

Les conséquences directes sont la décentralisation et la désintermédiation. Le corollaire direct est la mise en place d'un système d'autorégulation par l'usage et la masse plutôt que par une autorité fédérative. En effet, étant donné qu'il n'y a plus d'opérateur centralisé responsable de monitorer et de réguler les activités des utilisateurs des plateformes blockchain, la communauté d'utilisateurs est la seule capable de remédier à des torts ou de corriger les déviances qui peuvent se rencontrer sur ces plateformes. Ainsi, malgré la forte formalisation et auto-exécution des clauses contractuelles intégrées au sein des Smart Contracts, ces clauses peuvent néanmoins être modifiées ou réinterprétées par la communauté, dès lors qu'il y a un consensus. Cela permet ainsi d'introduire une souplesse nouvelle au sein de ces contrats soi-disant « irrévocables », qui s'éloigne de la logique latine qui admet à peine la modification par le juge. On rejoint ainsi un intérêt renouvelé pour la médiation — bien qu'il s'agisse, cette fois, d'une médiation décentralisée.

A - Des contrats décentralisés

Dans le système financier traditionnel, les intermédiaires financiers ont le pouvoir de modifier ou de renverser unilatéralement les transactions considérées illégitimes. Inversement, sur un réseau blockchain, après qu'une transaction a été effectuée, elle ne peut —théoriquement— pas être renversée. Ainsi, la communauté d'utilisateurs d'une blockchain est liée par l'exécution automatique des Smart Contracts avec lesquels elle a interagi auparavant. Il n'y a pas de mesure conservatoire, pas de retour en arrière si une transaction s'avère être "frauduleuse" ou contestée (à moins que tous les nœuds actifs du réseau en conviennent autrement collectivement).

Évidemment, il est toujours possible pour la communauté de prévoir au sein même du contrat une fonction de sortie, qui peut par exemple reposer sur l'intervention d'un tiers de confiance. C'est le cas par exemple des Smart Contracts qui incorporent leurs propres mécanismes de résolution des conflits, avec des systèmes d'arbitrage fondés sur le jugement subjectif de juges ou d'arbitres privés. Ces systèmes restent cependant complexes à développer, et vont en quelque sorte au travers de la motivation première des Smart Contracts, qui se présentent

comme une alternative « trustless » et décentralisée, visant à combler les déficits des contrats juridiques traditionnels.

Évidemment, le concept même d'un système «trustless» (c'est-à-dire un système qui élimine le besoin de confiance) n'est rien d'autre qu'un idéal. Le fonctionnement de toute blockchain repose sur un certain nombre d'acteurs auxquels il faut faire confiance. Cela nous conduit à une deuxième limitation des Smart Contracts, liée au mode de fonctionnement des systèmes fondés sur les technologies blockchain —dont on peut déjà observer les limites et potentiels effets pervers.

Sur une blockchain, la validation des transactions et l'exécution des Smart Contracts sont effectuées de manière décentralisée. Au lieu de confier à un opérateur centralisé la tâche de vérifier les transactions, l'historique des transactions est détenu —en sa totalité— par tous les membres du réseau, qui participent tous à leur validation et leur vérification (un processus dénommé de « minage »). Et c'est le fait qu'elles soient partagées en réseau qui les rend infalsifiables, car si quelqu'un essaie de modifier même l'une seule de ces transactions, la fraude sera immédiatement détectée par tous les membres du réseau.

Or, étant donné le protocole adopté par la plupart des applications blockchain existantes (Proof-of-Work), il y a cependant le risque que toute personne (ou groupe de personnes) qui contrôle plus de 50% du réseau ait potentiellement la possibilité de censurer des transactions, ou même d'influencer l'exécution de certains Smart Contracts afin que le système lui soit plus favorable (le soi-disant « 51% attack »).

Bien qu'il ne s'agisse d'une possibilité jusqu'à présent que théorique, le problème existe bel et bien en pratique, puisque les plateformes blockchain les plus importantes aujourd'hui —Bitcoin et Ethereum— sont en grande partie contrôlées par un très petit nombre d'acteurs, qui pourraient facilement se coordonner pour contrôler plus du 50% de réseau. Évidemment, la probabilité d'un abus est assez basse puisque c'est dans l'intérêt de la communauté que la confiance dans le système soit conservée (la valeur du système repose sur la confiance qu'il véhicule).

Un risque plus important, en ce qui concerne la perte de l'immutabilité des transactions sur une blockchain, a été récemment mis en œuvre par la communauté d'Ethereum, suite à une décision de la communauté d'intervenir en modifiant le protocole d'Ethereum afin de récupérer les 50 millions de dollars qui avaient été volés lors du hack que connut TheDAO[18]. Malgré les critiques qu'elle a subies, cette intervention, qui a permis de remédier à un tort dû à une faille informatique, constitue une première étape vers l'établissement d'une gouvernance décentralisée dans un réseau blockchain.

