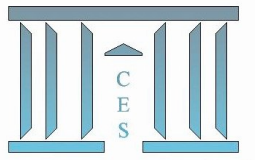
Blockchain publique et contrats intelligents (Smart Contrats). Les possibilités ouvertes par Ethéreum... et ses limites

Dominique Guegan

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Documents de Travail du
Centre d'Economie de la Sorbonne

CES Working Papers



Blockchain publique et contrats intelligents (Smart Contrats). Les possibilities ouvertes par Ethéreum… et ses limites

Dominique GUEGAN

2017.57

Maison des Sciences Économiques, 106-112 boulevard de L'Hôpital, 75647 Paris Cedex 13
http://centredeconomiesorbonne.univ-paris1.fr/
ISSN : 1955-611X

Blockchain publique et contrats intelligents (Smart Contrats)
Les possibilités ouvertes par Ethéreum… et ses limites

Dominique Guégan, Professeure, Université Paris1 Panthéon – Sorbonne, LabEx ReFi, IPAG

Ethereum est un protocole d'échanges décentralisés qui ne produit pas seulement une crypto-monnaie, mais permet aussi la création par les utilisateurs de smart contrats. Mais si la plateforme laisse beaucoup de libertés aux acteurs en termes de développement d'applications, des questions de sécurité et de robustesse se posent encore concernant le protocole, les plateformes, les bugs dans le code des contrats....

Dans un précédent article[1], nous avons évoqué la différence entre blockchain publique et blockchain privé. Le blockchain public[2] le plus connu est le Bitcoin[3]. Dans le présent développement, nous nous intéressons plus précisément à la mise en œuvre de contrats intelligents à l'aide de la technologie blockchain issue de celle utilisée pour développer le blockchain Ethereum dont l'Ether est le crypto combustible.
Bitcoin et Ethereum sont tous les deux des réseaux peer-to-peer et utilisent des concepts proches dont la chaine de blocs distribués, mais le blockchain de Satoshi est principalement orienté vers des transactions financières alors que celui de Vitalik Buterin ne se limite pas aux transactions et à la création de monnaie, mais permet aussi la création de contrats intelligents[4]. L'idée avec Ethereum est de mettre en place des Organisations Autonomes décentralisées (DAO) (forme de contrats de long terme qui contiennent des actifs et codent des règlements pour une organisation entière). Ainsi les concepts internes et la mise en œuvre diffèrent entre Bitcoin et Ethereum.

- Protocole Ethereum

En 2014 Vitalik Buterin introduit l'Ether (1ère phase 2013, lancement 2015, phase 2 : 2016). L'Ethereum est une crypto monnaie basée sur un protocole d'échange décentralisé et donc s'apparente à un blockchain public. L'objectif est de fournir un blockchain avec un langage de programmation permettant de coder des fonctions d'états arbitraires répondant à de multiples applications. Le langage utilisé est un langage de programmation intégré de Turing[5], permettant à quiconque d'écrire des contrats intelligents et des applications décentralisées avec ses propres règles pour la propriété, les formats de transaction et les fonctions de transition d'état. Ethereum est une machine virtuelle ayant son propre mécanisme de code interne, développant son propre algorithme avec un temps de résolution très rapide (15 secondes). L'idée est d'utiliser le principe du Blockchain (registre distribué

[1] Revue Banque 810, juillet/août 2017
[2] Blockchain étant un terme anglosaxon, quand il est utilisé dans un texte en français, son genre est masculin.
[3] Les limites de cette crypto monnaie et de son utilisation sont développées dans un papier publié dans « Capital Markets Law Journal », 2017, issue 4, « Bitcoins and Challenges for Financial Regulation », D. Guégan et A. Soritopoulou.
[4] [4] Actuellement il existe plus de 800 crypto-monnaies, certaines sont aussi basées sur la Preuve de Travail (PoW) et proposent des transactions sécurisées comme Litecoin (transactions plus rapides), Peercoin (transactions moins chères), ou s'intéressent à d'autres domaines que les transactions financières, colored coin ou Namecoin (système d'enregistrement de noms), etc.
[5] C'est-à-dire permettant à un être humain d'écrire un programme informatique (le code source) destiné à être exécuté par une machine, généralement un ordinateur. Le code source subit une transformation ou une évaluation dans une forme exploitable par la machine, ce qui permet d'obtenir un programme exécutable.

qui utilise la cryptographie) en le couplant avec des programmes autonomes capables d'exécuter automatiquement des conditions pré-définies. Rappelons que Bitcoin utilise des scripts qui ne permettent ni les « boucles », ni la vérification a priori de l'existence de fonds, ni la possibilité d'établir des contrats avec plusieurs étapes, ni des offres d'échange décentralisées ou des protocoles d'engagement cryptographique en deux étapes. L'idée sous-jacente à Ethereum est de construire une nouvelle chaine de blocs (avec la possibilité d'avoir des fonctionnalités illimitées), avec de nouveaux scripts en vue de proposer des applications basées sur le consensus, évolutives, standardisées, avec une fonctionnalité complète, une facilité de développement et une interopérabilité.

