

---

# RAPPORT

## DE STAGE DE FIN D'ETUDE

---

Développement d'une plateforme décentralisée  
de paiement en ligne via la monnaie digitale  
pour les sites E-Commerce

Présenté et soutenu par

**Maarouf Hamza**

Le : ../../2022

Encadré par

**Mr. Zohir CHIBA**

**Mr. Brahim RAOUYANE**

**Mr. Youssef Boussofa**

Composition du jury

**Mr. Zohir CHIBA**

Professeur à la Faculté des Sciences Ain-chock

**Mr. Brahim RAOUYANE**

Professeur à la Faculté des Sciences Ain-chock

**Mr.**

Professeur à la Faculté des Sciences Ain-chock

**Mr.**

Professeur à la Faculté des Sciences Ain-chock



# Remerciements

Au terme de ce travail, Je tiens à exprimer mes sincères remerciements à ceux qui m'ont beaucoup appris au cours de ce stage, et même à ceux qui ont eu la gentillesse de faire de ce stage un moment très profitable.

Aussi, je remercie **Pr. Zouhair CHIBA** et **Pr. Brahim RAOUYANE**, mes maîtres de stage qui m'ont formé et accompagné tout au long de cette expérience professionnelle avec beaucoup de patience et de pédagogie. Enfin, je tiens à remercier l'ensemble des collaborateurs de **MATIOUS Digital** et en particulier **M. Youssef Boussafa** pour les conseils qu'ils m'ont prodigués durant ces six mois.

# Résumé

Le mécanisme des transactions blockchain est intrinsèquement peer-to-peer et décentralisé. Néanmoins, la plupart des solutions de paiement cryptographiques établies aujourd'hui gèrent les portefeuilles de leurs clients, en intégrant une technologie décentralisée dans des structures centralisées. En conséquence, les acheteurs paient des intermédiaires qui créditent le compte du commerçant.

Donc le but de résoudre ce problème, la société **Matious Digital** a proposé la réalisation d'une solution de paiement de portefeuille à portefeuille (P2P), qui donnera la possibilité pour les commerçants de recevoir tous les jetons préférés avec la conversion d'actifs basée sur la blockchain des paiements entrants en temps réel.

**Mots clés : Blockchain, Web3, décentralisé, P2P, Paiements, Jetons, Transactions, cryptomonnaie**

# Abstract

The mechanism of blockchain transactions is inherently peer-to-peer and decentralized. Nevertheless, most of today's established Crypto Payment solutions manage wallets for their customers, embedding a decentralized technology into centralized structures. As a result, the buyers pay middlemen who credit the merchant's account.

In order to solve this problem, the company **Matiours Digital** has proposed the realization of a wallet-to-wallet (P2P) payment solution, which will give merchants the ability to receive all preferred tokens with asset conversion based on the blockchain of incoming payments in real time.

**Keywords : Blockchain, Web3, decentralized, P2P, Payments, Tokens, Transactions, cryptocurrency**

# Table des matières

|   |    |
|---|----|
| Liste des abréviations.....                               | 10 |
| Introduction générale.....                                | 11 |
| Chapitre 1 :Cadre général du projet.....                  | 12 |
| 1. Présentation de l'organisme d'accueil.....             | 13 |
| 1.1 Introduction.....                                     | 13 |
| 1.2 Fiche signalétique de l'entreprise.....               | 13 |
| 2. Contexte du projet.....                                | 14 |
| 2.1. Problématique.....                                   | 14 |
| 2.2. Solution proposée.....                               | 14 |
| 3 Planification du projet.....                            | 15 |
| 3.1 Planning du projet.....                               | 15 |
| 3.2 Diagramme de GANTT prévisionnel.....                  | 15 |
| 3.3 Diagramme de GANTT Réel.....                          | 16 |
| Conclusion.....   | 17 |
| Chapitre 2 : La Technologie Blockchain.....               | 18 |
| 2.1 Introduction.....                                     | 19 |
| 2.2 Définition.....                                       | 19 |
| 2.3 Concept.....  | 19 |
| 2.4 Mécanismes de validation de blocs.....                | 23 |
| 2.4.1 Proof of work.....                                  | 23 |
| 2.4.2 Proof of Stake.....                                 | 24 |
| 2.4.3 Delegated Proof of Stake (DPoS).....                | 24 |
| 2.4.4 Practical Byzantine Fault Tolerance (PBFT).....     | 24 |
| 2.5 Architecture.....                                     | 25 |
| 2.6 Fonctionnement.....                                   | 26 |
| 2.7 Définitions autour de la technologie Blockchain.....  | 27 |
| 2.8 Types de la Blockchain.....                           | 40 |
| 2.9 Avantages de la technologie Blockchain.....           | 41 |
| 2.10 Défis de la Blockchain.....                          | 42 |
| Conclusion.....   | 45 |
| Chapitre 3 :Analyse et Conception.....                    | 46 |
| 1. Uml.....   | 47 |
| 2. Analyse des besoins.....                               | 47 |
| 2.1. Acteurs principaux du système.....                   | 47 |
| 2.2. Identification des Rôle.....                         | 47 |
| .....   | 50 |
| 2.3. Diagramme de cas d'utilisation.....                  | 50 |
| 2.3.1. Définition.....                                    | 50 |
| 2.3.2 Diagramme des cas d'utilisations globale.....       | 51 |
| 2.3.3 Cas d'utilisation Authentification.....             | 52 |
| 2.3.4 Cas d'utilisation Authentification Web3.....        | 53 |
| 2.3.5 Cas d'utilisation Paiement.....                     | 53 |
| 2.3.6 Cas d'utilisation Gestion des intégrations.....     | 54 |
| 2.3.7 Cas d'utilisation Réalisation des transactions..... | 55 |
| 2.3.8 Cas d'utilisation Inscription.....                  | 56 |
| 3 Diagramme de séquence.....                              | 57 |
| 3.1 Diagramme de séquence d'authentification :.....       | 57 |
| 3.2 Diagramme de séquence d'authentification Web3:.....   | 58 |
| 3.3 Diagramme de séquence Paiement :.....                 | 59 |

|   |    |
|---|----|
| 3.4 Diagramme de séquence Réalisation des transactions..... | 60 |
| 3.4 Diagramme de séquence d'ajout une Intégration.....      | 62 |
| 3 Diagramme de classes.....                                 | 62 |
| Conclusion.....   | 63 |
| Chapitre 4 :Réalisation.....                                | 64 |
| 1. développement de la bibliothèque JavaScript.....         | 65 |
| 1.1 Choix des outils.....                                   | 65 |
| 3. développement de l'application Web.....                  | 69 |
| 2.1 Backend.....  | 69 |
| 2.2 Frontend.....   | 70 |
| 3. Les interfaces.....                                      | 71 |
| 3.1 Bibliothèque JavaScript.....                            | 71 |
| 3.1.1 Exemple de l'intégration.....                         | 71 |
| 3.1.2 Authentification Web3.....                            | 72 |
| 3.1.2 Paiement.....   | 74 |
| 3.2 Application Web.....                                    | 77 |
| 3.2.1 Login.....  | 77 |
| 3.2.2 Overview.....   | 77 |
| 3.2.3 Paiements.....  | 78 |
| 3.2.4 Intégration.....                                      | 78 |
| 3.2.5 Ajouter une Intégration.....                          | 79 |
| 3.2.6 Modifier une Intégration.....                         | 79 |
| 3.2.8 LogOut.....   | 80 |
| Conclusion.....   | 80 |
| <i>Conclusion Générale et Perspectives.....</i>             | 81 |
| <i>Bibliographie.....</i>                                   | 82 |
| <i>Webographie.....</i>                                     | 83 |

## Liste des tableaux

|  |    |
|--|----|
| 1. Table: Fiche signalétique de l'entreprise.....              | 14 |
| 2. Table: Planifications du projet.....                        | 15 |
| 3. Table: Comparaison entre Base de données et Blockchain..... | 19 |
| 4. Table: Champs dans une transaction.....                     | 28 |
| 5. Table: Structure de l'entete d'un block.....                | 30 |
| 6. Table: Structure d'un bloc.....                             | 31 |
| 7. Table: difference entre ethereum et bitcoin.....            | 38 |
| 8. Table: Identification des Rôle.....                         | 49 |



