Rapport du projet de développement d'un contrat intelligent

Étudiant
Vincent Bureau

Travail remis à
Nicolas Bourré

Date de remise
12 juin 2025

Dépôt Git : https://github.com/HydroshieldMKII/veille-technologique-420-1SH-SW.git

> [!CAUTION] Ce rapport n'est pas un conseil financier et ne doit pas être utilisé comme tel. Il est destiné à des fins éducatives et de démonstration uniquement. L'utilisation de la blockchain et des contrats intelligents comporte des risques, et il est important de faire preuve de prudence et de diligence raisonnable avant d'interagir avec des systèmes décentralisés.

Table des matières

Introduction

Les contrats intelligents représentent une avancée majeure dans le domaine de la technologie blockchain. Ces programmes auto-exécutables fonctionnent sur la blockchain et permettent d'éliminer les intermédiaires traditionnels dans de nombreux secteurs. Introduits conceptuellement par Nick Szabo en 1993, bien avant la création de Bitcoin, ils n'ont gagné en popularité qu'avec l'émergence d'Ethereum en 2015 où l'applicatif est devenu concret.

Ce projet vise à explorer le développement et l'implémentation de contrats intelligents sur la blockchain Ethereum. Mon objectif est d'en apprendre sur

cette technologie qui est au cœur de nombreuses applications décentralisées, en combinant programmation, cryptographie et concepts blockchain.

Vue d'ensemble du projet

Le projet HoneyMoney vise à démontrer la faisabilité de l'utilisation de la blockchain pour de la deFi (finance décentralisée) en démontrant les fonctionnalités et leur danger potentiel. Le tout est accessible via une interface web pour les utilisateurs et un assistant AI qui résume les actions effectuées.

Architecture technique

Le système comprend plusieurs composants interconnectés :

- Un contrat intelligent ERC20 déployé sur blockchain locale Ethereum
- Une interface utilisateur Angular avec intégration MetaMask
- Un système d'analyse IA utilisant Claude d'Anthropic
- Un workflow automatisé via n8n pour le traitement des événements
- Un visualiseur de blockchain intégré

Implémentation du contrat intelligent

Le contrat intelligent HoneyMoney est un jeton ERC20 basé sur la blockchain Ethereum. Le contrat implémente la librairie OpenZeppelin pour garantir la sécurité et la conformité aux standards ERC20. Il permet de gérer des fonds de manière décentralisée, avec des fonctionnalités telles que le transfert de fonds, la création et la suppression de fonds.

Gestion du jeton

Les actions possibles avec le jeton HoneyMoney sont les suivantes :

- Transfert de fonds entre utilisateurs
- Création de fonds par l'administrateur
- Suppression de fonds par l'administrateur
- Blacklistage d'utilisateurs pour empêcher les toutes actions envers l'utilisateur blacklisté

blacklist

Transfert de fonds

En utilisant le contrat HoneyMoney, les utilisateurs peuvent transférer des fonds entre eux de manière sécurisée et transparente. Le contrat gère les soldes des utilisateurs et assure que les transferts respectent les règles de la blockchain. Il vérifie si l'un des deux utilisateurs est bloqué, et si c'est le cas, il n'effectue pas le transfert. Il vérifie également si l'utilisateur a

suffisamment de fonds pour effectuer le transfert, et si c'est le cas, il met à jour les soldes des deux utilisateurs.

Création de fonds

L'administrateur du contrat peut créer de nouveaux fonds en émettant des jetons ERC20. Cette fonctionnalité permet de réguler la quantité de fonds disponibles dans le système ou de résoudre d'éventuelles litiges en créant des fonds supplémentaires pour les utilisateurs concernés. La création de fonds est limitée à l'administrateur du contrat, garantissant ainsi un contrôle centralisé sur cette action.

Cette action est indisponible si l'adresse cible est blacklistée.

Suppression de fonds

Similaire à la création de fonds, l'administrateur du contrat peut également supprimer des fonds. Cette fonctionnalité est utile pour gérer les erreurs ou les abus dans le système. La suppression de fonds est également limitée à l'administrateur du contrat, garantissant ainsi un contrôle centralisé sur cette action.

Cette action est indisponible si l'adresse cible est blacklistée.