« On peut imaginer de nombreuses applications associées à ces contrats : enchères, marchés de prédiction, gestion d'identité, réputation, traçabilité des produits alimentaires, achat de crédit d'énergie, bornes de ravitaillement électrique, instruments financiers auto-exécutifs ...»

- Ethereum et Contrats intelligents

Lorsque le code pour le contrat intelligent est prêt et compilé, ce contrat intelligent est déployé sur le blockchain (au sein d'un Environnement de Développement Intégré (IDE)) et reçoit une adresse (à laquelle on va pouvoir envoyer des messages)[6]. Afin de ne pas se retrouver sur l'architecture publique du blockchain Ethereum (ce qui couterait entre autres, très cher), des répertoires sont créés permettant d'accéder au squelette d'une application Ethereum : le smart contrat travaille ainsi « avec un environnement privé ». Notons qu'un contrat intelligent correspond à des programmes informatiques autonomes qui, une fois démarrés, exécutent automatiquement des conditions définies au préalable avec des instructions conditionnelles du type « if…. Then…. », utilisant les informations disponibles sur le blockchain. Ces contrats doivent être capables de réduire les coûts de vérification, d'exécution, d'arbitrage et de fraude. Ils peuvent être amenés à gérer des fonds ou à authentifier des entités externes. Et donc dans cet environnement, le code doit être digne de confiance. Ainsi les développeurs et les utilisateurs de contrats intelligents doivent être en mesure de vérifier les propriétés de ces contrats et de disposer de correcteurs fiables.
On peut imaginer de nombreuses applications associées à ces contrats : enchères, marchés de prédiction, gestion d'identité, réputation, location d'appartements, de voitures, paiement des musiciens, traçabilité des produits alimentaires ou autres, achat de crédit d'énergie, bornes de ravitaillement électrique, instruments financiers auto-exécutifs (projet Corda).

- Questions ouvertes liées à cette technologique

Avant de se lancer dans l'aventure de l'utilisation d'un blockchain Ethereum pour réaliser des contrats intelligents, un certain nombre de questions nécessitent d'être prises en compte, qui pour l'instant n'ont pas été résolues. Elles concernent la logistique nécessaire pour mettre en œuvre les applications, la sécurité du code et des plateformes, le stockage, le hackage, le *phishing*, les failles dans le code, la question des boucles infinies, les bibliothèques, l'adéquation entre l'environnement EVM[7] et le code des contrats :
- La logistique nécessaire pour la majorité des applications proposées avec cette technologie n'est pas encore en place.
- La sécurité liée à la technologie utilisée est encore problématique. Ethereum a été créé pour développer des applications décentralisées (Dapps : *decentralized applications*), et il existe actuellement un écosystème de développement de ces applications avec des systèmes de programmation variés. Il est donc important pour les acteurs de s'assurer que le code utilisé peut

[6]Cette adresse est différente de celle d'un compte utilisé pour des transactions. Sur Ethereum, il y a donc deux types d'adresse.

[7] Machine Virtuelle Ethereum (voir ecadré fin de l'article).

