



Rapport de Projet Blockchain

Présenté dans le cadre de la matière

Blockchain Industrielle

À l'attention de

École Nationale d'Ingénieurs de Sfax (Ecole Nationale d'Ingénieurs de Sfax)
Département d'Informatique

Année Universitaire :

2024-2025

Réalisé par :

Naji Abderraouf

Traçabilité des Chaînes d'Approvisionnement avec Blockchain

Mr Tarek Frikha

Encadrant
académique



REMERCIEMENTS

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon projet et qui m'ont soutenu tout au long de ce travail. Sans leur aide et leur soutien, ce projet aurait été bien plus difficile à réaliser.

Je remercie tout particulièrement M. Tarek Frikha, mon encadrant académique, pour la confiance qu'il a placée en moi et pour son accompagnement précieux tout au long de ce projet. Ses conseils judicieux et son expertise m'ont permis d'avancer et de mener à bien cette réalisation.

Je remercie également l'École Nationale d'Ingénieurs de Sfax (ENIS) pour la qualité de la formation dispensée et pour avoir permis l'organisation de ce projet dans le cadre de la matière "Blockchain Industrielle".

Enfin, je tiens à exprimer ma gratitude envers le département d'Informatique de l'ENIS, ainsi que l'ensemble de l'équipe pédagogique, pour leur soutien et leurs encouragements durant mon parcours universitaire.



TABLE DES MATIÈRES

LISTE DES FIGURES	iv
LISTE DES ABRÉVIATIONS	vi
INTRODUCTION GÉNÉRALE	1
1 Cadre Général du Projet	3
1.1 Problématiques Identifiées	3
1.2 Solutions Proposées	4
2 Analyse et Spécification des Besoins	6
2.1 Besoins Fonctionnels	7
2.2 Besoins Non Fonctionnels	8
3 Architecture et Outils Utilisés	9
3.1 Architecture Globale	10
3.2 Outils Techniques	11
3.2.1 Hedera Hashgraph	11
3.2.2 Web3.js	12
3.2.3 MetaMask	12
3.2.4 MongoDB	13
3.2.5 Angular, Flutter et Postman	14
4 Réalisation	15
4.1 Développement et Déploiement	15
4.2 Fonctionnalités Implémentées	17
4.3 Résultats Obtenus	18
CONCLUSION GÉNÉRALE	20
4.4 Conclusion et Perspectives	20
4.4.1 Résumé des bénéfices obtenus grâce à la solution blockchain	20

TABLE DES MATIÈRES

4.4.2	Limites du projet actuel	20
4.4.3	Suggestions pour des travaux futurs	21
BIBLIOGRAPHIE		22



LISTE DES FIGURES

3.1	Schéma de l'Architecture Globale	10
3.2	Logo de Hedera Hashgraph.	11
3.3	Logo de Web3.js.	12
3.4	Logo de MetaMask.	12
3.5	Logo de MongoDB.	13
3.6	Logos d'Angular, Flutter et Postman.	14
4.1	Processus de développement des smart contracts.	15
4.2	Processus de déploiement des smart contracts.	16
4.3	Processus de déploiement des smart contracts.	17
4.4	Interface utilisateur pour le suivi des transactions.	18
4.5	Notification d'une transaction non conforme.	19



Liste des tableaux



LISTE DES ABRÉVIATIONS

YAML Yet Another Markup Language

Yum Yellowdog Updater, Modified

Apt Advanced Package Tool

K8s Kubernetes

RPM Red Hat Package Manager

CMS Content Management System

CPU Central Processing unit

SVN Subversion

MB Mega Byte

GB Gega Byte

VM Virtual Machine

DNS Domain Name System

DAST Dynamic application security testing



INTRODUCTION GÉNÉRALE

Dans un contexte technologique en constante évolution, la blockchain s'est imposée comme une solution innovante pour relever les défis liés à la fiabilité, la traçabilité et la sécurité des données. Initialement développée pour répondre aux besoins du secteur financier, cette technologie décentralisée trouve désormais des applications dans divers domaines, notamment l'industrie, la logistique, et la gestion des chaînes d'approvisionnement. Grâce à ses caractéristiques d'immutabilité, de transparence et d'auditabilité, la blockchain offre des solutions pertinentes aux problématiques modernes.

Cependant, malgré ses nombreux avantages, la mise en œuvre de la blockchain dans un environnement industriel soulève plusieurs défis. Parmi eux figurent la complexité des systèmes existants, le manque d'interopérabilité entre les solutions, ainsi que les contraintes liées au traitement et au stockage de grandes quantités de données sensibles. Ces problématiques limitent l'adoption généralisée de cette technologie dans des secteurs critiques.

