Mini-projet : Entreprise 4.0 (Blockchain)

Rapport de Mise en Place d'une Blockchain avec Quorum

Auteurs :
HNID Meyssam
DAMIENS Lana

Enseignant :
LEZOCHE Mario

A rendre pour le samedi 25 mai 2024 (Année 2023-2024)

Table des matières

Chapitre 1

Introduction à la Blockchain

La blockchain est une technologie révolutionnaire qui permet de créer un registre décentralisé, sécurisé et immuable de transactions. En d'autres termes, c'est une chaîne de blocs où chaque bloc contient des informations sur les transactions et est cryptographiquement lié au bloc précédent. Cette structure rend les données sur une blockchain résistantes aux modifications et garantit leur intégrité. Utilisée initialement pour le Bitcoin, la blockchain a rapidement trouvé des applications dans de nombreux autres domaines, tels que la finance, la gestion de la chaîne d'approvisionnement, et plus récemment, dans le développement de contrats intelligents.

Les contrats intelligents sont des programmes autonomes qui s'exécutent automatiquement lorsque des conditions prédéfinies sont remplies, ouvrant la voie à des transactions automatisées et sécurisées sans nécessiter d'intermédiaire. Cela permet de réduire les coûts, d'augmenter l'efficacité et de diminuer les risques de fraude. En entreprise, la blockchain peut transformer la manière dont les données sont échangées et gérées, offrant une transparence accrue et une meilleure traçabilité.

Dans ce rapport, nous allons détailler les étapes de mise en place d'une blockchain privée en utilisant Quorum, une version d'Ethereum adaptée aux besoins des entreprises. Nous couvrirons l'installation de Quorum, le lancement du réseau, l'interaction avec l'interface graphique, et le déploiement d'un Smart Contract.

Chapitre 2

Mise en Place de la Blockchain avec Quorum

2.1 Prérequis

Avant de commencer l'installation, nous nous sommes assurés d'avoir les prérequis suivants :

- Node.js et NPM version 14 ou supérieure
- Docker et Docker-compose
- Hardhat development framework
- Solc
- Commande `curl`
- MetaMask

2.2 Installer Quorum

Pour commencer, nous avons installé Quorum, une version de la blockchain Ethereum conçue pour les applications d'entreprise. Quorum se distingue par ses fonctionnalités de gestion des permissions et de confidentialité des transactions, essentielles pour les entreprises souhaitant utiliser la technologie blockchain de manière sécurisée et conforme à leurs besoins.

Fonctionnalité gestion des permissions

L'une des fonctionnalités principales de Quorum est la gestion des permissions. Contrairement à Ethereum public où tout le monde peut rejoindre et participer au réseau, Quorum permet de créer des environnements blockchain privés. Ainsi, seules les entités approuvées peuvent rejoindre le réseau et interagir avec lui. Cette fonctionnalité est essentielle pour les entreprises qui doivent contrôler avec précision qui peut accéder aux données et aux fonctionnalités du réseau, garantissant ainsi une meilleure sécurité et conformité.

Confidentialité des transactions

La confidentialité est une autre caractéristique clé de Quorum. Dans une blockchain publique comme Ethereum, toutes les transactions sont visibles par tous les participants du réseau. Cepen-

dant, pour les entreprises, il est souvent crucial de garder certaines transactions confidentielles. Quorum propose ainsi des mécanismes de confidentialité avancés tels que les transactions privées. Ces transactions permettent à deux parties de réaliser une transaction sans que les détails de cette dernière ne soient divulgués au reste du réseau. Cela est rendu possible grâce à des techniques cryptographiques avancées et à l'utilisation de gestionnaires de clés qui assurent que seules les parties concernées peuvent voir et vérifier les informations échangées.