[18] TheDAO est un Smart Contract déployé sur la blockchain d'Ethereum, qui se présente comme un fond d'investissement décentralisé. Après avoir levé plus de 150 millions de dollars, quelques semaines seulement après son lancement, TheDAO a été hackée. L'attaquant a exploité une faille dans le code du Smart Contract afin de vider le fond de plus de 50 millions de dollars.

B - Des contrats modérés par une communauté

En matière de projets Open Source, on mesure la force d'un projet à sa communauté.

Plus précisément, il est généralement plus juste de parler des communautés plutôt que d'une communauté Open Source : la diversité des profils, des intérêts et des missions qui leur incombent en font des communautés souvent très distinctes (entre ceux qui conçoivent la technologie, ceux qui la documentent, ceux qui développent en périphérie, ceux qui utilisent et participent à la dissémination de la technologie, etc.). L'action des communautés présentes dans Blockchain est ainsi multiple et passe donc par du développement (de code ou d'usages), la relecture par les pairs, la création de Smart Contracts, etc.

Là où son rôle devient intéressant, c'est lorsque la communauté est définie comme l'organe qui représente la gouvernance du projet (ce dernier n'étant pas nécessairement véhiculé par une association -- et, lorsque c'est le cas, l'association n'étant pas nécessairement en capacité de prendre toutes les décisions).

Nous nous intéressons ainsi aux situations où cette communauté pourra être amenée à se concerter et à prendre des décisions structurantes vis-à-vis des technologies et des règles de fonctionnement qui l'animent. La question est loin d'être anonyme puisqu'elle touche à la confiance que l'on décide d'accorder au code qui « gouverne » cette communauté : l'un des principes fondateurs de ces protocoles et des communautés qui les animent reposent sur le caractère infalsifiable, inviolable et immuable du code, seul à même d'éviter les abus, les manipulations et les situations de conflits d'intérêts.

Concernant le développement même des protocoles sous-jacent à la Blockchain, certaines règles se dessinent aujourd'hui afin de permettre un modèle de gouvernance compatible avec l'organisation horizontale qui caractérise les principales implémentations de la Blockchain.

Ce modèle permet de faire évoluer les technologies et privilégie a priori un code de qualité compte tenu de la concurrence entre les projets et de l'importance de la sécurité parmi les critères distinguant chaque projet.

La question est plus complexe en cas d'apparition de "failles" techniques qui nécessiteraient d'être corrigées rétroactivement, deux attitudes possibles se présentent. Deux écoles se dessinent :

- La première est de considérer que les failles du code doivent être acceptées par la communauté comme ayant « force de loi » au moment du dommage, et le code ne pouvant être corrigé que pour l'avenir. Ici, aucune réparation possible, « l'usage normal » du code n'entrant absolument pas en ligne de compte : ce contrat sans cause ni objet est gouverné uniquement par l'adage « Code is law. »

- L'autre attitude consiste à appliquer a posteriori des remèdes « techniques » afin de limiter au maximum les conséquences de la faille et de son exploitation par le « pirate ». Dès lors, cela signifie que les membres de la communauté sont reliés par une forme de « contrat social » qui dépasse le strict cadre du code et qui se révèlerait lorsque le code est détourné de son usage normal.

Ce qui semblerait être une décision assez simple pour un certain nombre d'acteurs impliqués sur TheDAO (« Faut-il remédier aux dommages causés par une faille logicielle? »), s'est avéré être une question très controversée et c'est toute la croyance dans la fiabilité d'un système supposé éviter les conflits d'intérêts et les logiques d'appropriation qui est ébranlée,

La question a conduit à un véritable schisme au sein de la communauté d'Ethereum, entre ceux qui désiraient intervenir pour renverser la transaction, et ceux qui veulent absolument respecter les paroles du code (en dépit de ses failles), même si cela va à l'encontre à l'intention initiale des développeurs. Ces derniers considèrent en effet que l'attaquant n'a rien fait de mal, il a simplement exploité les fonctionnalités cachées du code afin d'obtenir des fonds supplémentaires. Renverser la transaction pour récupérer ces fonds reviendrait donc à un vol.

Le bon sens -- qui caractérise le droit -- en la matière nous amène à penser que ce n'est pas nécessairement celui qui se voudra plus malin que le système en exploitant ses failles qui obtiendra gain de cause devant les juridictions nationales[19]. Aux yeux du Smart Contract, il n'y a pas de différence entre l'exécution d'une fonctionnalité attendue et l'exploitation d'une faille du contrat et un juge saisi d'un tel cas pourrait parfaitement trancher en faveur de l'un ou de l'autre des raisonnements[20].

On commence alors à saisir la difficulté associée à la notion de Smart Contracts, à savoir : où se situe exactement le contrat ? Dans les relations juridiques traditionnelles sous-jacentes ou dans les lignes de code ? Le Smarts Contrat sont-ils des contrats auto-exécutant ou simplement des voies d'exécution d'un contrat qu'il faudrait chercher ailleurs ? En fin de compte, la question est de savoir si l'*intention du code* doit prévaloir sur les *paroles du code*.