Dans le cadre de ce projet, nous proposons une solution innovante basée sur **Hedera Hashgraph**, une technologie de registre distribué qui se distingue par sa scalabilité et sa rapidité. Combinée à des outils tels que **MetaMask** pour la signature numérique et **MongoDB** pour le stockage des données non sensibles, cette solution vise à répondre aux besoins spécifiques de l'industrie. En exploitant une architecture modulaire et des outils modernes comme **Web3.js**, nous nous concentrons sur la traçabilité des transactions, la sécurisation des données, et l'intégration avec des systèmes existants.

L'objectif de ce projet est de démontrer comment une solution blockchain industrielle peut optimiser les processus de gestion des chaînes d'approvisionnement tout en garantissant une efficacité accrue et une sécurité renforcée. Ce rapport présente l'analyse des problématiques,

les besoins fonctionnels et non fonctionnels, ainsi que la conception et la mise en œuvre de la solution proposée.

En conclusion, ce projet s'inscrit dans une perspective d'amélioration continue et d'innovation pour répondre aux exigences croissantes des secteurs industriels. Nous espérons que les résultats obtenus fourniront une base solide pour des développements futurs et favoriseront une adoption plus large de la blockchain dans des applications critiques.

Chapitre

1

Cadre Général du Projet

1.1 Problématiques Identifiées

Dans un environnement où les chaînes d’approvisionnement deviennent de plus en plus complexes, plusieurs défis majeurs subsistent quant à l’utilisation de technologies traditionnelles pour gérer les transactions et les données. Voici les principales problématiques identifiées :

Fiabilité et traçabilité des transactions

Les systèmes actuels souffrent d’un manque de fiabilité dans le suivi des transactions. Les informations peuvent être modifiées ou perdues, ce qui réduit la confiance des parties prenantes dans les processus existants.

Manque d’interopérabilité entre systèmes

Les différentes entités impliquées dans une chaîne d’approvisionnement utilisent souvent des systèmes hétérogènes qui ne communiquent pas efficacement entre eux. Ce manque d’interopérabilité entraîne des retards, des erreurs et une inefficacité globale.

Sécurisation des données sensibles

Les solutions classiques présentent des vulnérabilités en matière de protection des données, ce qui peut exposer des informations critiques à des accès non autorisés, des altérations ou des pertes.

Complexité des solutions existantes

Les solutions blockchain actuelles sont souvent complexes à mettre en œuvre et nécessitent une expertise technique avancée. Cette complexité peut décourager les entreprises industrielles d'adopter ces technologies, malgré leurs avantages potentiels.

1.2 Solutions Proposées

Pour répondre aux problématiques identifiées, nous avons conçu une solution basée sur des technologies modernes et efficaces, adaptées aux besoins spécifiques des chaînes d'approvisionnement industrielles. Les solutions proposées sont les suivantes :

Utilisation de Hedera Hashgraph pour la transparence et l'immuabilité

Hedera Hashgraph est une technologie de registre distribué qui garantit l'immuabilité des données et assure une transparence totale dans les transactions. Grâce à son algorithme de consensus performant, cette solution offre une capacité de traitement parallèle et une faible latence, répondant ainsi aux besoins des environnements industriels.

Signature numérique via MetaMask

MetaMask est utilisé pour gérer les signatures numériques, ce qui renforce la sécurité et l'authenticité des transactions. Cet outil permet d'authentifier les parties prenantes de manière fiable tout en simplifiant leur interaction avec la solution blockchain.

Architecture modulaire avec Web3 et MongoDB

L'architecture de la solution repose sur une combinaison de Web3.js, pour l'interfaçage avec la blockchain, et de MongoDB, pour le stockage des données non sensibles. Cette approche

modulaire facilite l'intégration avec les systèmes existants et améliore la scalabilité de l'ensemble du système.

Résumé du Chapitre

Ce chapitre a présenté les principales problématiques rencontrées dans les chaînes d'approvisionnement, ainsi que les solutions proposées pour y remédier. En utilisant des technologies comme Hedera Hashgraph, MetaMask, Web3 et MongoDB, notre solution vise à améliorer la traçabilité, la sécurité et l'efficacité des processus industriels.

Analyse et Spécification des Besoins

Sommaire

2.1	Besoins Fonctionnels	7
2.2	Besoins Non Fonctionnels	8

2.1 Besoins Fonctionnels

Les besoins fonctionnels définissent les fonctionnalités principales que le système doit fournir pour répondre aux objectifs du projet. Voici les besoins identifiés pour la solution blockchain proposée :

Gestion et traçabilité des transactions

Le système doit permettre la gestion efficace des transactions tout en assurant leur traçabilité. Chaque transaction doit être enregistrée de manière immuable sur la blockchain, garantissant ainsi l'historique complet et transparent des échanges.

Sécurisation des données sensibles

La solution doit protéger les données sensibles des utilisateurs et des transactions. Cela inclut la garantie de la confidentialité des informations et leur protection contre les accès non autorisés ou les altérations.

Intégration avec des systèmes existants

Le système doit pouvoir s'intégrer facilement avec les infrastructures existantes des entreprises, comme les systèmes ERP (Enterprise Resource Planning) ou les bases de données déjà en place, afin de faciliter son adoption.