Lien d'installation de Quorum

2.3 Lancer le réseau

Une fois Quorum installé, la prochaine étape consiste à lancer le réseau. Cette opération implique la configuration et le démarrage des différents nœuds qui constitueront le réseau. Pour ce faire, on exécute le script `run.sh`, qui configure et démarre les nœuds en question, nécessaires pour faire fonctionner le réseau blockchain. Ce script lance un processus automatisé qui simplifie cette tâche à l'origine très complexe.

```
./run.sh
```

2.4 Visualiser l'interface graphique

Après avoir démarré le réseau Quorum avec succès, nous pouvons visualiser et interagir avec le réseau via une interface graphique accessible par un navigateur web. Cette interface offre une vue complète et intuitive du réseau, facilitant sa surveillance. Autrement dit, il est dorénavant possible de monitoire l'état du réseau, les transactions, ainsi que les contrats intelligents.

Voici quelques unes des fonctionnalités principales de l'interface graphique qui permettent le monitoring du réseau en temps réel :

Tableau de bord

Le tableau de bord offre une vue d'ensemble du réseau, incluant le nombre de nœuds actifs, le nombre de blocs créés, et le volume des transactions. Cela permet aux administrateurs de vérifier rapidement la santé et la performance du réseau.

Noeuds

Chaque nœud du réseau peut être surveillé individuellement. L'interface graphique affiche des détails tels que l'utilisation des ressources (CPU, mémoire), l'état de synchronisation, et les logs des nœuds. Cela aide à identifier et résoudre les problèmes potentiels rapidement.

Transactions

Les utilisateurs peuvent voir une liste de toutes les transactions qui ont eu lieu sur le réseau. Chaque transaction est accompagnée de détails tels que l'adresse de l'expéditeur et du destinataire, la valeur transférée, et l'état de la transaction (confirmée, en attente, etc.). Directement depuis l'interface, les utilisateurs peuvent également soumettre de nouvelles transactions. Cela inclut la spécification des paramètres nécessaires tels que les adresses des parties impliquées, le montant, et toute donnée additionnelle requise pour la transaction.

Contrats

Les utilisateurs peuvent déployer de nouveaux contrats intelligents sur le réseau en utilisant l'interface. Le processus de déploiement est simplifié grâce à des formulaires intuitifs où les utilisateurs peuvent entrer le code du contrat, les paramètres d'initialisation, et d'autres configurations nécessaires. Une fois déployés, les contrats intelligents peuvent être gérés et appelés directement depuis l'interface. Les utilisateurs peuvent exécuter les fonctions disponibles dans les contrats, passer des paramètres, et visualiser les résultats des exécutions en temps réel.

Contrôle d'accès

Seuls les utilisateurs autorisés peuvent accéder à l'interface graphique. Les administrateurs peuvent gérer les rôles et permissions, assurant que chaque utilisateur a accès uniquement aux fonctionnalités nécessaires pour son rôle spécifique.