## Liste des figures

|   |    |
|---|----|
| 1. Figure: Diagramme de Gantt prévisionnel.....   | 18 |
| 2. Figure: Diagramme de Gantt réel.....   | 18 |
| 3. Figure: Exemple simplifié d'une Blockchain.....  | 22 |
| 4. Figure: De gauche à droite, réseaux centralisés, décentralisés et distribués (Blockchain)..... | 23 |
| 5. Figure: Un flux de travail généralisé du processus de blockchain.....                          | 24 |
| 6. Figure: Réseau basé sur les Serveurs vs Réseau P2P.....  | 27 |
| 7. Figure: Blockchain.....  | 27 |
| 8. Figure: Les étapes sur un réseau Blockchain.....   | 28 |
| 9. Figure: Chaîne de propriété de transaction.....  | 31 |
| 10. Figure: Blocs dans la Blockchain.....   | 34 |
| 11. Figure: Contenus d'un hachage de Bloc.....  | 36 |
| 12. Figure: Preuve de Travail.....  | 38 |
| 13. Figure: Arbre de Merkel.....  | 40 |
| 14. Figure: Chiffrement symétrique.....   | 42 |
| 15. Figure: Chiffrement symétrique.....   | 43 |
| 16. Figure: Signature numérique.....  | 44 |
| 17. Figure: Ethereum Virtual machine (EVM).....   | 46 |
| 18. Figure: Opportunités et défis des blockchains.....  | 50 |
| 19. Figure: Diagramme des cas d'utilisation globale.....  | 58 |
| 20. Figure: Diagramme de cas d'utilisation s'authentifier.....                                    | 59 |
| 21. Figure: Diagramme de cas d'utilisation authentification Web3.....                             | 62 |
| 22. Figure: Diagramme de cas d'utilisation Paiement.....  | 62 |
| 23. Figure: Diagramme de cas d'utilisation Gestion des intégrations.....                          | 63 |
| 24. Figure: Diagramme de cas d'utilisation Réalisation des transactions.....                      | 64 |
| 25. Figure: Diagramme de cas d'utilisation Inscription.....                                       | 65 |
| 26. Figure: Diagramme séquence système lié au cas d'utilisation Authentification.....             | 68 |
| 27. Figure: Diagramme séquence système lié au cas d'utilisation Authentification Web3.....        | 69 |
| 28. Figure: Diagramme séquence système lié au cas d'utilisation Paiement.....                     | 70 |
| 29. Figure: Diagramme séquence système lié au cas d'utilisation Transaction.....                  | 71 |
| 30. Figure: Diagramme de séquence d'ajout une Intégration.....                                    | 72 |
| 31. Figure: Diagramme des Classes.....  | 75 |
| 32. Figure: Checkout Page.....  | 83 |
| 33. Figure: d'authentification web3.....  | 84 |
| 34. Figure: Metamask.....   | 84 |
| 35. Figure: WalletConnect.....  | 85 |
| 36. Figure: Paiement.....   | 86 |
| 37. Figure: Paiement avec ETH.....  | 86 |
| 38. Figure: Les détails de la transaction dans Etherscan ETH.....                                 | 87 |
| 39. Figure: Paiement avec UNI.....  | 87 |
| 40. Figure: Les détails de la transaction dans Etherscan UNI.....                                 | 88 |
| 41. Figure: Login.....  | 89 |
| 42. Figure: OverView.....   | 89 |
| 43. Figure: Paiement.....   | 90 |
| 44. Figure: Integration.....  | 90 |
| 45. Figure: Add Integration.....  | 91 |
| 46. Figure: Update Integration.....   | 91 |
| 47. Figure: LogOut.....   | 92 |

## Liste des abréviations

| <b>Abréviation</b> | <b>Signification</b>                              |
|--------------------|---|
| <b>P2P</b>         | <b>Peer-To-Peer “Portefeuille à Portefeuille”</b> |
| <b>UML</b>         | <b>Unified Modeling Language</b>                  |
| <b>PoW</b>         | <b>Proof Of Work</b>                              |
| <b>PoS</b>         | <b>Proof Of Stake</b>                             |
| <b>DPoS</b>        | <b>Delegated Proof Of Stake</b>                   |
| <b>PBFT</b>        | <b>Practical Byzantine Fault Tolerance</b>        |
| <b>DLT</b>         | <b>Distributed Ledger Technologies</b>            |
| <b>Dapp</b>        | <b>Application Décentralisée</b>                  |
| <b>SPV</b>         | <b>Simple Paiement Verification</b>               |
| <b>Tx</b>          | <b>Transaction</b>                                |
| <b>DNS</b>         | <b>Domain Name System</b>                         |
| <b>IoT</b>         | <b>Internet Of Things</b>                         |
| <b>EVM</b>         | <b>Ethereum Virtual Machine</b>                   |
| <b>RPC</b>         | <b>Remote Procedure Call</b>                      |

# Introduction générale

Le nombre de boutiques physiques et de sites qui acceptent le bitcoin et les cryptos monnaies croît à une vitesse exponentielle. Cela prouve à suffisance que le marché des cryptos monnaies est un secteur porteur, qui est encore à l'étape embryonnaire. Il est donc désormais possible d'acheter des produits de grande consommation en ligne ou en magasin grâce aux cryptos monnaies. On peut également acheter des voitures, des appartements, bref les cryptos monnaies sont des moyens de paiements de plus en plus acceptés.

La société Maticus Corps a proposé la réalisation d'une passerelle de paiement P2P basée sur la technologie Blockchain, qui assure le Flux de trésorerie instantané, Conversion automatique, Acceptation de jetons inégalée.

Le présent rapport, qui décrit les phases du projet, comporte quatre chapitres :

- ➔ **Le premier chapitre** présente le contexte général du projet comportant une présentation de l'organisme d'accueil, le cadre du projet, ses objectifs.
- ➔ **Le deuxième chapitre**
- ➔ **Le troisième chapitre** décrit le contexte global du projet, ainsi que la description des besoins fonctionnels, et aussi l'analyse et la conception.
- ➔ **Le quatrième chapitre** est consacrée à la présentation des outils et les technologies utilisées dans la réalisation ainsi que la présentation des interfaces de l'application développée.

# **Chapitre 1 :Cadre général du projet**

Dans ce chapitre, je présente mon organisme d'accueil du stage, par la suite, je fais une étude de l'existant afin de relever les insuffisances et de proposer une solution efficace

# 1. Présentation de l'organisme d'accueil

## 1.1 Introduction



**Matious Digital** est une agence digitale spécialisée dans l'accompagnement technologique et marketing pour des entreprises en Europe et en Amérique du nord.

Depuis la création de **Matious Digital** l'objectif principale qui vise principalement à le réaliser est d'utiliser des technologies de pointe pour développer des solutions futuristes.

La mission de **Matious Digital** concerne plusieurs axes tels que :

- **Développement web.**
- **Développement mobile.**
- **Data Science & ML.**
- **Marketing Digital.**
- **Applications décentralisées**
- **Développement de contrat intelligent**

## 1.2 Fiche signalétique de l'entreprise

Raison sociale                      **MATIOUS CORP**

Forme Juridique                      **S.a.r.l.**

|                            |  |
|----------------------------|--|
| <b>Activité principale</b> | <b>Producteur   Distributeur   Prestataire de services</b>       |
| <b>Directeur général</b>   | <b>Mati Mouni</b>  |
| <b>Siège social</b>        | <b>Central park, Im M, E2, N13, Mohammedia, Morocco</b>          |
| <b>Année de création</b>   | <b>2016</b>  |
| <b>Tél</b>                 | <b>+212.661.790.436 +212.808.518.588</b>                         |
| <b>Site Web</b>            | <b><a href="https://matio.us.com/">https://matio.us.com/</a></b> |

*1. Table: Fiche signalétique de l'entreprise*

## 2. Contexte du projet

### 2.1. Problématique

Actuellement, les paiements Web3 sont bruts et inégaux. Les utilisateurs sont souvent obligés d'échanger leurs cryptomonnaies et jetons sur des échanges tiers avant de pouvoir payer. Au cours de ces échanges, les utilisateurs sont confrontés à des procédures d'inscription compliquées ou doivent avoir un minimum de connaissances techniques.