Stockage des données non sensibles

Pour optimiser l'utilisation de la blockchain, seules les données critiques seront stockées sur la chaîne, tandis que les données non sensibles, comme les métadonnées, seront sauvegardées dans une base de données externe comme MongoDB. Cette séparation améliore la performance globale et réduit les coûts.

2.2 Besoins Non Fonctionnels

Les besoins non fonctionnels décrivent les contraintes et les caractéristiques de qualité que le système doit respecter pour garantir son bon fonctionnement dans des conditions réelles.

Sécurité

Le système doit garantir la confidentialité, l'intégrité et la disponibilité des données et transactions. Cela inclut :

- L'authentification des utilisateurs via des mécanismes robustes tels que MetaMask.
- La protection des données contre les altérations grâce à l'utilisation de Hedera Hashgraph.
- La résilience face aux attaques malveillantes pour maintenir une disponibilité constante du service.

Scalabilité

Le système doit pouvoir gérer simultanément un grand nombre de transactions tout en maintenant de faibles temps de réponse. Cette capacité repose sur :

- L'algorithme de consensus performant de Hedera Hashgraph, qui permet un traitement parallèle des transactions.
- Une architecture modulaire qui facilite l'ajout de nouvelles fonctionnalités ou ressources en fonction des besoins.

Résumé du Chapitre

Dans ce chapitre, nous avons détaillé les besoins fonctionnels et non fonctionnels de la solution proposée. Ces spécifications guident la conception et la mise en œuvre du système pour garantir qu'il répond aux attentes des utilisateurs et aux contraintes techniques. La combinaison de ces besoins assure une solution fiable, performante et adaptable.

Architecture et Outils Utilisés

Sommaire

3.1	Architecture Globale	10
3.2	Outils Techniques	11
3.2.1	Hedera Hashgraph	11
3.2.2	Web3.js	12
3.2.3	MetaMask	12
3.2.4	MongoDB	13
3.2.5	Angular, Flutter et Postman	14

3.1 Architecture Globale

La solution proposée repose sur une architecture modulaire et distribuée, combinant la blockchain Hedera Hashgraph pour la gestion des transactions critiques et MongoDB pour le stockage des données non sensibles. Cette architecture permet d'assurer une scalabilité et une sécurité optimales tout en facilitant l'intégration avec les systèmes existants.

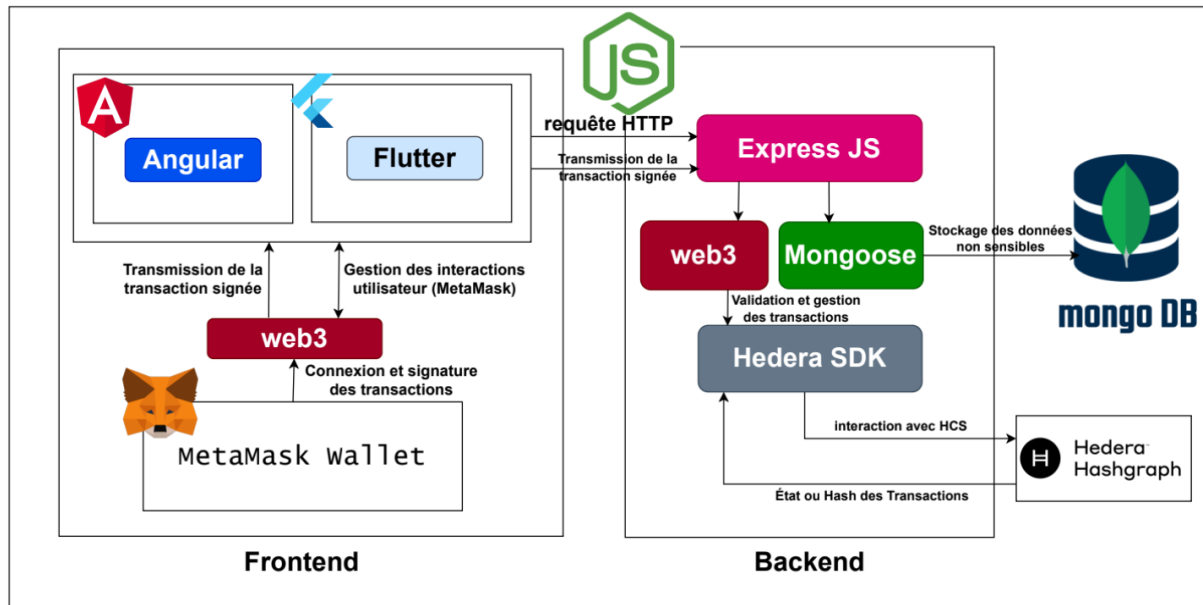


FIGURE 3.1 – Schéma de l'Architecture Globale

Comme illustré dans la figure 3.1, l'architecture comprend plusieurs couches :

- **Interface utilisateur :** Interaction via une application web sécurisée.
- **Middleware :** Utilisation de Web3.js pour la communication avec la blockchain.
- **Blockchain Hedera Hashgraph :** Gestion des transactions avec un haut niveau de fiabilité et de transparence.
- **Base de données MongoDB :** Stockage des métadonnées et des informations non sensibles.

