POLYTECH NICE-SOPHIA
Département Informatique

UNIVERSITÉ CÔTE D'AZUR

State of the art

PER2024-040 - (Semi)Decentralized Digital Currencies and Mutable Smart Contracts for Permissioned Blockchains Distributed Ledgers for the European Area

December 19, 2024

Students :

Arnaud	DUMANOIS	arnaud.dumanois@etu.univ-cotedazur.fr
Simon	BEUREL	simon.beurel@etu.univ-cotedazur.fr

Tutors :

Luigi LIQUORI

Keywords: Blockchain; Smart Contrats; Ethereum, Règlementation européenne

Abstract :

Au cours des dernières années, la technologie Blockchain s'est de plus en plus exportée et s'est imposée comme une nouvelle technologie apportant des solutions innovantes à des problèmes anciens. C'est notamment dans le cadre d'une finance décentralisée (DeFi) que le projet Ethereum utilise le plein potentiel de la blockchain à travers sa cryptomonnaie, l'Ether, mais également grâce aux contrats intelligents. Cependant, ces contrats font face à deux grandes problématiques liées, la première étant l'impossibilité de changer le code source du contrat une fois ce dernier déployé, la deuxième étant l'apparition de plus en plus de réglementations européennes pour fixer un cadre juridique à ces contrats.

Contents

Introduction

Dans la dernière decade, parmi toutes les nouvelles technologies émergentes, se cache une technologie révolutionnaire qui possède un gros impact : la Blockchain. La technologie Blockchain consiste à une décentralisation de la confiance, permettant de sécuriser les transactions et les échanges sans dépendre d'une autorité centrale. La Blockchain, notamment la technologie utilisée dans la plupart des grandes cryptomonnaies comme Bitcoin ou Ethereum est une technologie de plus en plus utilisée dans divers domaines.

Elle possède plusieurs caractéristiques comme notamment : la décentralisation (pas de serveur central mais des « nœuds » sur le réseau), et la pseudo-anonymisation des utilisateurs. L'un des cas d'utilisation de la blockchain est la DeFi (Decentralized Finance), qui correspond à un fonctionnement différent de la finance actuelle que nous connaissons tous avec des banques. Ici, le but est d'utiliser la blockchain pour référencer toutes les transactions en utilisant une cryptomonnaie associée.

Dans le cadre de notre PER, nous allons nous spécialiser sur un réseau peer-to-peer Blockchain en particulier : Le réseau Ethereum. Ce réseau est très innovant car, contrairement au réseau Bitcoin où seules des transactions A$\leftarrow B$ sont possibles (où A et B sont des utilisateurs), le réseau Ethereum permet, lui, la création de contrats sur lesquels un utilisateur peut interagir pour réaliser différentes actions, ces contrats s'appellent : Les Smarts Contracts (contrats intelligents).

Ces contrats intelligents se présentent comme des objets qui sont présents dans la blockchain sous forme de byte code, et sur lequel chaque utilisateur peut interagir en appelant des fonctions propres à chaque contrat. Ils sont (dans le cas du réseau Ethereum) écrit dans le langage Solidity, et c'est le byte code généré à la compilation qui est "broadcasté" sur la Blockchain Ethereum.

Cependant, l'un des principes de la Blockchain est que tout ce qui est présent dessus est immuable, ainsi, si une erreur est présente dans le contrat intelligent, il est impossible de le modifier, ce qui est très dangereux et peut causer des cyberattaques si le contrat est mal rédigé c'est tout le but de notre PER : Comprendre quels risques/impacts sont liés à l'immuabilité des contrats intelligents ? Analyser les solutions existantes pour bypasser l'immutabilité des contrats intelligents et proposer une nouvelle solution pour résoudre ce problème d'immuabilité.

Ce projet de recherche fait suite au stage de Mr. Rigotti Giovanni et de sa thèse de Master intitulé "Smart Contracts: A Research-Driven Approach to Enhance Upgradeability" [1]

1 Blockchain et Ethereum

Dans cette partie, nous détaillerons l'état de l'art de la technologie Blockchain et du réseau Ethereum en 2024.

1.1 Blockchain

La blockchain, en tant que technologie disruptive, a émergé comme une solution innovante pour résoudre les problèmes liés à la centralisation des données et à la transparence des transactions. Développée initialement pour soutenir les transactions en Bitcoin dans l'article fondateur de Satoshi Nakamoto en 2008[2], elle est aujourd'hui perçue comme un registre numérique décentralisé et immuable. Chaque enregistrement dans une blockchain est validé à l'aide de mécanismes de consensus, tels que la preuve de travail (Proof of Work) ou la preuve d'enjeu (Proof of Stake). Ces mécanismes garantissent l'intégrité des données en rendant les manipulations malveillantes quasiment impossibles.[2]