Dans ce contexte, plusieurs solutions ont été développées mais La plupart des solutions de paiement web3 établies aujourd'hui gèrent les portefeuilles de leurs clients, en intégrant une technologie décentralisée dans des structures centralisées. En conséquence, les acheteurs paient des intermédiaires pour créditer le compte du marchand.

### 2.2. Solution proposée

Pour résoudre ces problèmes on a développé une passerelle de paiement peer-to-peer qui utilise les smart contracts pour la conversion à la volée.

Ce projet a pour objectif :

- **Flux de trésorerie instantané** : les paiements reçus sont réglés et disponibles pour les commerçants en temps réel.
- **Conversion automatique** : les jetons sont automatiquement convertis (par exemple en stablecoins) dans le cadre de la transaction de paiement.
- **Acceptation de jetons inégalée**

## 3 Planification du projet

### 3.1 Planning du projet

Afin de mener le projet en toute clarté, on a recouru à quelques méthodes de gestion de projet.

La planification du projet consiste à organiser le déroulement des étapes à réaliser dans le temps afin d'avoir une vision claire du temps et avoir connaissance de l'ensemble des tâches, de leurs dépendances et de leur organisation chronologique au sein du projet.

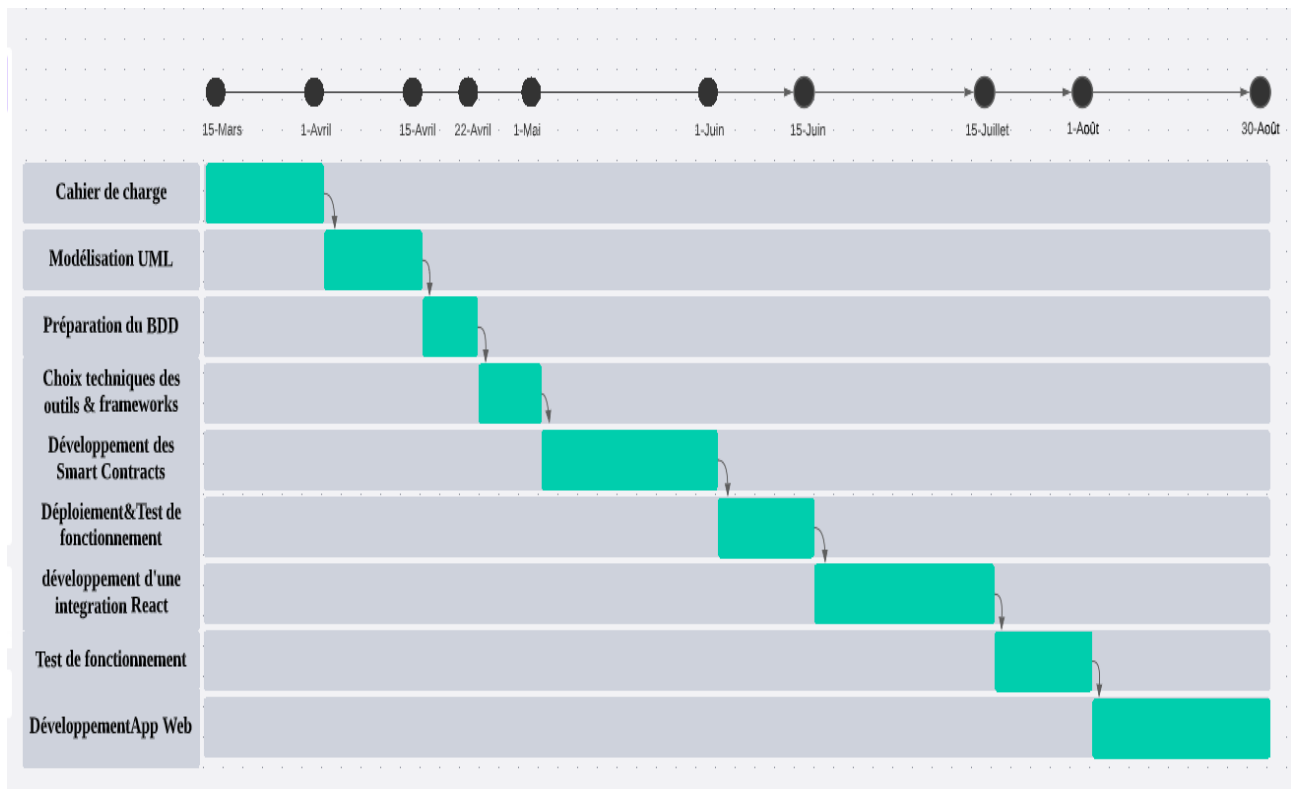
Dans cette partie de gestion de projet, on découpe le projet en actions à exécuter pour répondre aux besoins de l'entreprise et on réalise un plan des capacités temps et ressources en donnant une estimation de la durée des tâches.

| # | Nom                                      | Durée    | Début            | Fin              |
|---|--|----------|------------------|------------------|
| 1 | Cahier de charge                         | 15Jours  | 15 Mars          | 1 Avril. 2022    |
| 2 | Modélisation UML                         | 15Jours  | 1 Avril. 2022    | 15 Avril. 2022   |
| 3 | Préparation du BDD                       | 7Jours   | 15 Avril. 2022   | 22 Avril. 2022   |
| 4 | Choix techniques des outils & frameworks | 8Jours   | 22 Avril. 2022   | 30 Avril. 2022   |
| 5 | Développement des Smart Contracts        | 1mois    | 1 Mai. 2022      | 1 juin. 2022     |
| 6 | Déploiement & Test de fonctionnement     | 15Jours  | 1 juin. 2022     | 15 juin. 2022    |
| 7 | développement d'une integration React    | 1 Mois   | 15 Juin. 2022    | 15 juillet. 2022 |
| 8 | Test de fonctionnement                   | 15 Jours | 15 juillet. 2022 | 30 juillet. 2022 |
| 9 | Développement App Web                    | 1 Mois   | 1 Août. 2022     | 30 Août. 2022    |

2. Table: Planifications du projet

### 3.2 Diagramme de GANTT prévisionnel

L'objectif de ce planning est de déterminer les étapes du projet et le timing. Ce planning joue un rôle primordial pour la réalisation et le suivi du projet, il est établi dans le début de chaque projet afin de suivre le bon déroulement de chaque tâche.