Cette architecture est conçue pour être flexible et évolutive, répondant aux besoins des chaînes d'approvisionnement industrielles.

3.2 Outils Techniques

Pour la mise en œuvre de cette architecture, plusieurs outils ont été sélectionnés en raison de leurs performances et de leur adéquation aux besoins du projet. Voici une description des outils utilisés :

3.2.1 Hedera Hashgraph



FIGURE 3.2 – Logo de Hedera Hashgraph.

Hedera Hashgraph est une technologie de registre distribué qui offre une alternative performante aux blockchains traditionnelles. Voici les principales raisons justifiant son choix :

- **Scalabilité** : Grâce à son algorithme de consensus basé sur le gossip protocol, Hedera Hashgraph est capable de traiter des milliers de transactions par seconde.
- **Sécurité** : Le mécanisme de consensus asynchrone byzantin garantit une immuabilité totale des transactions.
- **Faible latence** : Les transactions sont confirmées en quelques secondes, ce qui est essentiel pour des applications industrielles.

3.2.2 Web3.js



FIGURE 3.3 – Logo de Web3.js.

Web3.js est une bibliothèque JavaScript qui permet de communiquer avec des blockchains compatibles avec Ethereum et des technologies associées. Dans notre projet, elle joue un rôle essentiel :

- **Interface avec la blockchain** : Permet d'interagir avec Hedera Hashgraph pour envoyer et lire des transactions.
- **Facilité d'intégration** : Sa compatibilité avec les frameworks front-end modernes facilite son intégration dans l'application.

3.2.3 MetaMask



FIGURE 3.4 – Logo de MetaMask.

MetaMask est une extension de navigateur et une application mobile utilisée pour gérer les portefeuilles numériques et signer les transactions sur une blockchain. Dans notre projet, elle est utilisée pour :

- **Gestion des identités numériques** : Les utilisateurs authentifient et signent leurs transactions à l'aide de MetaMask.
- **Sécurité** : Elle fournit une couche de protection supplémentaire en stockant les clés privées localement sur les appareils des utilisateurs.

3.2.4 MongoDB



FIGURE 3.5 – Logo de MongoDB.

MongoDB est une base de données NoSQL utilisée pour le stockage des données non sensibles dans le cadre de notre solution. Ses avantages incluent :

- **Flexibilité** : Sa structure en documents permet de stocker des données semi-structurées, adaptées à une variété de cas d'utilisation.
- **Performance** : Elle offre des temps de réponse rapides pour la gestion de grandes quantités de données.
- **Évolutivité** : MongoDB est conçu pour gérer efficacement des volumes croissants de données.

3.2.5 Angular, Flutter et Postman



FIGURE 3.6 – Logos d’Angular, Flutter et Postman.

Angular : Framework pour la création d’interfaces utilisateur dynamiques et modernes.

Flutter : Framework pour le développement d’applications mobiles multiplateformes.

Postman : Outil pour tester et automatiser les requêtes API.

Résumé du Chapitre

Ce chapitre a présenté l’architecture globale de la solution blockchain ainsi que les outils techniques utilisés. Hedera Hashgraph, Web3.js, MetaMask, MongoDB, Angular, Flutter et Postman travaillent ensemble pour fournir une solution fiable, performante, et adaptée aux besoins des chaînes d’approvisionnement industrielles.

Chapitre

4

Réalisation

4.1 Développement et Déploiement

Dans cette section, nous détaillons le processus de développement et de déploiement des smart contracts utilisés dans notre solution blockchain.

Développement des Smart Contracts

Les smart contracts ont été développés en utilisant un langage compatible avec Hedera Hashgraph. Ces contrats permettent :

- La création, la lecture et la validation des transactions.
- La gestion des utilisateurs et des autorisations.
- L'émission d'alertes en cas de transactions non conformes.

▸ Bureau ▸ BlockChainProject ▸ contracts





<input type="checkbox"/> Nom	Modifié le	Type	Taille
 EntityManager.sol	04/12/2024 4:49 PM	Fichier SOL	5 Ko
 Lock.sol	23/11/2024 12:04 AM	Fichier SOL	1 Ko
 LotManager.sol	05/12/2024 7:34 PM	Fichier SOL	7 Ko
 SupplyChainManager.sol	05/12/2024 7:37 PM	Fichier SOL	6 Ko


FIGURE 4.1 – Processus de développement des smart contracts.

Déploiement des Smart Contracts

Le déploiement des smart contracts a été réalisé via un script automatisé utilisant Web3.js.