Mise en pause du contrat

L'administrateur du contrat peut mettre le contrat en pause, ce qui empêche toutes les actions de transfert de fonds, de création/suppression de fonds et blacklistage. Cette fonctionnalité est utile pour gérer les situations d'urgence ou les problèmes de sécurité. Lorsque le contrat est en pause, les utilisateurs ne peuvent pas effectuer d'actions jusqu'à ce que l'administrateur le réactive.

Mécanismes de sécurité

Le contrat ajoute une fonctionnalité de gestion des utilisateurs, permettant de limiter les actions indiquées précédemment à des utilisateurs spécifiques que chacun peut gérer. La librairie OpenZeppelin fournit des événements pour notifier et déclencher des actions automatisées, garantissant la traçabilité et la transparence des opérations.

Composants du système

HoneyMoney Analyser

Une intelligence artificielle (Claude de Anthropic) est intégrée pour résumer les actions effectuées et analyser si des transactions suspectes ont été effectuées, les résultats sont publiés sur la plateforme discord. La librairie OpenZeppelin et le contrat solidity fournissent des événements pour notifier et déclencher un webhook n8n. Cette action déclenche un workflow qui

reçoit les événements du contrat, les envoi à l'IA pour analyse, génère un fichier HTML résumant la transaction et les envois sur la plateforme discord.

n8n-flow

Workflow n8n

transaction1 transaction2

Résumé de la transaction HTML

claude-recap-1 claude-recap-2

Résumé de la transaction par Claude

Interface utilisateur

L'interface utilisateur est construite avec Angular et tailwinds + spartan/NG et permet aux utilisateurs d'interagir avec le contrat HoneyMoney. Elle offre une vue d'ensemble des soldes, des utilisateurs, des transactions récentes et des actions possibles. Les utilisateurs peuvent se connecter à leur portefeuille Ethereum via MetaMask pour interagir avec le contrat.

honey-money-ui

Présentation intégrée sur le fonctionnement de la blockchain

L'interface utilisateur inclut une présentation interactive intégrée qui explique le fonctionnement de la blockchain, les transactions, les blocs et les contrats intelligents. Cette présentation est conçue pour aider les utilisateurs à comprendre comment fonctionne la blockchain à l'aide de reveal.js

Leaderboard

L'interface utilisateur affiche un leaderboard des 10 utilisateurs ayant le plus haut soldes de jetons HoneyMoney. Cela permet aux utilisateurs de voir qui détient le plus de fonds dans le système et d'encourager la compétition entre les utilisateurs.

leaderboard

Visualisation des blocs sur la blockchain

Étant donné que le contrat est déployé sur une blockchain locale, les plateformes comme Etherscan ne sont pas disponibles. Le navigateur web charge les blocs de la blockchain locale et les affiche dans une interface utilisateur. Les utilisateurs peuvent modifier un bloc temporairement pour simuler de fausses transactions, mais ces modifications ne sont pas enregistrées sur la blockchain réelle. Cela permet de visualiser

l'immutabilité des transactions et de comprendre comment les blocs sont liés entre eux.

blockchain-viewer

Processus de développement

Méthodologie

Le projet a été réalisé en plusieurs étapes :

1. **Recherche et exploration** : Étude des concepts de la blockchain, des contrats intelligents et des outils de développement.
2. **Développement du contrat** : Création du contrat intelligent en utilisant Solidity et intégration des fonctionnalités souhaitées.
3. **Tests et validation** : Déploiement du contrat sur une blockchain locale (Hardhat) et réalisation de tests pour s'assurer de son bon fonctionnement.
4. **Intégration de l'IA** : Mise en place d'un système d'analyse des transactions à l'aide d'une intelligence artificielle.
5. **Développement de l'interface utilisateur** : Création d'une interface web pour interagir avec le contrat intelligent.

Défis techniques et solutions

Déploiement du contrat avec Hardhat sur un domaine Configuration complexe pour exposer le nœud blockchain local via un proxy inverse et gestion des CORS.

Configuration d'un wallet MetaMask pour interagir avec le contrat Mise en place du réseau local dans MetaMask et gestion des clés privées pour les comptes de test.

Configuration de n8n pour recevoir les événements du contrat et déclencher des actions Implémentation d'un système de webhooks pour capturer les événements blockchain et les traiter automatiquement.