Au-delà des crypto-monnaies, la blockchain a démontré sa polyvalence dans de nombreux domaines. Dans le secteur financier, elle est utilisée pour des paiements transfrontaliers rapides et économiques. Par exemple, Ripple et Stellar[3][4] facilitent ces transactions tout en éliminant les intermédiaires coûteux. Dans le domaine de la santé, des solutions comme Medicalchain[5] permettent de stocker et partager des dossiers médicaux de manière sécurisée, offrant un accès contrôlé aux parties prenantes, y compris les médecins et les patients. En logistique, des entreprises comme IBM et Maersk[6] ont développé des plateformes basées sur la blockchain pour optimiser la traçabilité des chaînes d'approvisionnement, minimisant ainsi les risques de fraude et améliorant l'efficacité opérationnelle.[7]

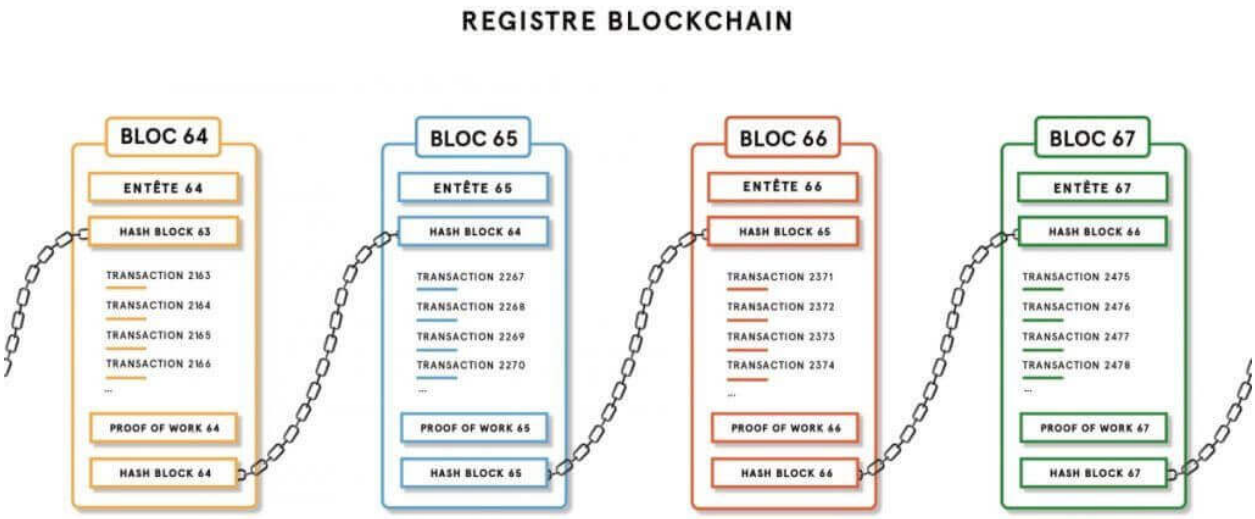

Figure 1: Schema classique du principe de Blockchain

1.2 Réseau Ethereum

C'est dans le registre du développement de la technologie Blockchain que le réseau Ethereum a trouvé naissance. En 2014, Vitalik Buterin publie un papier de recherche présentant un nouveau

réseau Blockchain [8] dont le but principal est de pouvoir proposer une blockchain pour des applications décentralisées et dotée d'un langage de programmation Turing-complet : Ethereum. Cette vision est, à l'époque, très ambitieuse et s'inscrit dans un esprit de concurrencer le réseau Bitcoin[2], car là où Bitcoin permet simplement de faire que de simples transactions, le cœur d'Ethereum est de permettre à ses utilisateurs de pouvoir utiliser des applications décentralisées et notamment des contrats intelligents.

Dès 2014, Vitalik Buterin définit les contrats intelligents par "des accords autonomes qui exécutent automatiquement des instructions définies lorsqu'un ensemble de conditions prédéfinies est rempli". Ces contrats ont pour but de pouvoir ajouter de la flexibilité au réseau blockchain en permettant à quiconque qui le souhaite de pouvoir créer son propre contrat et de le déployer sur la blockchain.

Le réseau Ethereum intègre également un nouveau concept non utilisé dans Bitcoin: le "gaz" et les "frais". Le Gaz Ethereum correspond à des frais de transactions qui seront payés par la personne voulant procéder à cette transaction, et ils seront attribués au mineur du bloc qui contiendra cette transaction. Cette nouveauté est notamment introduite pour éviter la surcharge du réseau par de potentiels utilisateurs malveillants qui souhaiteraient réaliser des boucles infinis, par exemple lors de transactions.