Ce script suit les étapes suivantes :

1. Connexion à un compte administrateur via MetaMask pour autoriser le déploiement.
2. Compilation et enregistrement du smart contract sur le réseau Hedera Hashgraph.
3. Génération d'un identifiant unique pour le contrat, utilisé dans les requêtes ultérieures.
4. Tests pour valider les fonctionnalités principales (création de transactions, alertes, etc.).



```
require("dotenv").config(); // Charger les variables d'environnement depuis .env
const hre = require("hardhat");

async function main() {
  try {
    // Vérification de la disponibilité de la clé privée dans les variables d'environnement
    if (!process.env.PRIVATE_KEY) {
      console.error("Erreur : La clé privée est manquante dans les variables d'environnement.");
      process.exit(1);
    }

    // Afficher la clé privée dans la console (pour le debug, mais à supprimer en production)
    console.log("Clé privée:", process.env.PRIVATE_KEY);

    // Récupérer le wallet à partir de la clé privée
    const wallet = new hre.ethers.Wallet(process.env.PRIVATE_KEY, hre.ethers.provider);
    console.log("Déploiement avec l'adresse :", wallet.address);

    // Déployer EntityManager
    console.log("Déploiement de EntityManager...");
    const EntityManager = await hre.ethers.getContractFactory("EntityManager", wallet);
    const entityManager = await EntityManager.deploy();
    console.log("En attente du déploiement de EntityManager...");
    await entityManager.deployed(); // S'assurer que le contrat est bien déployé
    console.log(`EntityManager déployé à l'adresse : ${entityManager.address}`);
  } catch (error) {
    console.error("Erreur lors du déploiement:", error);
  }
}
```

FIGURE 4.2 – Processus de déploiement des smart contracts.

```
// Déployer LotManager avec l'adresse de EntityManager
console.log("Déploiement de LotManager...");
const LotManager = await hre.ethers.getContractFactory("LotManager", wallet);
const lotManager = await LotManager.deploy(entityManager.address); // Passer l'adresse de EntityManager
console.log("En attente du déploiement de LotManager...");
await lotManager.deployed(); // S'assurer que le contrat est bien déployé
console.log(`LotManager déployé à l'adresse : ${lotManager.address}`);

// Déployer SupplyChainManager avec les adresses des contrats EntityManager et LotManager
console.log("Déploiement de SupplyChainManager...");
const SupplyChainManager = await hre.ethers.getContractFactory("SupplyChainManager", wallet);
const supplyChainManager = await SupplyChainManager.deploy(entityManager.address, lotManager.address);
console.log("En attente du déploiement de SupplyChainManager...");
await supplyChainManager.deployed(); // S'assurer que le contrat est bien déployé
console.log(`SupplyChainManager déployé à l'adresse : ${supplyChainManager.address}`);

console.log("Tous les contrats ont été déployés avec succès.");
} catch (error) {
  console.error("Une erreur est survenue pendant le déploiement :", error);
  process.exit(1); // Quitter avec un code d'erreur
}
}

// Exécution du script
main()
  .then(() => process.exit(0)) // Sortir avec un code de succès
  .catch((error) => {
    console.error("Erreur dans le script principal :", error);
    process.exit(1); // Sortir avec un code d'erreur
  });
```

FIGURE 4.3 – Processus de déploiement des smart contracts.

4.2 Fonctionnalités Implémentées

Les fonctionnalités suivantes ont été intégrées dans notre solution pour répondre aux besoins des utilisateurs.

Suivi des Transactions

Chaque transaction est enregistrée de manière immuable sur la blockchain Hedera Hashgraph. Les utilisateurs peuvent consulter l'historique complet des transactions via une interface web, leur permettant de :

- Vérifier l'état des transactions.
- Auditer les modifications apportées aux enregistrements.
- Recevoir des confirmations instantanées pour chaque opération effectuée.

Alertes en Cas de Transactions Non Conformes

Une fonctionnalité d'alerte a été mise en place pour détecter les transactions suspectes ou non conformes. Les critères incluent :