Configuration des scripts (Disable automine) Gestion manuelle de la création de blocs pour simuler un environnement plus réaliste.

Gestion des array en Solidity Optimisation des structures de données pour réduire les coûts de gas et améliorer les performances.

Récupération des blocs complets sur la blockchain Développement d'une interface pour interroger directement le nœud Ethereum et extraire les données des blocs.

Résultats et analyse

Résultats fonctionnels

Un contrat intelligent fonctionnel sur la blockchain locale. Le contrat est capable de gérer les transferts de fonds et d'appliquer des règles de gestion des utilisateurs. Les résultats obtenus correspondent aux attentes initiales en termes de fonctionnalités et de règles de sécurité du contrat intelligent.

Évaluation des performances

L'effort investi dans le projet était conforme aux prévisions. La majorité du temps a été consacrée à la compréhension des concepts de la blockchain, à l'apprentissage de Solidity et à la mise en place de l'environnement de développement.

Évaluation de sécurité

Le contrat utilise les standards OpenZeppelin éprouvés, implémente des mécanismes de pause d'urgence et intègre un système de surveillance automatisé via IA pour détecter les activités suspectes.

Évaluation critique

Approches alternatives

Les résultats auraient pu être améliorés si le contrat avait été déployé sur une blockchain publique comme Ethereum ou Polygon, plutôt que sur une blockchain locale. Cela aurait permis de tester les interactions avec un réseau réel et d'utiliser des outils comme Etherscan pour visualiser les transactions et les blocs. Cependant, cela aurait également ajouté de la complexité au projet et des coûts réels liés aux frais de transaction.

Un autre chemin possible aurait été d'utiliser ganache et remix IDE pour simuler une blockchain locale, mais cela aurait limité la portée du projet. Le déploiement aurait été limité de manière manuelle sans aucune configuration possible et ganache n'est pas du tout axée sur le développement.

Comparaison avec des projets similaires

Il existe de nombreux projets similaires comme Bitcoin, Monero et d'autres blockchains qui utilisent différentes méthodes que les contrats intelligents pour gérer les transactions et les données.

Analyse des risques

Le projet remet en question l'utilisation de la blockchain pour la finance décentralisée en démontrant les fonctionnalités et leur danger potentiel. Il

met en évidence les risques associés à l'utilisation de la blockchain pour la gestion des fonds, notamment la possibilité de transactions frauduleuses et d'abus de pouvoir par les administrateurs. Lorsque le contrat est bien développé, il peut être un outil très puissant pour la gestion des fonds et des autres applications nécessitant une confiance décentralisée.

Conclusion

Le projet HoneyMoney démontre la faisabilité de l'utilisation de la blockchain pour la finance décentralisée (deFi) en fournissant une interface utilisateur intuitive et des fonctionnalités de gestion des fonds. Le contrat est sécurisé et conforme aux standards ERC20, et l'intégration de l'intelligence artificielle permet d'analyser les transactions et de détecter les activités suspectes.

Réalisations techniques

Le projet a permis de créer un écosystème complet comprenant un contrat intelligent sécurisé, une interface utilisateur moderne, un système de surveillance automatisé et des outils éducatifs interactifs.

Valeur éducative

Le projet offre une présentation interactive pour aider les utilisateurs à comprendre le fonctionnement de la blockchain et démontre concrètement les concepts théoriques de la finance décentralisée.

Considérations futures

L'architecture développée pourrait être étendue pour inclure des fonctionnalités DeFi plus avancées comme les pools de liquidité, les prêts décentralisés ou l'intégration avec d'autres protocoles blockchain.

Médiagraphie

Ethereum development environment for professionals by Nomic Foundation. Hardhat. (n.d.). https://hardhat.org/

Solidity. Solidity. (n.d.). https://docs.soliditylang.org/en/v0.8.30/

A powerful Workflow Automation Tool. n8n. (n.d.). https://n8n.io/

Nginx proxy manager. Nginx Proxy Manager. (n.d.). https://nginxproxymanager.com/

Connect, protect, and build everywhere. Cloudflare. (n.d.). https://www.cloudflare.com/

Solidity Programming Language. (n.d.). https://soliditylang.org/

Wikimedia Foundation. (2025, May 22). Smart contract. Wikipedia. https://en.wikipedia.org/wiki/Smart_contract