URL de l'interface graphique

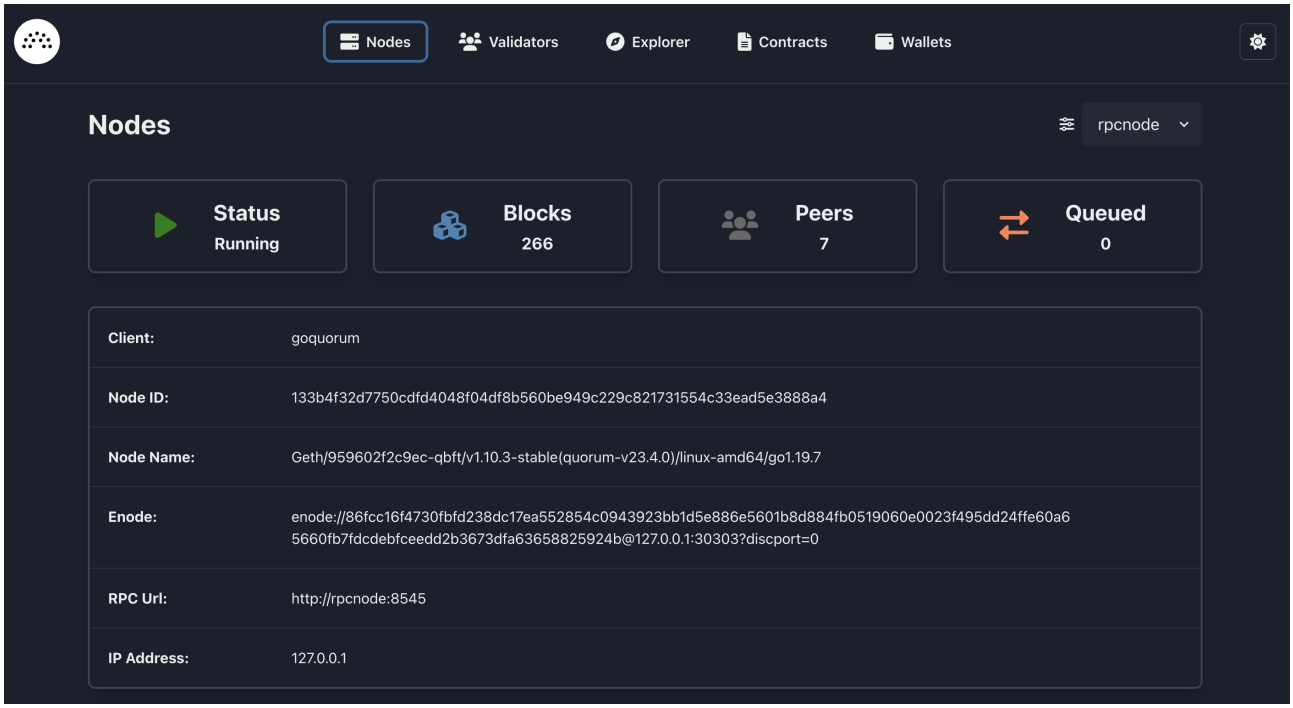

FIGURE 2.1 – Interface graphique

Chapitre 3

Déployer un Smart Contract

Dans cette étape, nous allons mettre en place un Smart Contract simple pour stocker une valeur numérique, avec une contrainte spécifique : la dernière chiffre de la valeur doit être impair.

3.1 Code pour le Smart Contract

Voici le code du Smart Contract :

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

contract LastNumberOddSimpleStorage {
    uint public storedData;
    event stored(address _to, uint _amount);

    constructor(uint initVal) public {
        emit stored(msg.sender, initVal);
        storedData = initVal;
    }

    function set(uint x) public {
        require(x % 10 % 2 == 1, "Last digit of the value
        must be odd.");
        emit stored(msg.sender, x);
        storedData = x;
    }