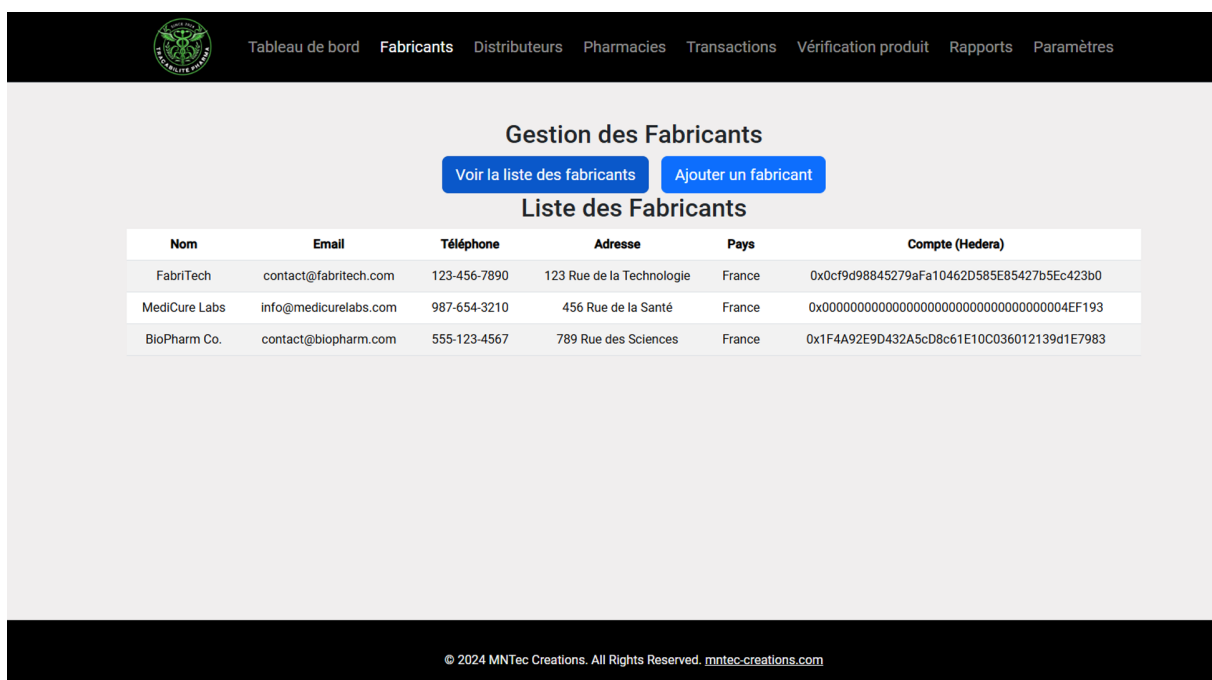
- Des montants dépassant une limite prédéfinie.
- Une tentative d'accès non autorisé.
- Une incohérence dans les données envoyées.

En cas de détection, une alerte est envoyée à l'administrateur via un email ou une notification en temps réel.

4.3 Résultats Obtenus

Les résultats obtenus démontrent l'efficacité de la solution développée. Voici quelques captures d'écran et illustrations.

Exemple de Suivi des Transactions



The screenshot displays a web application interface for 'Gestion des Fabricants'. At the top, there is a navigation bar with a logo and several menu items: 'Tableau de bord', 'Fabricants', 'Distributeurs', 'Pharmacies', 'Transactions', 'Vérification produit', 'Rapports', and 'Paramètres'. Below the navigation bar, the main heading 'Gestion des Fabricants' is centered, followed by two buttons: 'Voir la liste des fabricants' and 'Ajouter un fabricant'. Underneath these buttons is the title 'Liste des Fabricants' and a table with six columns: 'Nom', 'Email', 'Téléphone', 'Adresse', 'Pays', and 'Compte (Hedera)'. The table contains three rows of data for different manufacturers.

Nom	Email	Téléphone	Adresse	Pays	Compte (Hedera)
FabriTech	contact@fabritech.com	123-456-7890	123 Rue de la Technologie	France	0x0cf9d98845279aFa10462D585E85427b5Ec423b0
MediCure Labs	info@medicurelabs.com	987-654-3210	456 Rue de la Santé	France	0x00000000000000000000000000000000004EF193
BioPharm Co.	contact@biopharm.com	555-123-4567	789 Rue des Sciences	France	0x1F4A92E9D432A5cD8c61E10C036012139d1E7983

At the bottom of the interface, a footer indicates the copyright: '© 2024 MNTec Creations. All Rights Reserved. mntec-creations.com'.

FIGURE 4.4 – Interface utilisateur pour le suivi des transactions.

