More Flexible Smart Contract Languages for Permissioned Blockchains and Distributed Ledgers

Louis-Joseph AMAS

16 décembre 2020

Abstract

Ce rapport présente mon projet de fin d'études mené à l'Inria Sophia Antipolis-Méditerranée. Le projet consista à explorer le fonctionnement des Smart Contract. L'objectif a été de rechercher des moyens de rendre les smarts contracts (qui par définition son non immuable) modifiable. Plus en détails, j'ai dû concevoir la mise en place de prototype (PoC: Proof of concept) démontrant la possibilité de modification, de plus, j'ai dû comprendre les problèmes soulevés par cette nouvelle liberté et proposé de solutions. Dans le cadre de ce projet, j'ai pu travailler avec Mr Luigi Liquori, Mr Daniel de Carvalho et Mr Mansur Khazeev dans le cadre de multiples réunions. J'ai pu réaliser trois présentations oral sur ce sujet afin d'aider Mr Khazeev à réaliser sa thèse.

Sommaire

1 Introduction

Tout d'abord, il est nécessaire d'expliquer le concept des technologies concernées. La blockchain est une technologie permettant de faire la représentation d'un registre de compte de manière numérique et distribué. Cela permet de s'affranchir d'un tiers de confiance, la première technologie réalisant ces objectifs est le Bitcoin.

Le Bitcoin est une monnaie décentralisée basé sur la blockchain. C'est un registre de compte contenant toutes les transactions effectuées. Pour sécuriser les accès à ce registre (permettre au ayant droit d'un compte de dépenser son argent), Bitcoin combine deux principes fondamentaux, la cryptographie à clé publique ainsi que la signature numérique, La cryptographie à clé publique permet de vérifier que un utilisateur possède réellement un compte. La signature numérique permet de vérifier l'intégrité des transactions. Les transactions sont des éléments d'un bloc et les blocs sont chaînés (mis les uns après les autres) à l'aide de la signature numérique. Ce principe de chaîne permet d'assurer que si l'une des transactions est modifiée alors la signature est modifiée. Une personne malveillante modifiant une transaction déjà inscrite dans la chaîne de bloc sera détectée.

Des améliorations à ce principe de blockchain on était faite et il existe maintenant une technologie appelée "smart contract". Cette technologie est un moyen d'exécuter un programme de manière distribuée, permettant de s'affranchir d'un tiers de confiance. Cela peut être utilisé pour réaliser par exemple des systèmes de vote sans possibilité de triche ou de modification du résultat ou bien représenter un jeu d'argent (poker, paris sportif...). Cette technologie est enregistrée de la même manière dans un registre non modifiable. L'ojectif de ce projet et de rechercher les manières de pouvoir modifier le programme exécuter par un smart contract afin d'adapter le comportement dynamiquement selon les besoins.

2 Context technologique

Status: En cours

Sujets:

- Blockchain
- Bitcoin
- Ethereum
- EVM
- Solidity

2.1 Blockchain

2.1.1 Bitcoin

Une (ou un) blockchain, ou chaîne de blocs Techniquement, il s'agit d'une base de données distribuée dont les informations stockées sont vérifiées et groupées à intervalles de temps réguliers sous forme de bloc, formant ainsi une chaîne de blocs. L'ensemble est sécurisé par cryptographie. La chaîne de blocs est alors stocké sur de multiple machine en même temps que l'on appelle des nœuds. Cette chaîne permet de protéger contre la falsification et la modification de la base de données par les nœuds de stockage. C'est donc un registre distribué et sécurisé de toutes les transactions effectuées depuis le démarrage du système réparti.

La première technologie déployant un tel système est le Bitcoin. Le Bitcoin utilise la blockchain afin de représenter un registre de compte distribué. Les utilisateurs souhaitant utiliser cette cryptomonnaie génère une paire de clé asymétrique.

Pour rappel, les chiffrements asymétriques sont constitués de deux clés l'une publique disponible par tous le monde et l'une privée qui doit rester secrète. Les méthodes de chiffrements à clé publique permettent e de rendre un message inintelligible par toute personne n'ayant pas la clé privée d'un message chiffré par la clé publique complémentaire. Cette technique de chiffrement permet aussi de vérifier (Signer numériquement) qu'un message a bien été écrit par quelqu'un possédant une paire de clé (Privée, publique).