    function get() view public returns (uint retVal) {
        return storedData;
    }
}
```

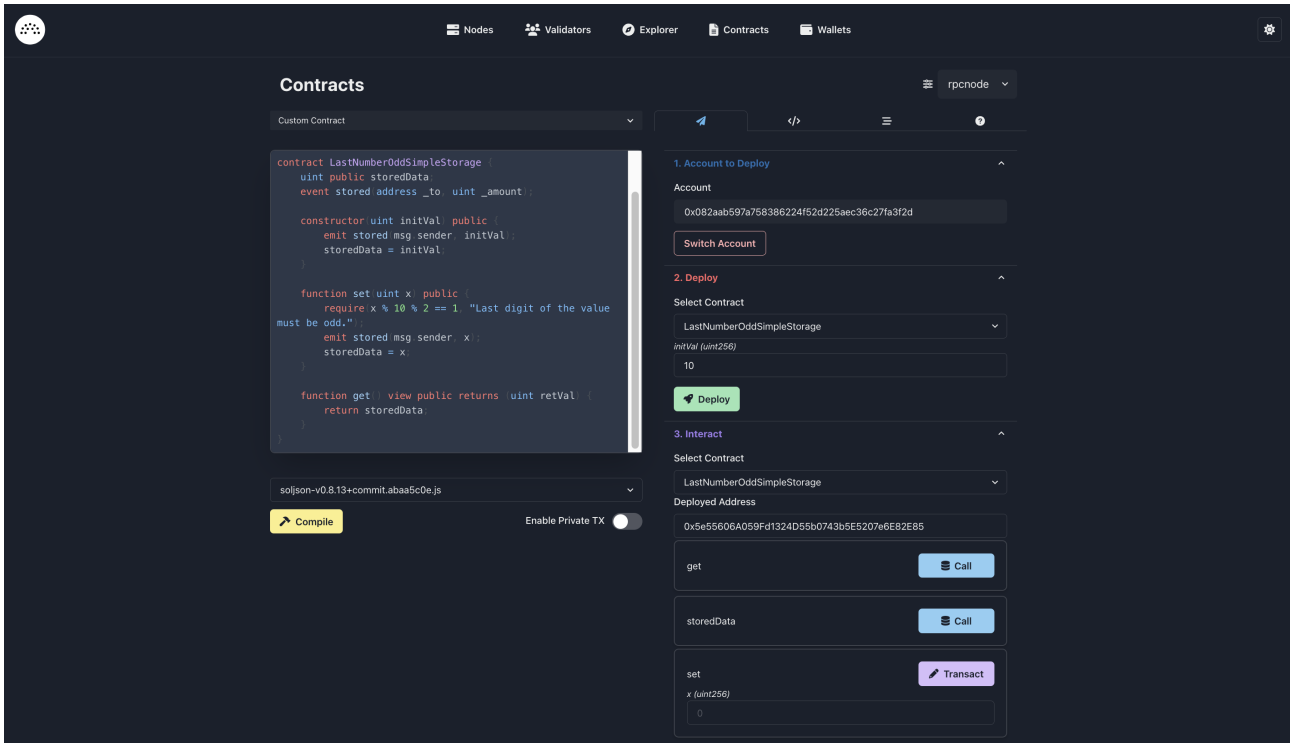

FIGURE 3.1 – Interface graphique : Partie Contract

3.2 Explication du code ligne par ligne

SPDX-License-Identifier : Indique la licence du code pour des raisons de conformité.

pragma solidity : Spécifie la version du compilateur Solidity utilisée, assurant la compatibilité.

LastNumberOddSimpleStorage : Nom du contrat, décrivant sa fonctionnalité

storedData : Une variable publique qui stocke la valeur numérique.

event stored : Un événement émis chaque fois que la valeur storedData est mise à jour.

constructor : Fonction exécutée lors du déploiement du contrat, initialisant storedData avec une valeur donnée.

require : Vérifie que le dernier chiffre de initVal est impair. Sinon, le déploiement échoue avec un message d'erreur.

set : Met à jour storedData avec une nouvelle valeur x, après vérification que le dernier chiffre est impair.

require : Vérifie la contrainte avant de mettre à jour storedData.

get : Retourne la valeur actuelle de storedData.

3.3 Se connecter avec MetaMask

Pour interagir avec notre Smart Contract, nous devons nous connecter à notre réseau Quorum via MetaMask, une extension de navigateur qui permet de gérer des comptes Ethereum et d'interagir avec des dApps (applications décentralisées).