Bien qu'initialement le réseau Ethereum utilisait le concept de Proof-of-Work, ce dernier a été remplacé en 2022 par le mécanisme de consensus Proof-of-Stake (PoS) lors de "The Merge" [9], une mise à jour majeure visant à améliorer l'efficacité énergétique et la sécurité du réseau. Contrairement au Proof-of-Work, qui repose sur la puissance de calcul des mineurs pour valider les transactions et sécuriser la blockchain, le Proof-of-Stake utilise des validateurs sélectionnés en fonction de la quantité d'ether (ETH) qu'ils ont mis en jeu, ou "stake". Ce modèle réduit considérablement la consommation d'énergie, car il n'exige pas de résoudre des calculs complexes pour valider un bloc. Les validateurs sont incités à agir honnêtement grâce à un mécanisme de pénalités, appelé "slashing", qui peut confisquer une partie ou la totalité de leur mise en cas de comportement malveillant. Cette transition a permis à Ethereum de poser les bases d'un réseau plus durable, évolutif et sécurisé, tout en favorisant une participation plus large à son écosystème.

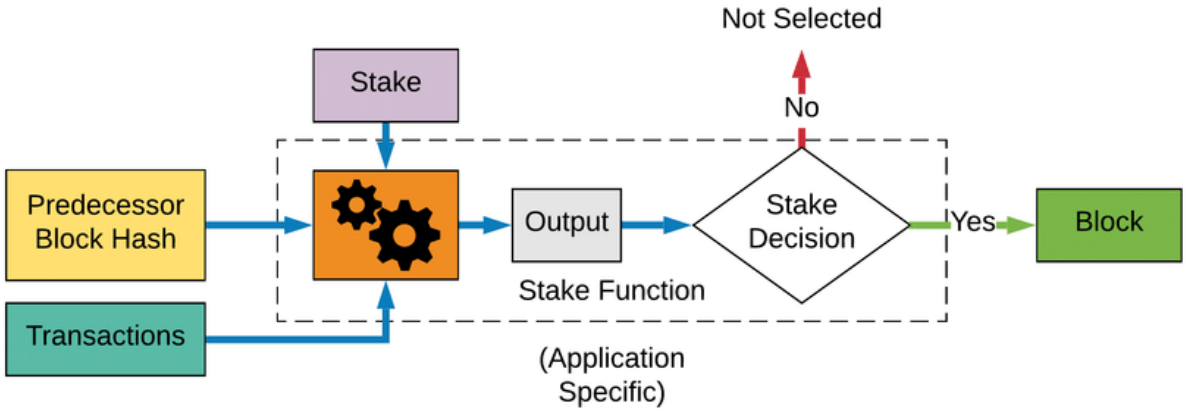

Figure 2: Schéma explicatif du Proof-of-Stake (PoS) utilisé dans Ethereum [10]

2 Smarts Contracts Solidity Language

Dans cette partie, nous détaillerons le concept de Smarts Contrats écrits avec le langage Solidity, et les propositions de directives élaborées au fur et à mesure du temps.

2.1 Présentation globale

Les contrats intelligents [11] (Smarts Contracts) sont un élément essentiel de la philosophie du réseau Ethereum, et pour cause, ils sont mentionnés dès 2014 dans le white paper publié par Vitalik Buterin, c'est-à-dire aux origines d'Ethereum. La réflexion des contrats intelligents vient notamment d'un problème de rigidité perçu sur Bitcoin (le concurrent principal de l'époque) : Il est impossible pour Bitcoin de pouvoir créer un service directement implémenté dans la blockchain permettant de pouvoir réaliser des actions logiques écrites par un humain sans passer par une application tierce. Or, c'est exactement à cette problématique que les contrats intelligents vont répondre, car grâce à Ethereum, il est possible de pouvoir intégrer directement au coeur de la blockchain du bytecode avec lequel n'importe quel utilisateur sera capable d'interagir, et dont celui aura un comportement logique prédéfini par un humain.

L'un des secteurs d'utilisation des contrats intelligents est le secteur de la finance. En effet, ce secteur, qui est très centralisé, a vocation avec le réseau Ethereum à devenir de plus en plus décentralisé (ce qu'on appelle la Decentralized Finance, DeFi), et c'est dans cette optique que de nombreuses banques s'intéressent de plus en plus à cette technologie.

2.2 Présentation technique

Les smart contracts Ethereum sont principalement écrits en Solidity, un langage de programmation de haut niveau conçu spécifiquement pour le développement de contrats intelligents sur Ethereum. Solidity est un langage statiquement typé, inspiré de JavaScript, Python et C++, et conçu pour être déployé sur la machine virtuelle Ethereum (EVM). Il permet la définition de fonctions, variables et structures de données adaptées aux exigences de la blockchain, telles que la gestion de la logique de transaction et des états. Solidity supporte les fonctions et structures complexes, telles que les contrats hérités et les interfaces, et offre des mécanismes de gestion de la sécurité, comme la prévention des attaques courantes (par exemple, réentrance).