Bitcoin utilise cette fonctionnalité afin d'autoriser une personne à échanger son argent. Un bloc de la blockchain Bitcoin contient des transactions (des échanges de jeton BTC) signer par la clé privée de l'utilisateur souhaitant transférer ses jetons. Bitcoin calcul le hash de toutes ses transactions et crée un bloc à l'aide de la concaténation de tous les hashs de toutes les transactions ainsi que le hash du bloc précédent.

Cela permet de rendre quasi impossible la falsification et la modification de la chaîne. En effet, un attaquant voulant modifier la chaîne serait détecté car, le hash de son bloc serait invalide s'il modifie même qu'une transaction. Le Bitcoin réalise l'exploit de créer un équivalent a l'argent liquide mais numériquement.

2.1.2 Ethereum

Ethereum est une technologie qui révolutionne la blockchain en y ajoutant une fonctionnalité très intéressante nommé les smarts contract. Cette technologie est toujours basée sur le même principe que Bitcoin afin de valider ses transactions mais ajoute quelques nouvelles fonctionnalités.

2.2 Smart contract

Les contrats intelligents ou smart contracts sont des protocoles permettant d'exécuter du code de manière distribuée dans un environnement blockchain. Ce mécanisme permet de s'affranchir de l'architecture client / serveur mais aussi, des architectures distribuées traditionnels. Les développeurs de tel smart contract peuvent assumer que leurs contract une fois déployé sur la blockchain est immuable et décentralisé.

L'immuabilité de tel contrat permet de gagner la confiance des utilisateurs. Un utilisateur peut regarder le code du contrat déployé est être sûr de son comportement. Cela permet par exemple de programmer des jeux d'argents de manière numérique sans tierce de confiance mathématiquement vérifiable juste.

2.2.1 EVM

Pour créer une tel technologie Ethereum utilise l'Ethereum Virtual Machine (EVM) cette machine virtuelle peut être comparé à la JVM mais de manière distribuée. L'EVM est une machine basée sur la pile distribuée. L'EVM est exécutée sur chaque nœud du réseau Ethereum. Et chaque exécution est signé de la même manière que Bitcoin afin de protéger contre la falsification ou la modification d'une exécution.

2.2.2 Solidity

Solidity est un langage de programmation orienté objet dédié à l'écriture de contrats intelligents. Il est utilisé pour implémenter des smarts contrat sur diverses blockchains, notamment Ethereum. Solidity est un langage inspiré des langages orientées objets ainsi que du langage Javascript. L'objectif est de rendre simple et compréhensible le développement de smart contracts à tous. Néanmoins sa simplicité peut causer des problèmes car, il n'y a pas de moyen simple de vérifier l'exactitude du programme. À la différence d'un

langage fonctionnel. Sachant que les smarts contracts représentent souvent de l'argent il est très important de faire attention au code déployé.

3 Objectifs

- Prototype de smart contract modifiable.
- Comparaison avec une bibliothèque populaire.
- Comparaison avec le standard.

4 Prototypes

- Phase de recherche.
- Comparaison avec une bibliothèque populaire.
- Comparaison avec le standard.

4.1 Phase de recherche

Contactez l'équipe de développeur de eth.

4.2 Premier prototype

4.3 Comparaison avec la bibliothèque populaire

4.4 Puissance du standard

5 Problèmes soulevés

- Comment garder la confiance des utilisateurs.
- Comment sécuriser (ACL).

5.1 Garder la confiance des utilisateurs

5.2 Systèmes de gouvernance

5.2.1 Avec un administrateur

5.2.2 Avec un système de vote

5.2.3 Multisig wallet

6 Pedagogie

- Récapitulatif des réunions.
- Organisation du transfert de connaissance.

6.1 Recapilulatif des réunions

6.2 Mon organisation

7 Ouvertures

- Ajouter du type checking dans solidity
- Nouvel opcode ?

7.1 Améliorer la manière de créer des contracts dynamiquement modifiable dans solidity

7.2 Ajout d'un système de type plus intelligent pour prévenir les erreurs

8 Conclusion

References

[1] Satoshi Nakamoto: Bitcoin white paper,
`https://bitcoin.org/bitcoin.pdf`

[2] Vitalik Buterin: Ethereum white paper,
`https://ethereum.org/en/whitepaper/`

[3] L.M Goodman. Tezos white paper,
`https://tezos.com/`

[4] OpenZeppelin,
`https://docs.openzeppelin.com/openzeppelin/`

[5] The state of smarts contract Upgrades,
`https://blog.openzeppelin.com/the-state-of-smart-contract-upgrades/`