Exemple d'Alerte de Transaction Non Conforme

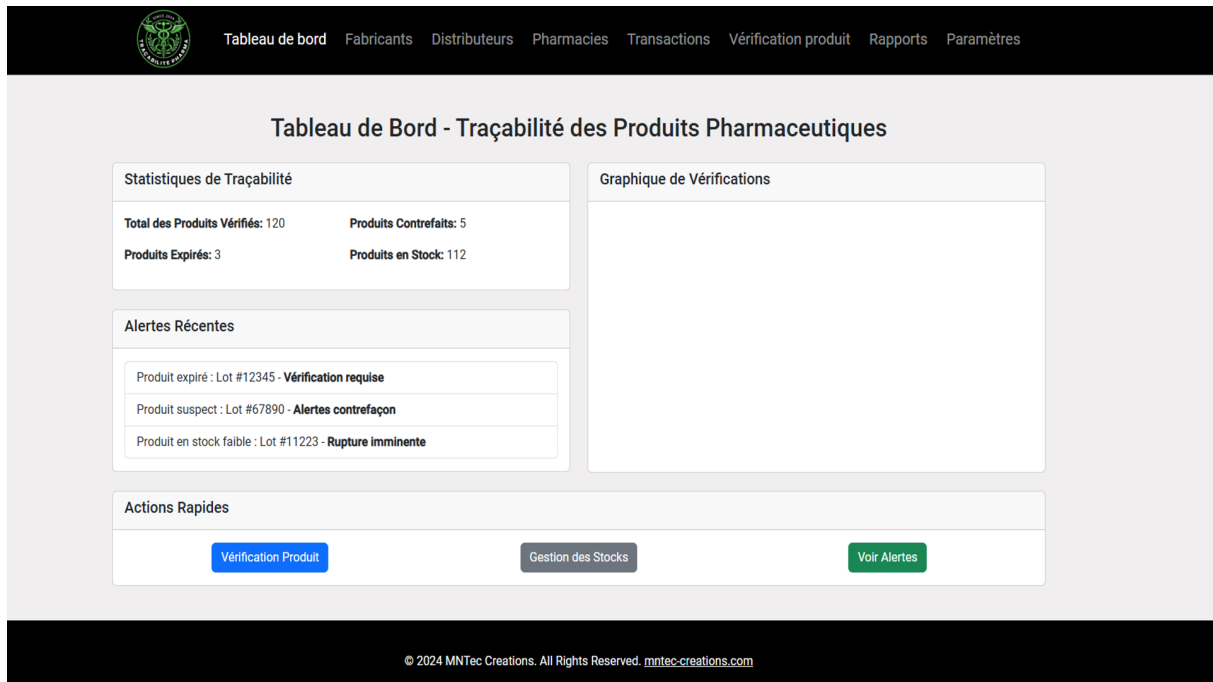


FIGURE 4.5 – Notification d'une transaction non conforme.

Ces résultats confirment que la solution répond efficacement aux besoins des utilisateurs en termes de traçabilité, de sécurité et de gestion des anomalies.

Résumé du Chapitre

Ce chapitre a présenté les étapes de développement et de déploiement des smart contracts, les principales fonctionnalités implémentées, et les résultats obtenus. Les captures d'écran fournies illustrent le fonctionnement et l'efficacité de la solution mise en œuvre.



CONCLUSION GÉNÉRALE

4.4 Conclusion et Perspectives

4.4.1 Résumé des bénéfices obtenus grâce à la solution blockchain

La solution blockchain implémentée dans le cadre de ce projet a permis de réaliser des avancées significatives en matière de sécurité, de transparence et de traçabilité des transactions. Grâce à l'utilisation de la blockchain, toutes les actions sont enregistrées de manière immuable et vérifiable, garantissant ainsi l'intégrité des données et la réduction des risques de fraude ou de falsification.

En outre, la décentralisation des transactions, sans besoin d'un intermédiaire central, a permis de réduire les coûts liés à la gestion des systèmes centralisés tout en augmentant la rapidité des échanges. L'automatisation des processus via des contrats intelligents a également optimisé les processus métier, rendant les transactions plus efficaces et réduisant les erreurs humaines.

La solution blockchain a également facilité la gestion des autorisations et de l'accès aux informations sensibles, en assurant un contrôle précis et transparent des droits d'accès.

4.4.2 Limites du projet actuel

Malgré les bénéfices notables de l'intégration de la blockchain, plusieurs limitations ont été identifiées au cours de ce projet. Tout d'abord, la scalabilité reste une problématique majeure,

notamment en raison des limitations de performance inhérentes aux réseaux blockchain publics ou privés, qui peuvent ralentir les transactions à grande échelle.

De plus, bien que les contrats intelligents aient automatisé de nombreux processus, leur développement et leur gestion restent complexes, et peuvent entraîner des vulnérabilités si mal codés ou mal vérifiés. Le manque d'une normalisation uniforme des protocoles blockchain est également une difficulté qui peut entraîner une interopérabilité limitée avec d'autres systèmes ou plateformes.

La gestion des coûts associés aux transactions sur la blockchain peut également devenir un facteur limitant. Les frais de transaction peuvent augmenter en fonction de la charge du réseau, ce qui peut rendre l'utilisation de la blockchain coûteuse pour certaines applications.

4.4.3 Suggestions pour des travaux futurs

Afin d'améliorer l'implémentation actuelle et de préparer une adoption plus large de la solution blockchain, plusieurs pistes de travail peuvent être envisagées :

- **Optimisation des coûts** : La recherche de solutions permettant de réduire les frais de transaction, telles que le passage à des réseaux blockchain plus performants ou l'utilisation de technologies de "Layer 2" (comme les solutions de scalabilité sur Ethereum) peut être explorée. Cela permettrait de rendre l'utilisation de la blockchain plus rentable, surtout pour les applications à fort volume de transactions.
- **Intégration avec plus de systèmes** : Il serait bénéfique d'étendre l'intégration de la blockchain à d'autres systèmes d'entreprise ou applications existants, notamment ceux basés sur des bases de données traditionnelles, pour améliorer l'interopérabilité. Une telle intégration favoriserait la transition des entreprises vers des solutions plus modernes, tout en préservant l'intégrité des données.
- **Amélioration de la scalabilité** : Des travaux peuvent être menés sur l'optimisation de la blockchain en utilisant des approches comme le sharding ou les solutions de consensus alternatives, telles que Proof of Stake (PoS), pour améliorer la vitesse et la capacité du réseau tout en réduisant la consommation énergétique.