1. Ouvrez MetaMask et connectez-vous.
2. Allez dans les paramètres de MetaMask (*Settings*).
3. Sous l'onglet *Networks*, cliquez sur *Add Network*.
4. Remplissez les informations réseau avec les détails de votre réseau Quorum.

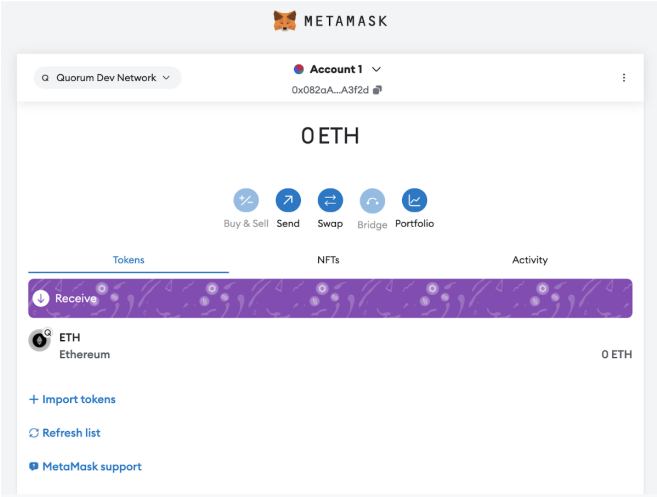

FIGURE 3.2 – Metamask

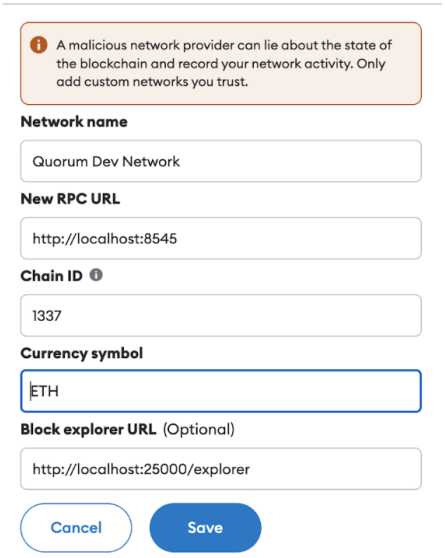

FIGURE 3.3 – Lier Metamask avec Quorum

3.4 Déploiement du Smart Contract

Une fois MetaMask configuré pour se connecter à notre réseau Quorum, nous pouvons déployer notre Smart Contract depuis l'interface graphique de Quorum.

1. Allez dans le menu *Contracts* de l'interface graphique de Quorum.
2. Sélectionnez *Deploy Contract* et collez le code du Smart Contract dans l'éditeur.
3. Configurez les paramètres de déploiement (comme la valeur initiale `initVal`).
4. Cliquez sur *Deploy* et confirmez la transaction dans MetaMask.

Voici ce que nous avons obtenu :

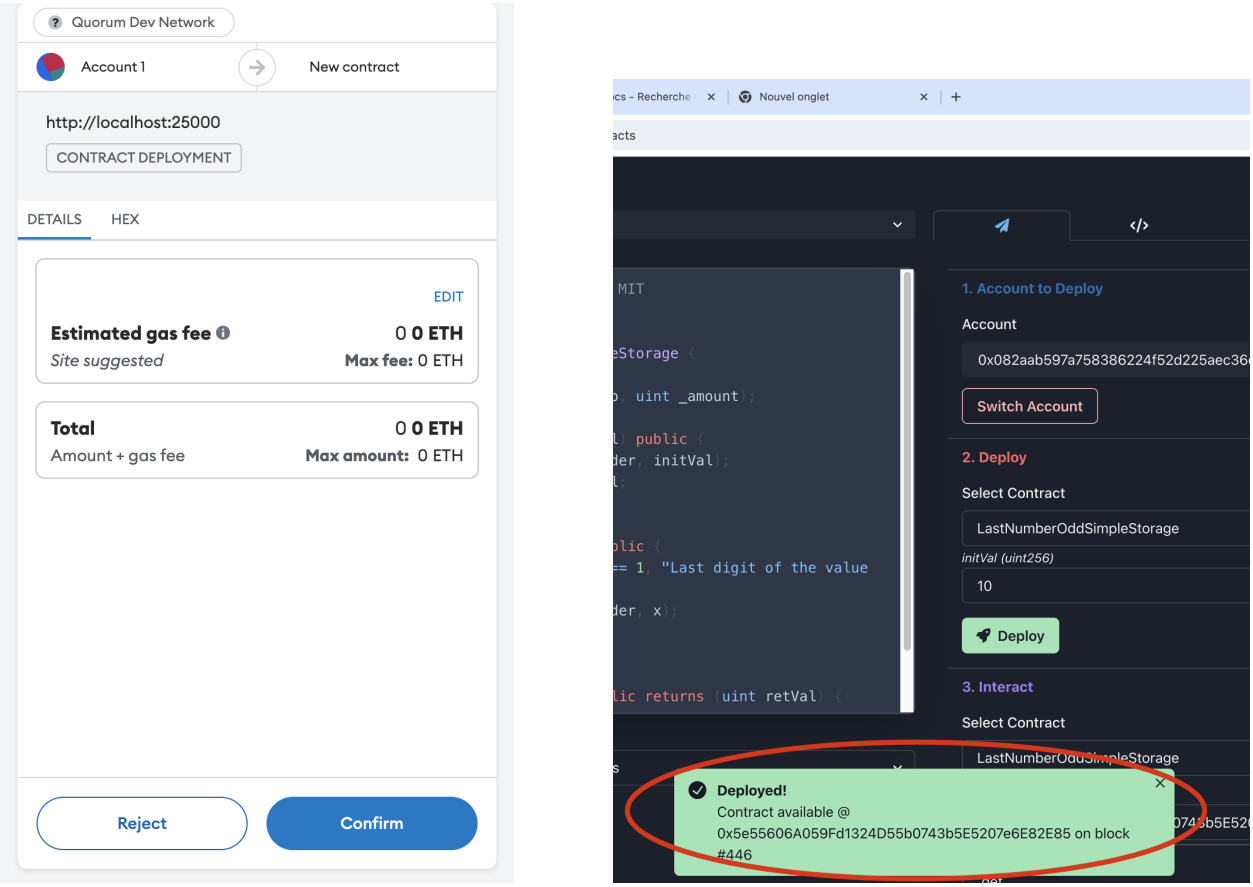

FIGURE 3.6 – Déploiement du contract

Chapitre 4

Conclusion

En suivant les étapes décrites dans ce rapport, nous avons réussi à mettre en place un réseau blockchain avec Quorum, à visualiser l'interface graphique et à déployer un Smart Contract simple. Ce processus a démontré la puissance et la flexibilité de Quorum en tant que solution blockchain adaptée aux besoins des entreprises.