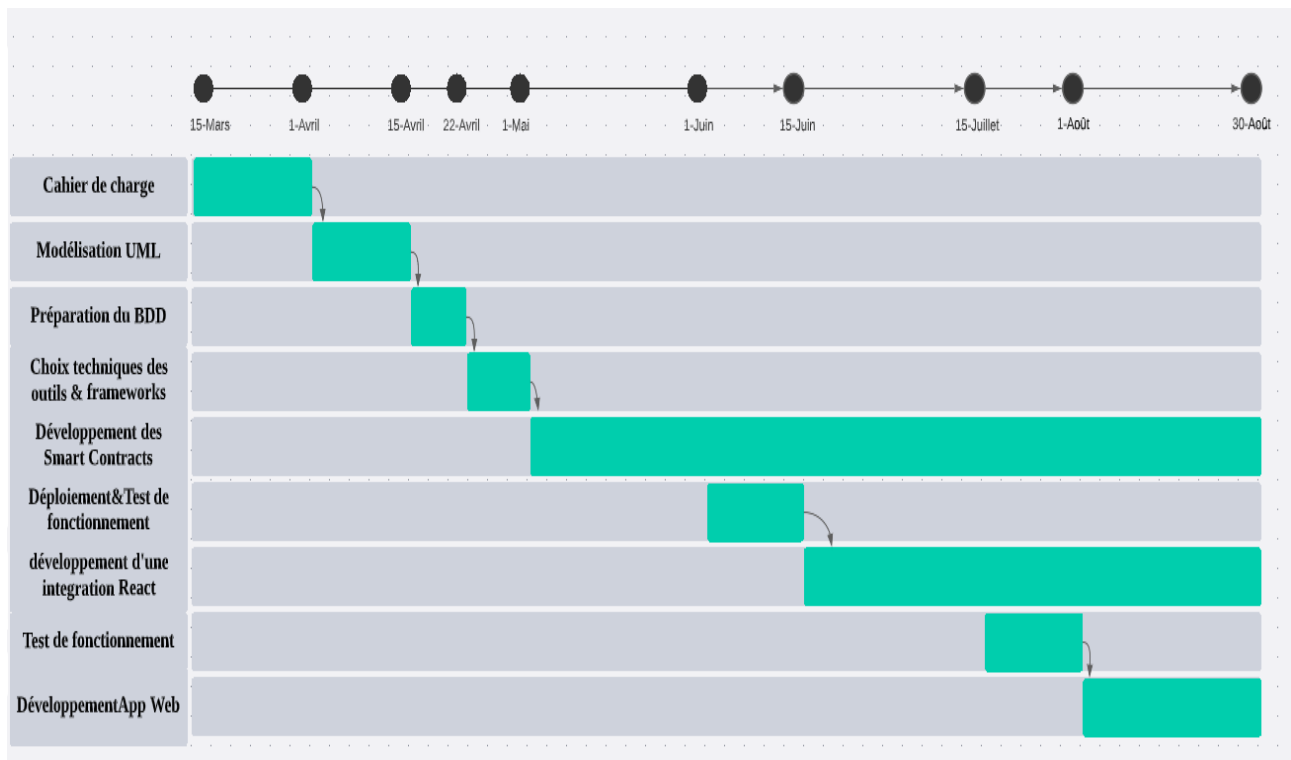


1. Figure: Diagramme de Gantt prévisionnel

### 3.3 Diagramme de GANTT Réel

Ce planning présente l'état d'avancement réel du projet, il est développé à la fin du projet





2. Figure: Diagramme de Gantt réel

## Conclusion

Ce chapitre a été consacré au début à une présentation de l'organisme d'accueil, ensuite nous avons donné la description du problème posé et nous avons ainsi défini les différents objectifs de notre application, et à la fin nous avons situé les différents outils de collaboration.

## **Chapitre 2 : La Technologie Blockchain**

Ce chapitre présente les notions générales de la technologie Blockchain, incluant l'historique, les domaines d'application, et l'architecture et le fonctionnement.

## 2.1 Introduction

Le 31 octobre 2008, un inconnu utilisant le pseudonyme « **Satoshi Nakamoto** » a écrit dans une liste de diffusion d'e-mails réservée aux cypherpunks (un mouvement de personnes utilisant la cryptographie pour protéger la vie privée). : “Je travaille sur un nouveau système de monnaie électronique entièrement de pair-à-pair, sans tiers de confiance”. Ce texte est accompagné d’un lien qui amène vers **Bitcoin.org** et sur lequel est hébergé le livre blanc du Bitcoin, rédigé dans un anglais impeccable, résumant le fonctionnement du nouveau protocole. Le premier concept de Blockchain a été appliqué le 03 Janvier 2009 dans le cadre de Bitcoin.

La technologie à la base de Bitcoin et d’autres crypto-monnaies, est une base de données de grand livre distribuée pour l’enregistrement des transactions, permettant ainsi aux utilisateurs de partager leur grand livre de transactions.

## 2.2 Définition

Une blockchain est une base de données de transactions qui est mise à jour et partagée sur de nombreux ordinateurs d'un réseau. Chaque fois qu'un nouvel ensemble de transactions est ajouté, il s'appelle "bloc" - d'où le nom de blockchain. La plupart des blockchains sont publics, et vous ne pouvez ajouter que des données et non pas les supprimer. Si quelqu'un voulait modifier l'une des informations ou flouer le système, il devrait le faire sur la majorité des ordinateurs du réseau. C'est beaucoup ! Les blockchains reconnues comme Ethereum sont de fait hautement sécurisées.

## 2.3 Concept

Une blockchain, ou chaîne de blocs, est définie comme une base de données distribué (ledger) qui conserve un enregistrement permanent et immuable (infalsifiable) des données transactionnelles liées entre elles par une chaîne (par blocs).

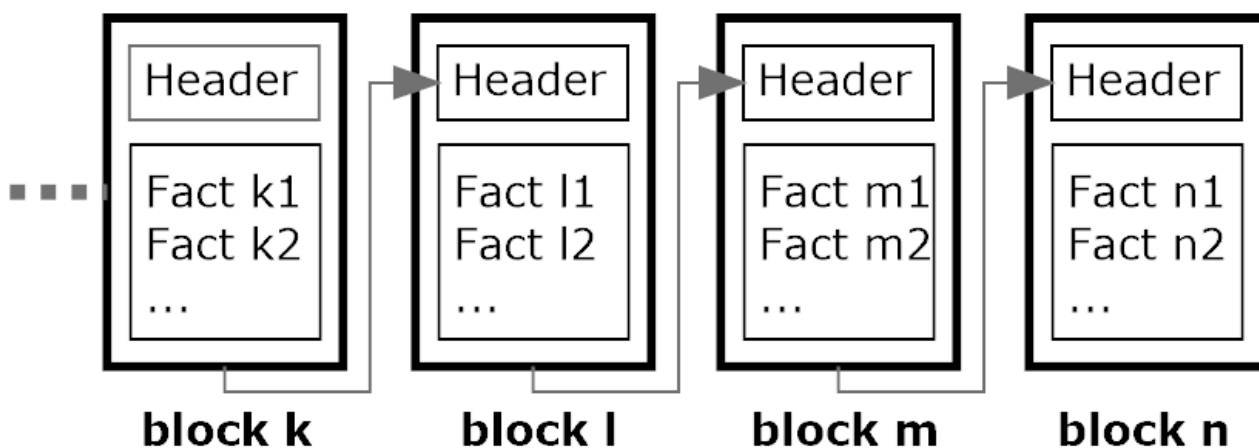
| Propriétés  | Blockchain   | Base de données traditionnelle                    |
|-------------|--|---|
| Opérations  | Seulement des opérations d'insertion                               | Peut effectuer des opérations CRUD                |
| Réplication | Réplication complète du bloc sur chaque pair                       | Maître esclave multi-maître                       |
| Consensus   | La majorité des pairs s'accordent sur le résultat des transactions | Transactions distribuées (validation en 2 phases) |
| Invariants  | Tout le monde peut valider les transactions sur le réseau          | Contraintes d'intégrité                           |

3. Table: Comparaison entre Base de données et Blockchain

Une blockchain est un système totalement décentralisé et basé sur un réseau pair à pair (peer-to-peer). Chaque objet du réseau conserve une copie du ledger afin d'éviter d'avoir un point unique de défaillance. Toutes les copies sont mises à jour et validées simultanément. Bien que l'objectif initial de la création de la blockchain fût la résolution du problème de la dépense multiple en crypto monnaie (monnaie virtuelle). Cette technologie peut être explorée dans de nombreux cas d'utilisation et utilisée comme un moyen sécurisé de gestion et protection de toute sorte de données (monétaire ou pas).

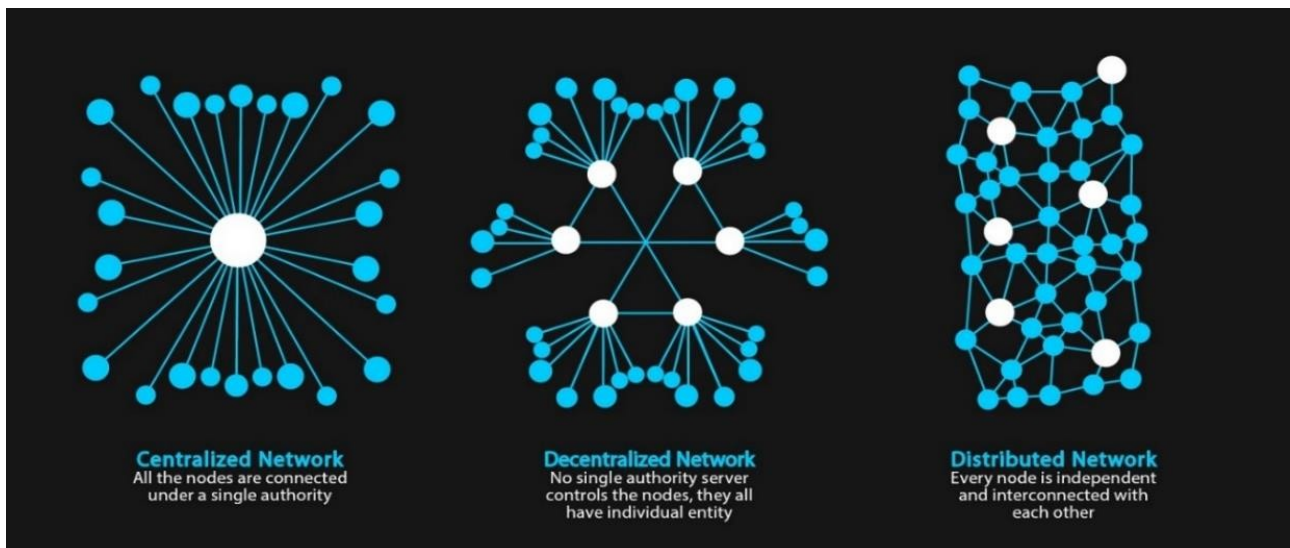
Le ledger est composé d'un ensemble de blocs. Chaque bloc contient deux parties. La première partie représente le corps du bloc. Il contient les transactions, appelées également faits (facts), que la base de données doit enregistrer. Ces faits peuvent être des transactions monétaires, des données médicales, des informations industrielles, des logs systèmes, etc. La deuxième partie est l'entête (header) du bloc. Ce dernier contient des informations concernant le bloc telque l'horodatage (timestamp), le hach des transactions, etc. Ainsi que le hachage du bloc précédent. De ce fait, l'ensemble des blocs existants forme une chaîne de blocs liés et ordonnés. Plus la chaîne est longue, plus il est difficile de la falsifier.