- **Sécurisation des contrats intelligents** : L'amélioration des outils de développement et de vérification des contrats intelligents est essentielle pour garantir leur sécurité. Des recherches et des audits supplémentaires sur les contrats intelligents devraient être réalisés pour minimiser les risques de bugs ou de vulnérabilités exploitables.
- **Exploration des cas d'usage supplémentaires** : D'autres cas d'usage dans des secteurs tels que la gestion de la chaîne d'approvisionnement, la traçabilité des produits, ou la gestion des identités numériques pourraient être étudiés pour étendre l'application de la blockchain dans divers domaines.

En conclusion, bien que la solution blockchain déployée dans ce projet ait apporté des avantages considérables en termes de sécurité, de transparence et d'efficacité, il reste plusieurs domaines à explorer pour en maximiser le potentiel et améliorer son adoption à grande échelle.



BIBLIOGRAPHIE

- [1] 1. **Swan, M.** (2015). *Blockchain : Blueprint for a New Economy*. O'Reilly Media, Inc. Ce livre offre une introduction complète à la blockchain et explique comment elle peut transformer différents secteurs, notamment dans le domaine de la traçabilité des chaînes d'approvisionnement.
2. **Buterin, V.** (2013). *Ethereum White Paper*. Ethereum Foundation. Un document fondamental pour comprendre les bases de la blockchain Ethereum et son fonctionnement, qui est souvent utilisé dans des applications de gestion de la chaîne d'approvisionnement.
3. **Hedera Hashgraph.** (2024). *Hedera Hashgraph : A Distributed Ledger Technology*. Documentation officielle de Hedera Hashgraph, qui détaille le fonctionnement de ce système de registre distribué et ses avantages pour la transparence et la sécurité des transactions dans des systèmes complexes.
4. **MetaMask.** (2024). *MetaMask Documentation*. Documentation officielle de MetaMask, un portefeuille numérique utilisé pour gérer les transactions sur la blockchain et pour signer des contrats intelligents de manière sécurisée.
5. **MongoDB.** (2024). *MongoDB : The Definitive Guide*. O'Reilly Media, Inc. Ce livre fournit une description détaillée de MongoDB, une base de données NoSQL qui peut être utilisée pour le stockage de données non sensibles dans un système blockchain.
6. **Szabo, N.** (1997). *The Idea of Smart Contracts*. Article pionnier sur les contrats intelligents, essentiel pour comprendre comment ils peuvent être utilisés pour automatiser des processus dans des systèmes blockchain, comme celui de la traçabilité des transactions.
7. **Wood, G.** (2014). *Ethereum : A Secure Decentralized Generalized Transaction Ledger*. Une exploration approfondie d'Ethereum et de son rôle dans la facilitation des applications blockchain, notamment dans les systèmes de traçabilité.

8. **Benassi, M.** (2021). *Blockchain for Supply Chain : Concepts, Applications, and Research Directions*. Springer. Un ouvrage qui explore l'application de la blockchain dans les chaînes d'approvisionnement, offrant une perspective sur les défis, les opportunités et les solutions.
9. **O'Leary, D. E.** (2020). *Blockchain, Cryptocurrency, and the Supply Chain*. Ce livre examine les implications de la blockchain et des cryptomonnaies pour les chaînes d'approvisionnement, en mettant l'accent sur la traçabilité, la transparence et la sécurité des transactions.
10. **Swan, M.** (2018). *Blockchain for Business : The New Industrial Revolution*. Ce livre se concentre sur les applications de la blockchain dans le monde des affaires, avec un accent particulier sur l'amélioration de la traçabilité et de la sécurité des processus d'approvisionnement.
11. **Wang, Z., Chen, X.** (2020). *Blockchain Technology and its Application in Supply Chain Management*. Springer. Cette ressource explore comment la technologie blockchain peut être utilisée pour optimiser la gestion des chaînes d'approvisionnement et améliorer la sécurité et la transparence des transactions.
12. **Bitcoin.org.** (2024). *Bitcoin White Paper*. Le document fondateur de Bitcoin, qui introduit les principes de la blockchain et du consensus décentralisé, concepts clés pour comprendre le fonctionnement des systèmes de traçabilité basés sur la blockchain.
13. **R3 Corda.** (2024). *Corda Blockchain for Supply Chain*. Documentation officielle de Corda, un registre distribué qui peut être utilisé dans les applications blockchain pour améliorer la transparence et la traçabilité dans les chaînes d'approvisionnement.