assurer la sécurité du système et produire l'ensemble des tâches souhaitées avec les objectifs pré-établis tout en étant compatible avec le blockchain Ethereum.
- Il existe un certain nombre de bugs actuellement dans le code de ces contrats. Ces bugs vont des dépendances entre les commandes de transactions aux exceptions non réalisées.
- Un autre bug est lié à l'environnement DAO (Organisation Autonome Décentralisée) : un exemple est l'incident de juin 2016 au cours duquel, lors du transfert d'Ethers d'un compte à un autre, un attaquant a créé une fonction de récurrence permettant de retirer plus d'Ethers que la valeur du compte de contrepartie, entrainant une perte de 70 millions de dollars. Même si les développeurs ont remplacé la transaction de l'attaquant par une transaction de remboursement, une des conséquences a entrainé un *fork* et il existe ainsi deux devises Ethereum distinctes depuis décembre 2016 : Ethereum (ETH) et Ethereum Classic (ETC)
- Les contrats intelligents peuvent stocker de la monnaie ; s'ils sont hackés et que la crypto-monnaie est volée, le transfert se fait de manière irréversible et il est très difficile à tracer.
- Le montant des sommes stockées étant important, c'est un élément incitatif fort pour des attaques de la part des hackers.
- A priori tous les codes contractuels sont stockés publiquement sur la chaîne de blocs, ce qui permet aux attaquants de sonder le système avec toute l'information et de tester une série d'attaques.
- Le stockage, l'accès aux emplacements de mémoire à court terme, l'appel à d'autres contrats, avec possibilité de boucles infinies, seulement limitées par le coût en gaz[8], posent des risques réels avec la possibilité d'incursion de mineurs malveillants (voir exemple DAO ci-dessus)
- Est-ce que l'environnement complexe du Turning complet utilisé pour un système transactionnel de gestion d'argent est approprié ? Les contrats ont pour objectifs d'être amenés à communiquer entre eux. Ce paradigme de communication à contrats croisés a des conséquences importantes car les interactions contractuelles sont souvent complexes, et bien que les contrats soient sécurisés, leur sécurité peut néanmoins être violée au moment de l'appel du code d'un contrat tiers potentiellement malveillant et inconnu.
- Les bibliothèques fournies par les développeurs professionnels sont souvent en code «standard» et leur implémentation pour coder un modèle d'exécution EVM peut s'avérer complexe et diverger des implémentations compatibles avec le consensus invoqué par les utilisateurs. Des propositions sont en cours pour adapter l'environnement EVM, citons EtherLite qui propose un moteur d'exécution symbolique pour vérifier les bugs communs dans les contrats intelligents, ou bien Solidity IDE (intègrant Why3) écrit dans le langage Solidity de niveau supérieur. Donc nous constatons que de nombreux développements informatiques sont encore nécessaires pour s'assurer de la robustesse du code sous-jacent à la technologie Ethereum utilisée pour développer des contrats intelligents.
- Le piratage des plateformes ; très récemment (17 juillet 2017), un pirate est arrivé à voler plus de 8 millions de dollars en Ethers dans les toutes premières minutes de la levée de fonds (ICO) lancé par CoinDash. Le pirate a pris le contrôle du site officiel de CoinDash en remplaçant l'adresse Ethereum pour la récolte de fonds publiée sur la page web de CoinDash par sa propre adresse de portefeuille Ether. Ainsi lorsque des personnes envoyaient des Ethers à CoinDash ceux-ci étaient reçus par le pirate.
- Le *phishing* qui conduit le client vers un faux site de confiance où l'on demande la clef privée ou le mot de passe, ce qui permet au pirate d'accéder aux Ethers.

« De nombreux développements informatiques sont encore nécessaires pour s'assurer de la robustesse du code sous-jacent à la technologie Ethereum utilisée pour développer des contrats intelligents. »

- Quelle gouvernance pour Ethereum ?

Il est important de noter que Ethereum est un protocole d'échanges décentralisés qui à la différence de Bitcoin ne produit pas seulement une crypto-monnaie (le projet Ethereum à l'origine a été alimenté par une levée de fonds (ICO) de 18 millions de dollars), l'Ether, mais permet la création par les

[8] Unité interne utilisée dans l'EVM (voir encadré fin de l'article).