Les contrats intelligents sont exécutés sur la Machine Virtuelle Ethereum (EVM)[12], un logiciel en perpétuelle exécution sur certains nœuds présents au sein du réseau Ethereum. L'EVM est responsable de l'interprétation des smart contracts et de l'exécution des instructions contenues dans leur code. Chaque nœud du réseau Ethereum peut exécuter une instance de l'EVM, ce qui assure la synchronisation et l'unicité des transactions sur la blockchain. L'EVM permet de gérer les appels de fonctions, l'accès aux données du contrat et l'ajout de nouveaux blocs au sein de la blockchain.

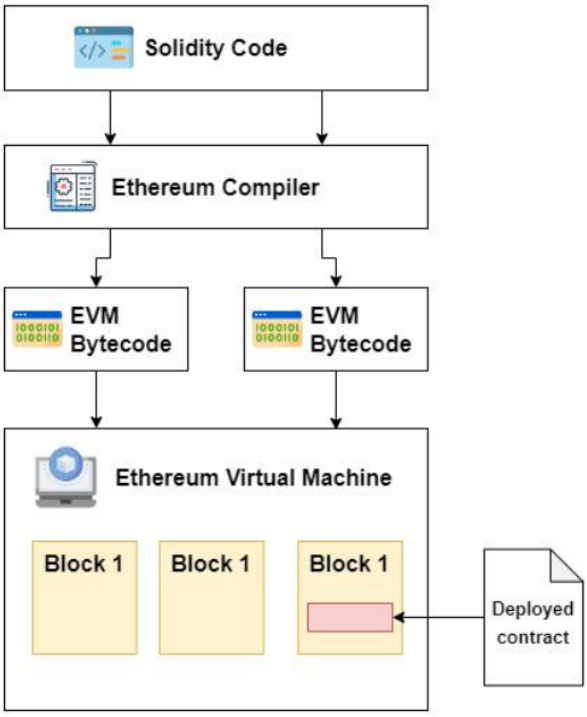

Figure 3: Schema explicatif d'un déploiement d'un contrat intelligent[13]

2.3 Propositions de directives

Officiellement, il n'existe pas de règles imposées par la Fondation Ethereum[14] pour créer des contrats intelligents ; la seule recommandation, comme indiqué lors du paragraphe précédent, est qu'il faut que ce contrat soit rédigé dans le langage Solidity. Cependant, à cause du développement accru des contrats intelligents au cours de ces dernières années, les différents gouvernements mondiaux imposent de plus en plus de directives techniques et légales pour garantir la sécurité de leurs concitoyens.

En 2016, à la suite d'une collaboration entre la banque britannique Barclays et le chercheur Mr. Christopher D. Clack et al., un papier de recherche est publié dont le but est de pouvoir créer un template [15] en posant des bases fondamentales pour la création des futurs contrats intelligents. L'un des points importants est la définition d'un contrat intelligent, sur ce sujet les créateurs de ce document indiquent qu'un contrat intelligent est un "accord automatisable et exécutable". L'un des aspects très importants qui est évoqué est que les contrats intelligents doivent être **capables de subir des modifications au fur et à mesure du temps** pour trois raisons. La première est la raison juridique, en effet, ces contrats sont souvent déployés par des banques, des institutions publiques, etc. et ils doivent donc toujours être conformes vis-à-vis des différentes lois qui sont appliquées. La deuxième raison est la raison technologique, qui est expliquée par le fait que tout code informatique écrit peut être sujet à des bugs, des failles de sécurité, etc. La troisième raison est liée au fait que comme pour le monde technologique, le monde judiciaire peut faire des erreurs vis-à-vis des différentes clauses qui peuvent être indiquées dans un contrat.

En résumé, beaucoup de personnalités ont tenté/sont en train de proposer une directive commune vis-à-vis de l'élaboration de contrats intelligents mis à disposition au grand public. Cependant, le point noir des contrats intelligents qui sera abordé dans la partie suivante est la difficulté à retirer le caractère immuable des contrats, ce qui empêche de garder les contrats intelligents en accord avec les réglementations légales et technologiques.

3 Versionning des Smarts Contracts

Le versionning est défini par Wikipédia comme “le processus d’attribution de noms de version uniques ou de numéros de version uniques à des états uniques de logiciels informatiques”. Dans cette partie nous détaillerons l’importance d’avoir un système de versionning des contrats intelligents, les failles liées à ces contrats et nous présenterons notre projet nommé “PolyCoin”.