En effet, si un utilisateur malicieux veut modifier ou échanger une transaction sur un bloc, il doit modifier tous les blocs suivants, puisqu'ils sont liés par leurs hachs. Ensuite, il doit changer la version de la chaîne de blocs que chaque objet participant stocke.



3. Figure: Exemple simplifié d'une Blockchain

Une blockchain suit un réseau P2P. il s'agit essentiellement d'un cadre de réseau multi-réseaux intégré entre pairs, composé de cryptographie, d'algorithmes et d'expressions mathématiques visant à résoudre les limitations classiques de la synchronisation de bases de données distribuées à l'aide d'algorithmes de consensus distribués.



4. Figure: De gauche à droite, réseaux centralisés, décentralisés et distribués (Blockchain)

La technologie Blockchain se caractérise principalement de six éléments majeurs : décentralisé, transparente, sécurisé et immuable, autonome, open source et anonyme. Comme décrit ci- dessus :

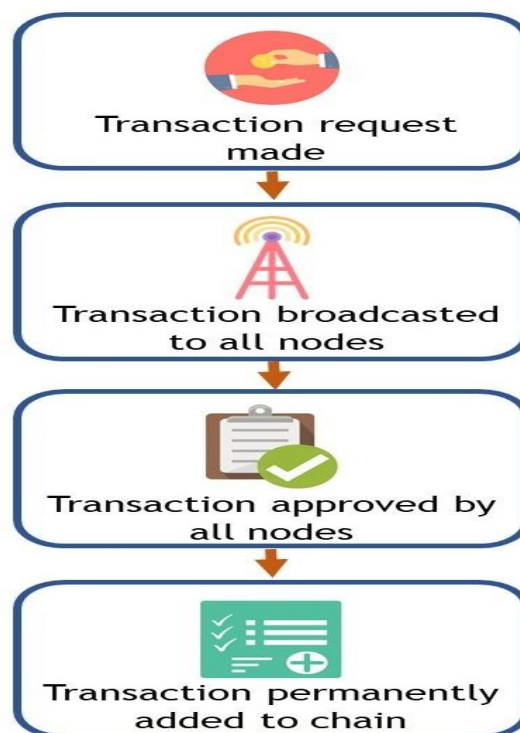
- ➔ **Elle est décentralisée** : La blockchain contient Un système de bases de données décentralisé avec un contrôle en libre accès pour tous ceux qui sont connectés au réseau. Les données peuvent être consultées, surveillées, stockées et mises à jour sur plusieurs systèmes. Ces données ne sont pas toutes regroupées dans le serveur d'un intermédiaire central, mais au contraire « distribuées », c'est-à-dire hébergées chez chaque participant ; il n'y a donc pas d'autorité unique pouvant approuver les transactions ou définir des règles spécifiques pour que les transactions soient acceptées. Cela signifie que la confiance est énorme, car tous les participants du réseau doivent parvenir à un consensus pour accepter les transactions.
- ➔ **Elle est transparente** : C'est l'avantage le plus important. Tous les participants peuvent voir les blocs et les transactions qui y sont stockés dedans. Les données enregistrées et stockées dans la blockchain sont transparentes pour les utilisateurs potentiels et peuvent être mises à jour facilement. Cela ne signifie toutefois pas que tout le monde peut voir le contenu réel des transactions, qui sont protégés par une clé privée.
- ➔ **le consensus** : la blockchain correspond à un historique de transactions sur lequel tout le monde s'accorde, ce consensus sur le séquençement des transactions permet de résoudre le problème dit de la "double dépense" : un Bitcoin dépensé dans une transaction ne peut pas être dépensé une deuxième fois dans une transaction qui serait diffusée ultérieurement sur le réseau. La deuxième transaction serait rejetée par le réseau.
- ➔ **Elle est sécurisée** : La base de données peut uniquement être étendue et les enregistrements précédents ne peuvent pas être modifiés (au moins, le coût est très élevé si quelqu'un souhaite modifier les enregistrements précédents).

Ces enregistrements sont dits Immuables, une fois stockés, deviennent réservés pour toujours et ne peuvent pas être modifiés facilement sans le contrôle simultané de plus de 51% des nœuds du réseau.

Le système cryptographique de validation garantit qu'il est quasiment impossible de réécrire une transaction une fois son bloc validé (personne n'a réussi à le faire depuis la création du Bitcoin).

- ➔ **Autonome** : Le système blockchain est indépendant et autonome, ce qui signifie que chaque nœud du système blockchain peut accéder aux données, les transférer, les stocker et les mettre à jour en toute sécurité, ce qui les rend fiables et exemptes de toute intervention externe.
- ➔ **Open source** : La technologie de la blockchain est formulée de manière à fournir un accès open source à toutes les personnes connectées au réseau. Cette polyvalence inimitable permet à quiconque non seulement de vérifier publiquement les enregistrements, mais également de développer diverses applications imminentes.
- ➔ **Anonyme** : Lorsque le transfert de données a lieu entre nœuds, l'identité de l'individu reste anonyme, ce qui en fait un système plus sécurisé et fiable.

Une personne faisant partie de ce réseau doit vérifier chaque nouvelle transaction effectuée. Une transaction de recherche dans un bloc d'une blockchain est vérifiée par tous les nœuds du réseau, elle devient de plus en plus immuable. La figure 28 ci-dessous illustre le flux de travail du processus de la chaîne de blocs.



5. Figure: Un flux de travail généralisé du processus de blockchain

Il existe deux types d'objets participants dans la blockchain : des objets qui peuvent uniquement lire les facts (mode passif), et des objets qui peuvent lire et écrire des facts (mode actif) appelés mineurs. Afin de rajouter un nouveau bloc à la blockchain, il faut suivre les étapes suivantes :

- Une transaction est regroupée avec d'autres transactions dans un bloc;
- Les mineurs vérifient que les transactions du bloc respectent les règles définies.
- Les mineurs exécutent un mécanisme de consensus pour valider le bloc ajouté.
- Une récompense est donnée aux mineur/mineurs qui valident le bloc.
- Les transactions vérifiées sont stockées dans la blockchain.

Afin de prouver la validation honnête d'un bloc, il existe de nombreux mécanismes de validation. Les plus utilisés sont le mécanisme de Proof of Work (PoW) et le mécanisme de Proof of Stake (PoS).

## 2.4 Mécanismes de validation de blocs

### 2.4.1 Proof of work

Dans ce mécanisme, un mineur doit effectuer une quantité de travail, qui est souvent un puzzle ou un défi mathématique, difficile à calculer mais facile à vérifier. La difficulté du défi est adaptée, par la blockchain, en fonction du temps nécessaire à la validation d'un bloc.

