كلية العلوم و التقنيات بطنجة
ⵜⴰⵖⵣⵍⵓⵏⵜ ⵏ ⵜⵜⵥⵏⵓⵍⵓⵊⵉⵜ ⴰ ⵜⵓⵙⵙⵉⵔⵜ ⵏ ⵟⵟⵏⵊⴰ
Faculté des Sciences et Techniques de Tanger

AIDS
MASTER'S DEGREE IN ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

‘SMART CONTRACT "MINI RESEAU SOCIAL" EN SOLIDITY’

Préparé par :
Yasmina Hachhouch

Introduction

Cet atelier est un travail applicatif pour mettre en pratique la création de smart contracts en Solidity. Nous avons utilisé l'IDE Remix pour écrire, compiler, et déployer notre contrat intelligent, nommé **MiniSocial**, et MetaMask pour l'interaction sur un réseau de test. Le but de ce projet est de simuler une mini-plateforme de réseaux sociaux qui permet aux utilisateurs de publier des messages, de consulter les messages des autres, de les "liker", et de récupérer des informations spécifiques.

Description du Smart Contract

Le smart contract **MiniSocial** est structuré de manière à gérer les interactions sociales de base. Voici une description des principaux éléments :

1. **Déclaration du Contrat** Le contrat commence par la spécification de la licence et de la version de Solidity. Nous avons ensuite déclaré le contrat avec une structure appelée Post pour définir les propriétés d'un message. Chaque post contient un identifiant unique, un message de type texte, l'adresse de l'auteur, et un compteur pour les likes.
2. **Événement NewPostEvent** Un événement est utilisé pour signaler la publication de chaque nouveau message, incluant des informations telles que le contenu du message, l'auteur, l'horodatage, et l'identifiant du post. Cela permet une meilleure transparence et facilite la gestion des notifications.
3. **Stockage des Messages** Un tableau dynamique est utilisé pour stocker tous les messages publiés. Cette approche permet de conserver les posts dans un ordre structuré et d'y accéder facilement lors de la récupération des données.
4. **Fonctionnalités Principales**
    - **Publication de Messages** : Une fonction vérifie que le message n'est pas vide et qu'il ne dépasse pas 140 caractères. Une fois validé, le message est ajouté au tableau avec l'adresse de l'auteur et un compteur de likes initialisé à zéro.
    - **Consultation des Messages** : Des fonctions sont fournies pour récupérer un message particulier, l'auteur d'un message, ou le nombre total de messages publiés. Ces fonctions permettent aux utilisateurs de consulter efficacement les informations stockées.
    - **Ajout de Likes** : Une fonction permet d'incrémenter le nombre de likes d'un message. Cela permet aux utilisateurs d'interagir avec le contenu, simulant ainsi l'engagement typique des plateformes de médias sociaux.

Déploiement et Tests

1. **Déploiement et Tests en Environnement Virtuel (VM)** Le contrat a été testé avec succès dans l'environnement virtuel de Remix (VM), où toutes les fonctionnalités, y compris la publication de messages, la récupération d'informations, et l'ajout de likes, ont parfaitement fonctionné. Chaque opération a donné les résultats attendus, et le contrat a démontré sa fiabilité en simulation.
2. **Déploiement avec MetaMask sur le Réseau de Test**
   Lors du déploiement sur le réseau de test SepoliaETH en utilisant MetaMask, j'ai rencontré un problème spécifique. Bien que le contrat ait été correctement

compilé, chaque tentative d’interaction avec le contrat, comme la publication de messages, a échoué, renvoyant une erreur indiquant que la transaction ne pouvait pas être effectuée. L'erreur semble être liée à un manque de fonds (ETH de test) sur les comptes MetaMask. En effet, le déploiement et l'exécution de transactions sur le réseau de test nécessitent des ETH de test pour couvrir les frais de gas, et sans ces fonds, les opérations échouent systématiquement.

Conclusion

L'implémentation de ce smart contract **MiniSocial** a permis de mieux comprendre les concepts fondamentaux de la création et de la gestion de smart contracts en Solidity. Les tests effectués dans l'environnement virtuel de Remix (VM) ont montré que le contrat fonctionnait parfaitement, offrant toutes les fonctionnalités prévues, comme la publication de messages, la récupération de contenu, et l'ajout de likes. Cependant, lors du déploiement sur le réseau de test SepoliaETH via MetaMask, des problèmes sont apparus, notamment des erreurs liées à un manque de fonds en ETH de test nécessaires pour exécuter les transactions. Ce projet a souligné l'importance de la préparation des ressources adéquates, comme des ETH de test, pour garantir le succès des tests sur un réseau de blockchain. Dans l'ensemble, cette expérience a offert des enseignements pratiques essentiels pour le développement de smart contracts robustes et prêts pour le déploiement réel.

4o