3.1 La sécurité des contrats

Les contrats intelligents Ethereum sont responsables de plusieurs millions voire milliards de dollars à travers le monde, ainsi, garantir une sécurité sans faille est l’un des éléments principaux lors de leur création, d’autant plus que ces contrats sont immuables, c’est-à-dire qu’une fois que ces derniers sont déployés, il est impossible de changer le code source sur la blockchain. La plus grosse attaque s’est déroulée en 2016, s’appelle TheDAO Hack [16] et au cours de cette attaque les pirates ont volé 3 600 000 Ether, ce qui vaut 13 501 686 370€ à l’heure où ce document est écrit (17 Décembre 2024)

En 2016, une équipe de plusieurs chercheurs se sont penchés sur l’étude de la sécurité des contrats intelligents présents sur la blockchain Ethereum [17]. Au cours de leur étude, ils ont développé un outil nommé Oyente [18] capable de pouvoir trouver différentes failles connues dans des contrats intelligents déployés. Leur analyse était assez simple : Ils ont récupéré 19 366 contrats intelligents déployés, et pour chaque contrat ils ont utilisé leur outil Oyente pour vérifier si ces derniers possédaient une faille ou non. Lors de la conclusion de leur étude, les différents chercheurs ont trouvé que 8 833 contrats / 19 366 possèdent une faille de sécurité détectée par l’outil Oyente.

De nombreuses failles existent dans le monde des contrats intelligents et, comme l’a montré l’étude présentée ci-dessus, de vrais contrats déployés sur la blockchain Ethereum qui possèdent de l’argent détiennent également des failles de sécurité. Cela pose une véritable problématique, car la caractéristique de l’immuabilité des contrats intelligents est un frein à une barrière pour un développeur souhaitant réparer un code source contenant une faille importante de sécurité.

3.2 Les systèmes versionning actuels

Différents systèmes de versionning ont été développés par la communauté Ethereum au cours des dernières années pour répondre aux différentes problématiques énoncées dernièrement. En 2023, une équipe de chercheurs islandais et canadiens s’est intéressée aux différentes solutions utilisées pour pallier ce problème de versionning. Ils ont pu identifier 3 méthodes utilisées [19]:

- **Patrons de proxy**: Ce sont des modèles de conception permettant de séparer la logique du contrat de son adresse de déploiement. L’idée est d’utiliser un contrat proxy qui redirige les appels vers un autre contrat, souvent appelé "logic contract", où la logique du code est réellement définie. Cela permet de mettre à jour la logique du contrat sans changer l’adresse du contrat déployé, garantissant ainsi la persistance des données et l’interaction avec les utilisateurs.

- **Séparation des Données et de la Logique**: C’est un principe qui consiste à dissocier le stockage des données (comme les états) et la logique de traitement (comme les fonctions d’exécution) dans un contrat intelligent. Cela permet de mettre à jour la logique sans modifier les données, offrant ainsi plus de flexibilité et de facilité de maintenance.

- **Migration des Données**: C’est un processus de transfert des données d’un ancien système ou contrat vers un nouveau, tout en préservant leur intégrité et leur accessibilité. Ce processus implique de déplacer les données stockées dans un contrat vers un autre contrat, souvent lorsque la logique du contrat est mise à jour ou modifiée. La migration des données permet de garantir que les informations précédemment enregistrées restent accessibles et cohérentes, même après le déploiement d’un nouveau contrat.

Dans certains cas, comme dans celui des patrons de proxy, des entreprises privées comme OpenZeppelin [20] ont développé des librairies permettant de faciliter l'intégration de ce patron dans un contrat.

3.3 PolyCoin

Au cours de notre projet, nous nous sommes rendus compte que les solutions proposées, comme les patrons de proxy par exemple, ne permettaient pas de pouvoir garder un élément important dans le domaine juridique : La traçabilité. En effet, de plus en plus les gouvernements du monde souhaitent pouvoir garder une trace écrite/numérique de tous les changements effectués sur un logiciel, un aliment, etc. Cette traçabilité, la technologie Blockchain y répond grandement, car elle permet de pouvoir sauvegarder des transactions (ici changement d'états) de manière permanente en garantissant que personne n'arrivera à altérer les blocs déjà construits.

C'est donc pour cela que nous avons décidé de créer notre propre solution de versionning : PolyCoin [21]. Cette solution est une blockchain permettant de pouvoir garder en mémoire au sein des blocks les différentes modifications réalisées sur un contrat intelligent (nouvelle version), mais également elle permet de pouvoir retrouver les personnes morales ou physiques qui ont été inclues dans le développement d'un contrat intelligent.

La blockchain PolyCoin possède deux types de blocs. Le premier type s'appelle le bloc IDENTIFIER, il permet de pouvoir renseigner des informations sur une personne morale ou physique en y renseignant notamment son nom, son certificat électronique délivré par une autorité de certification et l'adresse de son portefeuille sur la blockchain Ethereum, comme montré sur la Figure 4.