Une PoW est exigée pour la validation de chaque bloc. D'une part, elle a l'avantage de protéger l'intégrité des transactions et des blocs, car afin qu'un attaquant puisse modifier un bloc, il doit modifier tous les blocs qui le succèdent et fournir une nouvelle PoW pour chacun de ces blocs, ainsi que la mise à jour de tous les objets par la nouvelle version de la chaîne (falsifiée). Ce qui nécessite une énorme puissance de calcul et d'énergie. D'autre part, la PoW souffre de certaines lacunes qui peuvent avoir des mauvaises conséquences. Sans parler du fait que la PoW consomme une grande quantité d'énergie lors de la résolution du défi mathématique, ce mécanisme peut mener à une potentielle tragédie des ressources d'usage commun (tragedy of commons) [122]. En effet, au fil du temps, les récompenses diminueront, ce qui entraînera une diminution du nombre de mineurs, car les seuls frais qui seront gagnés viendront des transactions. Ces frais de transactions vont également diminuer due à la concurrence d'autres systèmes similaires. La diminution du nombre de mineurs rend l'écosystème blockchain vulnérable à l'attaque du 51% . Cette dernière se produit lorsqu'un mineur malicieux (ou un pool de mineurs malicieux) contrôlent 51%, ou plus, de la puissance de calcul du réseau. Ainsi, il peut créer des blocs de transactions frauduleux pour lui-même, ou pour une autre entité, tout en invalidant les transactions des autres utilisateurs dans le réseau. Enfin, dans certains mécanismes de consensus, tel que celui de la chaîne la plus longue (longest chain) appliqué dans Bitcoin (voir section 4.2 page ci-contre), de nombreux mineurs qui valident des blocs et réalisent la PoW ne sont pas récompensés, car ils n'ont pas assez de puissance pour construire la chaîne la

plus longue, ce qui leur cause beaucoup de pertes. La PoW représente la méthode de validation de bloc la plus adoptée par les systèmes blockchain.

Le concept du l'algorithme le plus utilisé dans la Blockchain existait bien avant sa naissance. Il a été publié à l'origine par Cynthia Dwork et Moni Naor en 1993, mais le terme «preuve de travail» a été inventé par Markus Jakobsson et Ari Juels dans un document publié en 1999 (Blockgeeks, 2017). Dans le cas de Bitcoin, la preuve de travail suppose que tous les membres du réseau votent en utilisant leur puissance de calcul en résolvant le PoW et la construction et la validation du bloc. La preuve de travail peut être considérée comme le principal composant afin de définir un calcul informatique coûteux, également appelé extraction qui doit être effectuée afin de générer un nouveau bloc.

Les mineurs servent à deux fins : vérifier la légitimité d'une transaction et éviter les doubles dépenses.

## **2.4.2 Proof of Stake**

Afin de résoudre les lacunes de la PoW (Preuve de Travail), la PoS (Preuve d'Enjeu) a été proposée. Dans ce mécanisme il n'y a pas de minage où on consomme beaucoup de ressources. Les mineurs sont appelés forgers. Un forger peut valider des blocs en fonction de la quantité d'argent qu'il possède. Ce qui signifie que plus il possède de monnaies, plus il augmente sa chance de validation. Si on compare la PoS à un jeu de pari, où chaque forger parie sur un bloc. On peut dire qu'une fois que les blocs honnêtes (ne contiennent aucune transaction frauduleuse) sont ajoutés à la chaîne, chaque forger touche une récompense relative à son pari. Et contrairement à la PoW où les mineurs malicieux sont pardonnés, dans la PoS un forger dont le bloc s'avère malhonnête est pénalisé et le montant du pari qu'il a mis est débité de son solde. Le point faible de la PoS est que les forgers qui possèdent beaucoup de monnaies sont ceux qui bénéficient le plus. Il existe plusieurs systèmes blockchain qui utilisent la PoS, et d'autres qui remplacent la PoW par la PoS.

Il existe d'autres mécanismes de validation de blocs tel que la Delegated Proof-of-Stake (DPoS), la Proof of Stake/Time (PoST), la Proof of Existence (PoE), etc.

## **2.4.3 Delegated Proof of Stake (DPoS)**

La principale différence entre les PoS et les DPoS réside dans le fait que les PoS sont un processus démocratique direct, tandis que le DPoS est démocratiquement représentatif – les parties prenantes élisent des délégués pour générer et valider un bloc. Avec beaucoup moins de nœuds pour valider le bloc, le bloc peut être confirmé rapidement, ce qui signifie que la transaction peut être confirmée rapidement (Zheng, 2016) et (Kikitamara, 2017)

## **2.4.4 Practical Byzantine Fault Tolerance (PBFT)**

Cet algorithme de consensus a été développé pour tolérer les fautes byzantines, par exemple le comportement arbitraire du nœud, qui rejoint et quitte le réseau à tout moment qui se produit généralement dans un système distribué. Cet algorithme présente une technique de réplication de machine à états permettant de gérer les erreurs byzantines. Théoriquement, il utilise un algorithme



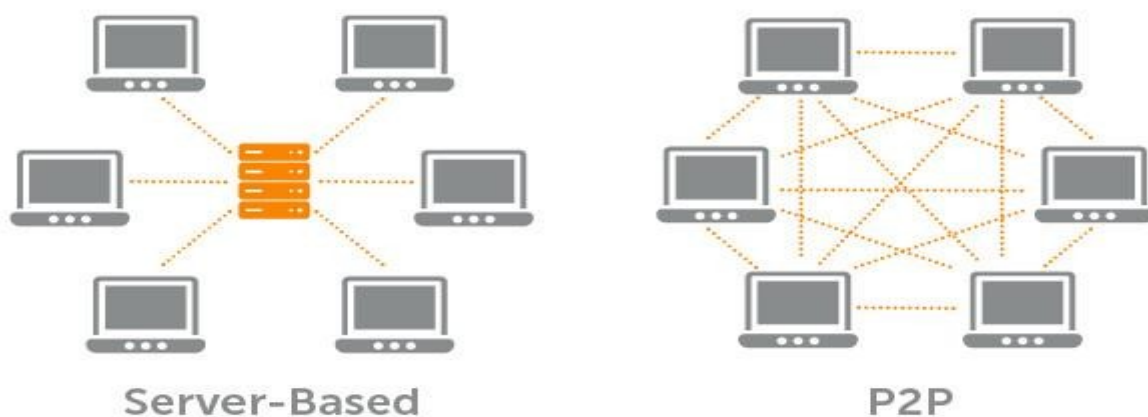
de réplication de la machine d'état avec un seul aller-retour de message pour exécuter des opérations en lecture seule et deux pour exécuter des opérations de lecture-écriture. En outre, il utilise un schéma d'authentification efficace basé sur les codes d'authentification du message en cours de fonctionnement normal ; la cryptographie à clé publique est utilisée uniquement lorsqu'il y a des erreurs (Castro et Liskov, 1999).

## 2.5 Architecture

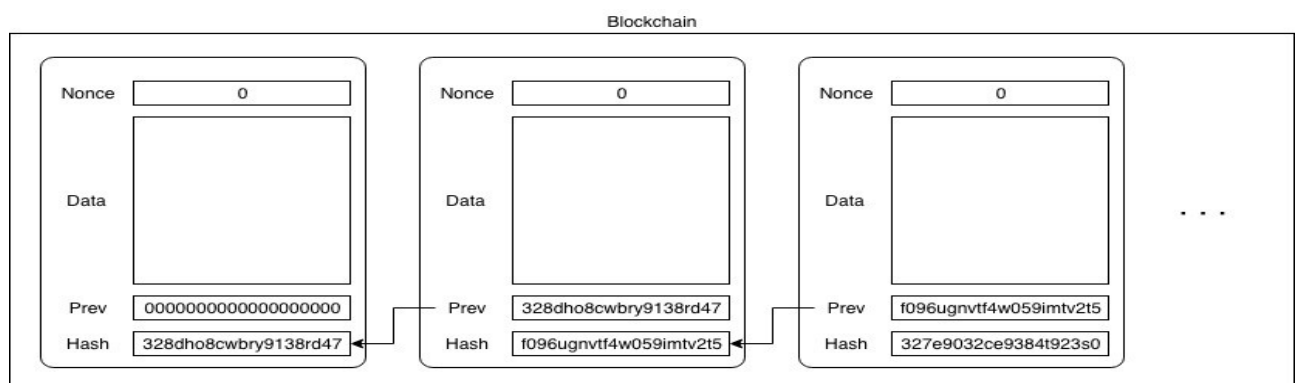
L'architecture réseau d'un réseau distribué Blockchain est Peer to Peer, Le réseau d'égal à égal, également appelé P2P, fait référence à un groupe d'ordinateurs agissant en tant que nœuds pour partager des fichiers entre eux-mêmes.