C'est un développement riche qui relance le débat ouvert en 2000 par Lawrence Lessig dans son article *Code is Law – On Liberty in Cyberspace* :

> *Ce n'est pas entre régulation et absence de régulation que nous avons à choisir. Le code régule. Il implémente – ou non – un certain nombre de valeurs. Il garantit certaines libertés, ou les empêche. Il protège la vie privée, ou promeut la surveillance. Des gens décident comment le code va se comporter. Des gens l'écrivent. La question n'est donc*

---

19 Voir à cet égard Matt Levine, Blockchain Company's Smart Contracts Were Dumb, 17 juin 2016 . https://www.bloomberg.com

20 Considérant soit que le hacker qui a tiré profit d'une « faille » est parfaitement dans son droit (à l'inverse de ceux qui ont rétroactivement modifié le contrat pour mieux satisfaire leurs intentions/l'esprit du contrat), soit, interprétant le contexte et l'intention du contrat, que le hack est contraire à l'intention des parties et qu'il convient donc de le corriger afin qu'il reflète au mieux l'intention des parties.

pas de savoir qui décidera de la manière dont le cyberespace est régulé : ce seront les codeurs. La seule question est de savoir si nous aurons collectivement un rôle dans leur choix – et donc dans la manière dont ces valeurs sont garanties – ou si nous laisserons aux codeurs le soin de choisir nos valeurs à notre place[21].

La logique d'immuabilité nécessaire en matière financière et bancaire, qui peut être arithmétique et rationnelle, doit être revue lorsque le dispositif porte atteinte à des humains. Le code doit travailler pour les humains, et être revu si ce n'est plus le cas. Cette régulation est finalement assez commune à celle que l'on retrouve dans toute communauté où l'usage prend une place prédominante[22]. Ajoutons enfin d'une part que cette mécanique a d'autant plus de sens dans le contexte international où le recours à une juge national est détaché des besoins opérationnels et d'autre part que le recours à la communauté est ici un danger limité dès lors que les interactions et échanges se font « à ciel ouvert » et de manière transparente (autant concernant le code du Smart Contract que des décisions qui peuvent le concerner).

Conclusions

La communauté a un rôle important à jouer lorsque la confiance dans la technologie est perdue, en raison de circonstances imprévues dues, par exemple, à une faille ou à une vulnérabilité dans le code. Si une technologie ne fonctionne pas correctement, il est fondamental que l'on puisse intervenir afin de rétablir les garanties originales du système, et idéalement, de restaurer la confiance dans la technologie.

Avec la blockchain, le pouvoir d'intervention s'est déplacé des autorités centralisées (tels que les opérateurs en lignes ou les autorités judiciaires) vers une communauté décentralisée de pairs, qui ont la possibilité de se coordonner afin de changer les règles du jeu.

Mais le pouvoir ne vient pas sans responsabilités. Les membres d'une communauté blockchain ont le pouvoir de modifier l'état d'une blockchain, et sont responsables de la façon dont ils choisissent d'exercer, ou de ne pas exercer ce pouvoir. En effet, s'il n'y a aucune d'autorité centrale capable de faire appliquer la loi sur une blockchain, la communauté a une responsabilité morale ou éthique d'intervenir afin de faire respecter les règles de droit (ainsi que celles de la communauté) afin de préserver l'ordre public.

La communauté, en tant que juge décentralisé œuvrant pour la préservation des « core values » du système et son usage conforme aux aspirations de ces concepteurs et utilisateurs, sera

---

[21] Traduction Framablog, voir « Code is Law – Traduction française du célèbre article de Lawrence Lessig » https://framablog.org/2010/05/22/code-is-law-lessig/

[22] L'Open Source est là encore une source d'inspiration utile en ce qu'elle connaît un cas analogue en matière de licence Open Source : le texte des licences étant parfois ambigüe (parce non informatisé), les communautés utilisatrice de ces licences développent parfois des interprétations divergentes de mêmes textes et il y a fort à parier qu'en cas de litige le juge accorde une place importante à une telle interprétation (comme source de droits) dans la mesure où elle ne porterait pas préjudice à un utilisateur de bonne foi.

donc en quelque sorte productrice d'une « lex electronica[23] » transcendant les frontières et les conceptions nationales. Contrairement aux juridictions nationales, sa capacité d'action sur le code lui permettra également de mettre en œuvre les sanctions qu'elle juge nécessaires, qu'il s'agisse de l'expulsion du membre ou du gel de ses « avoirs » sur le système.

[23] Voir notamment Mélanie Dulong de Rosnay, Les Golems du numérique. Droit d'auteur et Lex Electronica, Presses des Mines, coll. « Sciences sociales », 2016, 264 p.