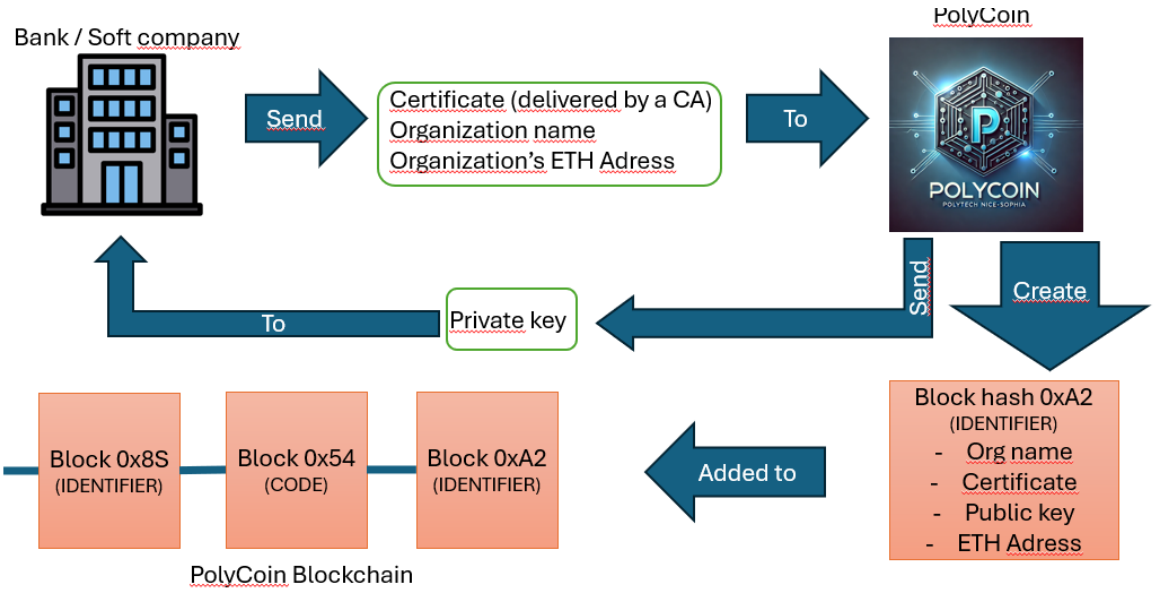

Figure 4: Schéma de la création d'un bloc IDENTIFIER[21]

Le deuxième type de block présent dans PolyCoin s'appelle le block CODE. Ce bloc correspond à une version d'un contrat intelligent qui a été développée par une personne morale ou physique déjà inscrite au sein de la blockchain PolyCoin. Pour pouvoir créer ce bloc, les entités responsables du contrat doivent chacune le signer avec leurs clés privées, puis, comme indiqué sur la figure 5, l'une des entités va fournir à PolyCoin le code source du contrat ainsi que les différentes signatures pour pouvoir créer ce nouveau bloc.

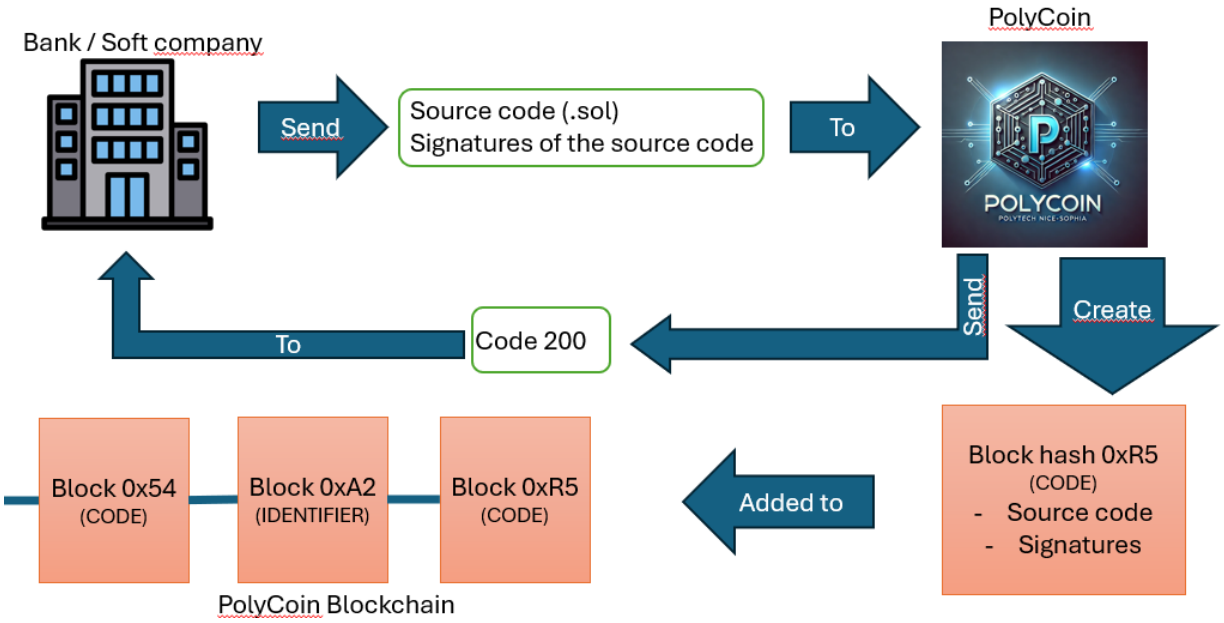

Figure 5: Schéma de la création d'un bloc CODE[21]