La Blockchain fonctionne donc sur un réseau distribué de serveurs, également appelé nœuds. Ces nœuds du réseau ont pour objectif de fournir un consensus sur l'état de la blockchain à tout moment, et contiennent une copie de la blockchain.

L'application fondamentale de la Blockchain est un grand livre de transactions, un peu comme un grand livre public sécurisé, qui stocke toutes les transactions qui ont lieu dans le réseau. Cela en fait un système décentralisé très sécurisé et transparent.



6. Figure: Réseau basé sur les Serveurs vs Réseau P2P

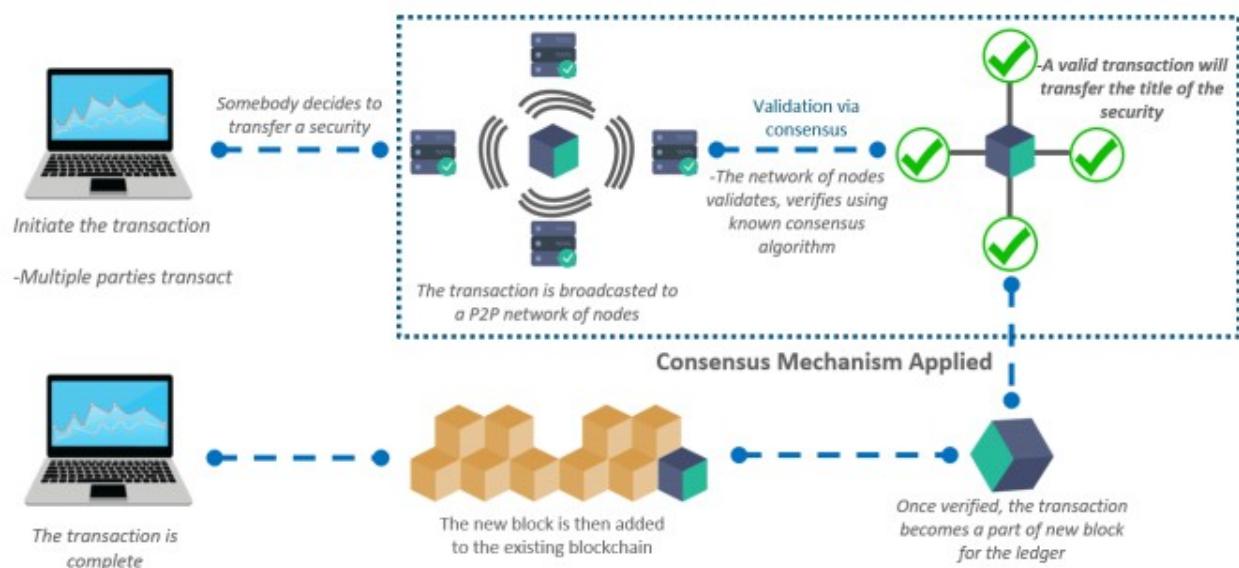


7. Figure: Blockchain

## 2.6 Fonctionnement

Les interactions entre les comptes d'un réseau blockchain sont appelées "transactions". Il peut s'agir de transactions monétaires, comme l'envoi d'éther, la crypto-monnaie utilisée dans Ethereum. Elles peuvent également être des transmissions de données, comme un commentaire ou un nom d'utilisateur. Un ensemble de transactions est appelé un "bloc".

Chaque compte sur la blockchain possède une signature unique, qui permet à chacun de savoir quel compte a initié la transaction. Sur une blockchain publique, tout le monde peut lire ou écrire des données. La lecture des données est gratuite, mais l'écriture sur la blockchain publique est payante. Ce coût, appelé "gaz" et fixé en éther, contribue à décourager le spam et à sécuriser le réseau.



8. Figure: Les étapes sur un réseau Blockchain

La figure 8 illustre le mécanisme de fonctionnement des transactions dans le réseau Blockchain.

Les étapes de ce mécanisme sont les suivantes :

1. Quelqu'un demande une transaction.
2. La transaction est diffusée sur un réseau P2P public (réseau Blockchain) composé de plusieurs nœuds.
3. Le réseau de nœuds valide la transaction en utilisant les algorithmes de hachage.
4. Une fois vérifiée, la transaction est combinée avec d'autres transactions pour créer un nouveau bloc de données pour le grand livre.
5. Le nouveau bloc est ajouté à la chaîne de blocs existante, sous une forme qui est permanente et inaltérable.

6. Enfin la transaction sera effectuée avec succès.

## 2.7 Définitions autour de la technologie Blockchain

Dans cette section, On explique les principaux concepts liés à la technologie Blockchain et son fonctionnement. Ces concepts sont : **Nœuds**, **peer to peer (P2P)**, **DLT (Distributed Ledger Technologies)**, **Transactions**, **Blocs**, **Hachage** , **Merkle tree**, **Minage**, **Consensus**, **Attaque 51%**, **Ether**, **Dapp**, **Gas**, **Ethereum**, **ERC-20**, **ERC-721**, **EVM**, **La cryptographie symétrique**, **Contrat Intelligent**, **Nonce**, **La cryptographie asymétrique**, **Signature numérique**

- **Les nœuds** ou les clients connectés au réseau dans le système Blockchain constituent une partie essentielle du système. Ils prennent en charge diverses fonctions telles que le routage, l'extraction, le stockage des données de la blockchain et leur utilisation en tant que portefeuille. Tous les nœuds participent à la vérification et à la propagation des transactions et sont activés avec des fonctionnalités telles que la découverte et le maintien de la connexion avec leurs pairs. Ils conservent également une copie du registre, ou de la blockchain, qui contient des données sur toutes les transactions qui se sont déjà produites, éliminant ainsi la nécessité de disposer d'un serveur centralisé pour le stocker. Les nœuds peuvent également agir en tant que mineurs et aider à vérifier et valider toutes les transactions effectuées par tous les utilisateurs. Tous les mineurs sont des nœuds, mais tous les nœuds ne sont pas nécessairement des mineurs. Les nœuds sont principalement de trois types, à savoir les nœuds complets, les clients SPV (Simple Paiement Verification) et les clients Web.

Les clients Web, généralement appelés portefeuilles, sont stockés sur des serveurs tiers et sont accessibles via les navigateurs Web.

Les nœuds SPV incluent généralement des clients ne disposant pas de capacités matérielles suffisantes, telles que des périphériques mobiles, ou de périphériques plus limités, tels que des systèmes intégrés. Ces nœuds SPV n'ont pas besoin de stocker une copie de toutes les transactions dans la blockchain, mais uniquement les en-têtes de bloc. Ces nœuds ont un moyen légèrement différent de vérifier les transactions des nœuds complets, car ils ne gardent pas une trace de toutes les transactions se déroulant sur la blockchain. Ils dépendent de leurs nœuds homologues pour fournir les informations de transaction dont ils ont besoin, à la demande.

Les nœuds complets, en revanche, sont les nœuds qui stockent une copie à jour de la chaîne de blocs dans son intégralité.