utilisateurs de smart contrats. L'écosystème Ethereum est évolutif, ce qui lui donne sa force, son originalité mais aussi pour l'instant une certaine faiblesse car il lui manque des outils de qualité en termes de logiciels utilisables, en particulier pour les contrats intelligents conçus pour imposer des règles commerciales protégeant la mise à jour d'un système d'enregistrement.
Ainsi on constate que lié au Protocole Ethereum et à la mise en place de contrats intelligents pour faire des opérations de gestion, des questions de sécurité et de robustesse se posent concernant le protocole, la sécurité des plateformes, les bugs dans le code des contrats, les défauts de codes et de multi-signatures. Pour l'instant la plateforme Ethereum laisse beaucoup de libertés aux acteurs, ce qui est positif car cela permet l'innovation en termes de développement des applications via les contrats intelligents, mais d'un autre coté augmentent les problèmes que nous avons évoqués. Quelle est la stratégie de la plateforme à plus long terme ?
Il apparait donc nécessaire de mettre en place une forme de gouvernance, qui ne remet pas en cause la technologie blockchain, qui est a-juridique par excellence. Cette technologie a du succès mais il faut savoir ce qu'elle apporte de plus aux environnements actuels et avec quelles contraintes et limites.
Concernant les failles de programmation : elles incombent tout d'abord aux développeurs (c'est le cas dans la programmation des codes multi-signatures - où plusieurs approbations sont nécessaires pour une transaction - qui a été à l'origine d'une perte de 30 millions lors de l'utilisation du logiciel de Parity technology, dont une nouvelle version est apparue le 19 juillet 2017) ; quand elles sont à l'origine de perte d'argent, les mineurs ont soit le choix de « récupérer » les Ethers mais ceci remet en cause l'immuabilité du code, ou bien de geler les transactions correspondant aux Ethers concernés, solution peu sécure dans le temps. On revient alors à la question de l'utilisation du Blockchain pour des applications financières… Quelle en est réellement la finalité ?
Il est aussi de la responsabilité des banques centrales d'informer les consommateurs des risques actuels encourus lors de l'utilisation de cette technologie avec les connaissances d'aujourd'hui, et d'inciter le consommateur à prendre un certain nombre de précautions avant de se lancer dans cette aventure. Aux USA, la SEC a produit le 25 juillet 2017 un rapport indiquant que les levées de fonds en crypto-monnaies (ICO) tombaient sous le coup des lois fédérales relatives aux titres et actions, étant ainsi dans son rôle de protecteur des investisseurs. Il semble indispensable que d'autres régulateurs nationaux prennent des mesures du même type et adaptent leur approche de manière globale.

« Il est aussi de la responsabilité des banques centrales d'informer les consommateurs des risques actuels encourus lors de l'utilisation de cette technologie avec les connaissances d'aujourd'hui, et d'inciter le consommateur à prendre un certain nombre de précautions »

- Les spécifications techniques d'Ethereum

Dans Ethereum, on trouve deux types de concept : des comptes et des fonctions de transition. Chaque compte comprend le nonce, son solde en Ether, le code du contrat, le stockage du compte. On distingue deux types de comptes :
- les comptes de propriété externes (contrôlés par des clés privées et identifiées par une adresse dérivée de la clé publique correspondante obtenue à l'aide de la fonction cryptographique SHA3)
- et les comptes liés à un contrat (identifiés par une adresse dérivée de manière déterministe à partir d'une adresse de compte externe connue et contrôlés par leur code) : ces comptes contiennent l'adresse du contrat, le solde du contrat, le code et l'état de l'exécution du programme. Un compte externe ne contient pas de code, mais on peut envoyer des messages d'un compte externe en créant et en signant une transaction. Chaque fois que le compte de contrat reçoit un message, son code s'active, lui permettant de lire et d'écrire dans le système interne et d'envoyer d'autres messages ou de créer des contrats à son tour.
L'environnement d'exécution pour les contrats Ethereum est la Machine Virtuelle Ethereum (EVM) où tout est écrit dans un langage de bytecode à bas niveau (le Turing complet). Le logiciel EVM se trouve sur les noeuds de réseau qui exécutent le bytecode du contrat. Chaque bloc Ethereum est stocké sur

une racine appelée « Patricia tree », et contient l'état de chaque compte, l'historique complet de l'exécution des programmes distribués, le numéro du bloc et le processus de la validation.
Ethereum met en œuvre des frais de minage pour les mêmes raisons que Bitcoin. En effet l'EVM permet d'exécuter des contrats avec des boucles potentiellement infinies ou des opérations coûteuses en termes de calcul. Pour compenser les mineurs pour ces calculs qui prennent beaucoup de temps, les contrats consomment un gaz. Le gaz est une unité interne utilisée dans l'EVM (Le prix du gaz est un prix acheté en Ether (ETH)). A chaque instruction EVM est affecté le coût en gaz. Lorsqu'un compte appelle un compte, l'appel est accompagné de l'équilibre du compte et des données transférées. Le solde transféré est déposé sur le compte appelé. Lorsque le compte appelé est un compte externe, il se produit un simple transfert de solde. Lorsque le compte appelé est un contrat, après le transfert du solde, le code du contrat appelé est exécuté s'il y a assez de gaz pour exécuter toutes les instructions (chaque instruction consommant du gaz).
Les langages des contrats peuvent être Solidity (qui est similaire à Javascript), Viper ou Serpent (qui est similaire à Python), ou Lem. Ils sont compilés dans le bytecode EVM[9].

[9] Récemment un cadre permettant de développer des applications indépendantes et avec des références lisibles a été proposé : il s'agit de l'environnement KEVM qui formalise plusieurs langages : C, Java et JavaScrip