Le cœur du projet PolyCoin est d'utiliser le principe de la signature cryptographique pour retrouver qui est à l'origine d'un block CODE correspondant à une version d'un contrat intelligent. Comme indiqué sur la figure 6, pour réaliser ce principe, PolyCoin peut simplement chercher parmi toutes les clés publiques attribuées aux blocs IDENTIFIER, lesquelles permettent de vérifier un bloc CODE en question. Ainsi, PolyCoin permet de pouvoir répondre à deux problématiques liées aux contrats intelligents : Comment garder un suivi efficace de l'évolution d'un contrat intelligent et comment retrouver les responsables en cas de faille de sécurité ?

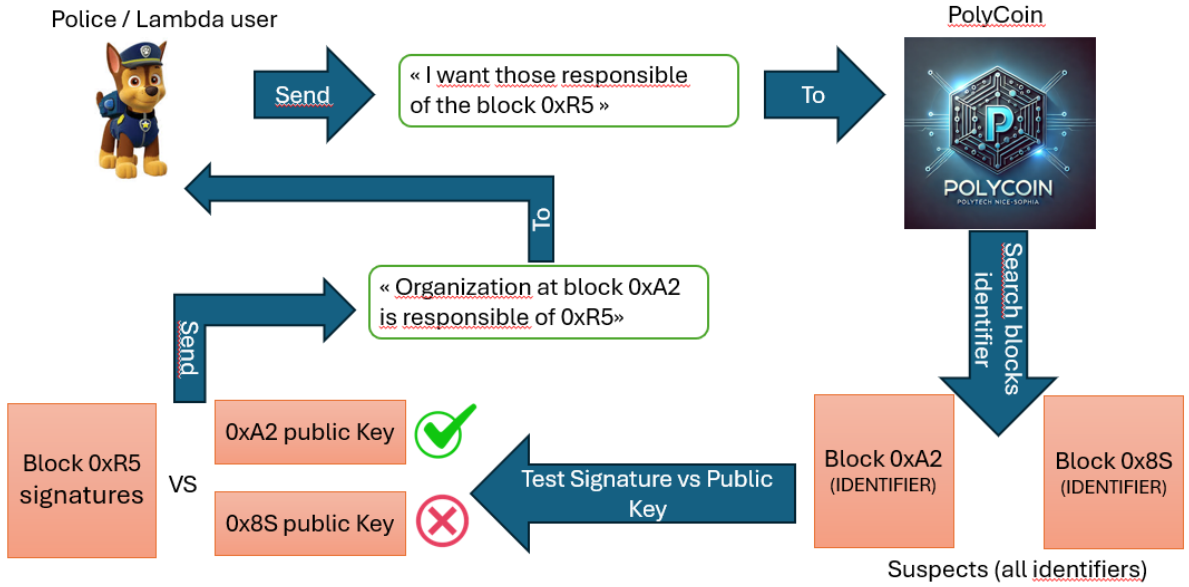

Figure 6: Schéma de la récupération du responsable d'un block CODE [21]

4 Plan d'avancement de la période à plein temps

Au cours de cette partie, nous détaillerons notre avancée vis-à-vis du rapport final à rendre dans le cadre de notre projet.

4.1 Introduction: Etat avancé/fini

Pour la première partie du rapport, nous sommes confiants sur notre avancée. L'élaboration de ce papier d'état de l'art nous a permis de pouvoir bien comprendre comment restituer à l'écrit les différentes connaissances apprises au cours de ces derniers mois. Lors de l'introduction, il sera essentiel de bien faire comprendre la technologie Blockchain, Ethereum, les contrats intelligents, mais également l'intérêt et les problématiques qui tournent autour de ces derniers.

4.2 Analyse des solutions existantes: Etat avancé

Pour la deuxième partie du rapport, nous allons faire une analyse des solutions existantes et utilisées au sein de la communauté Ethereum pour pouvoir maintenir un versionning des contrats intelligents. Nous sommes avancés sur cette partie car nous avons déjà réalisé des expériences sur des contrats réels et nous avons pu en prendre des notes. Il sera intéressant de comparer les différentes solutions en termes de complexité, mais également en comparant les différents coûts que cela implique.

4.3 Présentation de notre solution PolyCoin : En cours

Comme mentionné précédemment dans ce rapport, nous allons proposer notre propre solution pour répondre à ce problème de versioning : PolyCoin. Le développement théorique de PolyCoin est presque achevé, il ne reste plus qu'à finir de la développer en langage de programmation Python et à le déployer sur un serveur avec la documentation adéquate.