L'installation et la configuration de Quorum ont été relativement simples, et l'interface graphique a fourni une manière intuitive de gérer le réseau et les contrats intelligents. Le déploiement du Smart Contract a permis de mettre en évidence les capacités de Quorum en matière de gestion des permissions et de confidentialité des transactions.

Malgré quelques défis techniques, tels que la configuration correcte de l'URL RPC et l'intégration avec MetaMask, ces étapes nous ont permis d'acquérir une compréhension approfondie des principes fondamentaux de la blockchain et des contrats intelligents.

La blockchain et les contrats intelligents représentent une avancée majeure dans la façon dont les entreprises peuvent gérer les données et les transactions. En utilisant Quorum, nous avons pu explorer ces technologies de manière pratique et tangible, ouvrant la voie à des applications plus avancées et innovantes à l'avenir.

Annexe : Emplacement du Travail

Vous pouvez retrouver notre travail sur la page GitHub que nous avons créée pour le projet à l'adresse suivante :

`https://github.com/Mys-Hn/SIE-EN4-Blockchain-EN4-HNID-DAMIENS`

Nous avons également déposé un fichier zip comportant ce rapport et le code du travail sur le dépôt Arche à l'adresse suivante :

`https://arche.univ-lorraine.fr/mod/assign/view.php?id=1894502`

Merci de consulter ces emplacements pour accéder à l'ensemble des fichiers et du code source du projet.