4.4 Proposition d'une amélioration du compilateur : En cours

Comme mentionné dans le rapport, nous allons également essayer de proposer une amélioration au compilateur Solidity pour que ce dernier prenne en compte la blockchain PolyCoin dans son travail. Cette tâche est encore en cours de recherche et elle est notamment dépendante directement de l'avancée du développement de PolyCoin.

4.5 Conclusion: Non commencée

La partie qui conclura notre rapport n'est pas encore commencée, en effet nous réfléchirons à la conclusion du rapport final que quand nous aurons plus avancé dans notre travail de recherche et développement car de nouvelles problématiques/idées vont probablement apparaître au fur et à mesure des semaines.

Conclusions

En conclusion, nous avons pu voir à travers ce papier que la technologie Blockchain pouvait être utilisée dans des cas bien précis de la finance, comme dans la blockchain Ethereum, grâce aux contrats intelligents permettant de pouvoir apporter des actions logiques flexibles à la suite d'une transaction, contrairement à des blockchains plus rigides comme BitCoin.

Cependant, ces contrats intelligents sont les victimes de beaucoup d'attaques d'utilisateurs malveillants souhaitant récupérer l'intégralité des fonds qui sont déposés dessus. L'attaque la plus connue réalisée sur un contrat intelligent s'appelle TheDAO Hack et a permis aux attaquants de dérober des millions d'Ether dont la valeur totale s'élève à plusieurs milliards d'euros en décembre 2024.

De plus, les développeurs de contrats intelligents sont soumis à une contrainte particulière qui est qu'il est impossible de modifier directement un contrat déployé à une adresse spécifique sur la blockchain. Pour pallier à ce problème, de nombreuses solutions ont été proposées par la communauté comme les patrons de proxy, etc.

Au cours de notre projet, nous allons proposer notre propre solution de versionning de contrats intelligents : PolyCoin. Le projet PolyCoin consiste en une blockchain permettant de sauvegarder toutes les versions possibles d'un contrat intelligent, mais également de pouvoir retrouver les personnes morales ou physiques responsables de ce contrat.

Pour finir, nous souhaiterions, si le projet PolyCoin s'achève avec succès, proposer une modification du compilateur du langage Solidity (langage principal pour la création de contrats intelligents) pour que ce dernier intègre une vérification sur PolyCoin lorsqu'un utilisateur souhaite interagir avec un contrat déployé, pour s'assurer que ce dernier n'est pas un contrat malveillant.

References

[1] R. Giovanni, "Smart contracts: A research-driven approach to enhance upgradeability," 2024.

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Cryptography and Security Research*, 2008.

[3] D. Schwartz, N. Youngs, and A. Britto, "The ripple protocol consensus algorithm," 2018.

[4] D. Mazières, "The stellar consensus protocol: A federated model for internet-level consensus," 2016.

[5] Medicalchain, https://medicalchain.com/en/.

[6] IBM, https://www.ibm.com/fr-fr/blockchain-supply-chain.

[7] N. Kshetri, "1 blockchain's roles in meeting key supply chain management objectives," 2018.

[8] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform." 2014.

[9] D. Grandjean, L. Heimbach, and R. Wattenhofer, "Ethereum proof-of-stake consensus layer: Participation and decentralization," 2023.

[10] A. Paul, "Torrent driven (td) coin: A crypto coin with built in distributed data storage system," 2023.

[11] Q. Huang, "Ethereum: Introduction, expectation, and implementation," *Highlights in Science, Engineering and Technology*, 2023.

[12] R. Norvill, B. B. F. Pontiveros, R. State, and A. J. Cullen, "Visual emulation for ethereum's virtual machine," *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018.

[13] D. Rhodes, "What is ethereum virtual machine (evm)?" 2024.

[14] E. Foundation, https://ethereum.foundation/.

[15] C. D. Clack, V. A. Bakshi, and L. Braine, "Smart contract templates: foundations, design landscape and research directions," 2017.

[16] M. Mehar, C. Shier, A. Giambattista, E. Gong, G. Fletcher, R. Sanayhie, H. M. Kim, and M. Laskowski, "Understanding a revolutionary and flawed grand experiment in blockchain: The dao attack," *Journal of Cases on Information Technology 21(1) 19-32.*, 2019.

[17] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," *2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[18] Enzymefinance, "Oyente tool," https://github.com/enzymefinance/oyente, 2020.

[19] I. Qasse, M. Hamdaqa, and B. Þór Jónsson, "Smart contract upgradeability on the ethereum blockchain platform: An exploratory study," 2023.

[20] OpenZeppelin, "Openzeppelin's proxy library," https://docs.openzeppelin.com/contracts/4.x/api/proxy.

[21] A. Dumanois and S. Beurel, "Polycoin," https://github.com/simonbeurel/PolyCoin, 2024.