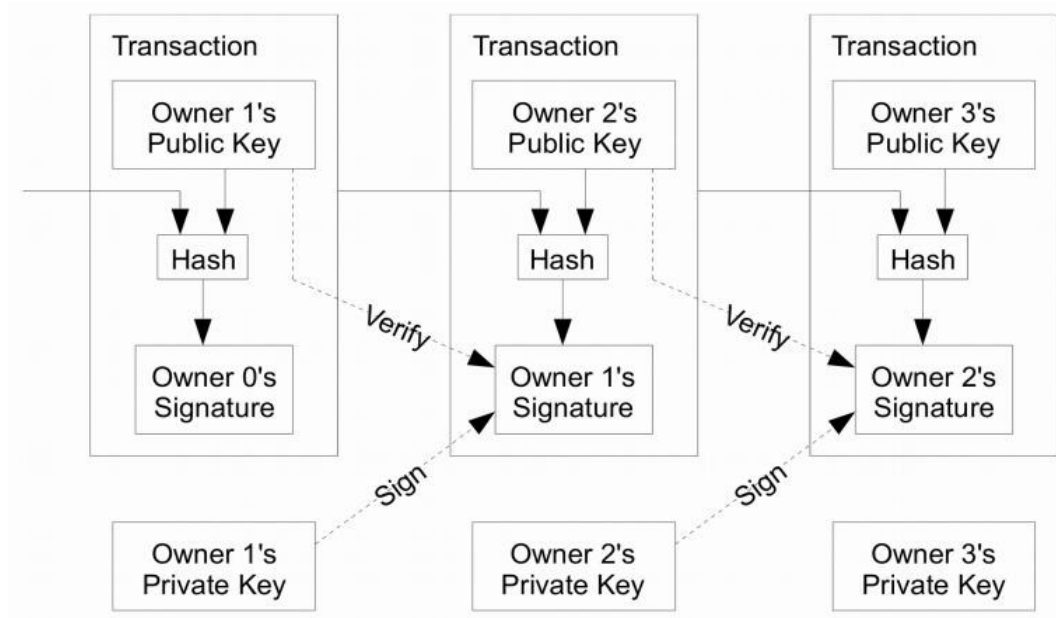
Une fois que l'un des nœuds est connecté au réseau local sans fil, le système recherche d'autres pairs auxquels se connecter, sur un port particulier via TCP. Ce processus de découverte de nœud est également appelé protocole de découverte.

- **P2P (Peer to Peer)** est un réseau dans lequel les ordinateurs servent de nœuds pour le partage de fichiers au sein du groupe. Les périphériques ou ordinateurs participant à ce réseau sont appelés Pairs. Chaque pair est égal à une autre. Il n'existe donc aucun périphérique administrateur central au centre du réseau ni une partie privilégiée.
  
- **DLT (Distributed Ledger Technology)** est un type de base de données consensuel partagé, répliqué et synchronisé sur les membres d'un réseau. La principale caractéristique de cette base de données est que les transactions et leurs détails sont enregistrés simultanément à plusieurs endroits. DLT n'a pas de magasin de données central.
  
- **Les transactions** sont stockées dans les fichiers appelés blocs. Ils sont cryptés et sont généralement liés à des transactions précédentes, formant ainsi une chaîne. Un propriétaire d'une valeur (devise numérique dans BC Bitcoin) signe numériquement la transaction précédente avec sa clé publique et crée un hachage. Le propriétaire de la transaction précédente signe alors le hachage avec sa clé privée.

La figure 7 illustre une version simplifiée de la chaîne de propriété. Dans des cas plus complexes, le nombre d'entrées et des sorties peuvent être multiples. Une transaction contient un certain nombre de champs, comme indiqué dans le tableau 4.

| Size               | Field          | Description                                    |
|--------------------|----------------|--|
| 4 bytes            | Version        | Specifies which rules this transaction follows |
| 1-9 bytes (VarInt) | Input Counter  | How many inputs are included                   |
| Variable           | Inputs         | One or more transaction inputs                 |
| 1-9 bytes (VarInt) | Output Counter | How many outputs are included                  |
| Variables          | Outputs        | One or more transaction outputs                |
| 4 bytes            | Locktime       | A Unix timestamp or block number               |

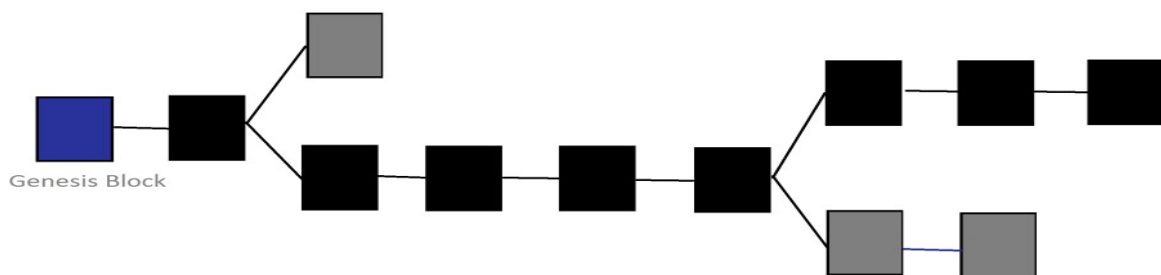
4. Table: Champs dans une transaction



9. Figure: Chaîne de propriété de transaction

Dans BC Bitcoin la transaction est un transfert de valeur Bitcoin qui diffuse sur le réseau et est collectée en blocs. Les transactions ne sont pas cryptées. Il est donc possible d'afficher chaque transaction collectée dans un bloc. Elles sont regroupées en blocs et exécutées sur tous les nœuds participants.

- **Un bloc** contient une liste de transactions, l'état le plus récent, un numéro de bloc et une valeur de difficulté. S'il existe des transactions en conflit sur le réseau (par exemple, des transactions qui doublent les dépenses), une seule d'entre elles est sélectionnée pour faire partie du bloc. Les blocs sont ajoutés à la Blockchain à intervalles régulières.



10. Figure: Blocs dans la Blockchain

La figure 10 illustre les liens entre les blocs de la chaîne de blocs. Les blocs marqués en noir indiquent la blockchain active en cours. Ceux qui sont marqués en gris sont appelés des blocs rassis.

La Blockchain est techniquement une liste de blocs ordonnée et horodatée, qui fournit un enregistrement de toutes les transactions qui ont eu lieu. Chacun de ces blocs est lié au précédent, à savoir son parent, par un hachage unique. Ces hachages sont générés via l'algorithme de hachage SHA256. En d'autres termes, l'en-tête de chaque bloc contient une référence au hachage de son parent. Cette liaison se poursuit jusqu'au premier bloc de la blockchain, également appelé bloc de genèse. Le bloc de genèse est le bloc bleu le plus à gauche indiqué à la figure 8. Un bloc peut avoir plusieurs enfants simultanément. Chaque enfant fait référence au hachage du même bloc parent. En fin de compte, l'un de ces blocs enfants devient la partie de la blockchain principale. Ce phénomène est connu sous le nom de forking. Cela se produit lorsque plusieurs mineurs extraient et vérifient différents blocs au même moment. La blockchain est également appelée immuable car c'est une dépense astronomique de recalculer les hachages de tous les blocs à partir du bloc de genèse.

Le tableau 5 décrit plus en détail la structure de l'entête du bloc, en expliquant les différents types de métadonnées associées aux blocs

| Size (bytes) | Field               | Description   |
|--------------|---------------------|---|
| 4            | Version             | Software version  |
| 32           | Previous Block Hash | Reference to hash of the previous block   |
| 32           | Merkle Root         | Hash of the root of the merkle tree containing all the transactions included in the block |
| 4            | Timestamp           | Approximate time when the block was created   |
| 4            | Difficulty Target   | Proof of work difficulty for the block  |
| 4            | Nonce               | Proof of work counter   |

5. Table: Structure de l'entete d'un block

Le champ racine merkle fait référence à la racine d'un arbre à distorsion qui stocke les informations de transaction dans chaque bloc.

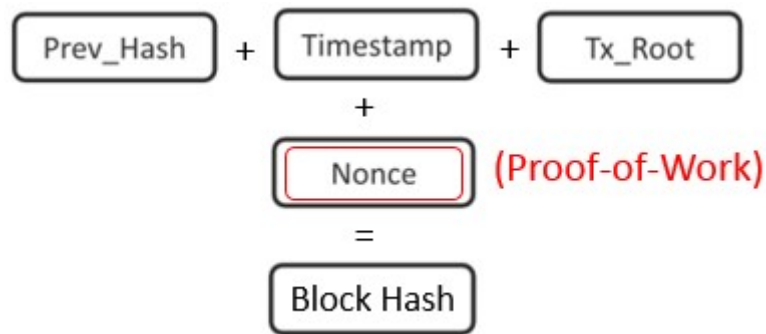
Puisque l'entête de bloc fait partie du bloc, un bloc complet, y compris toutes les informations de transaction, a une taille beaucoup plus grande qu'un en-tête de bloc. C'est la raison pour laquelle les clients et les portefeuilles SPV téléchargent uniquement les fichiers d'en-tête de la blockchain tout en récupérant les informations de leur bloc souhaité des

nœuds complets connectés au réseau. Le tableau 8 décrit les différents champs d'un bloc, ainsi que leur taille.

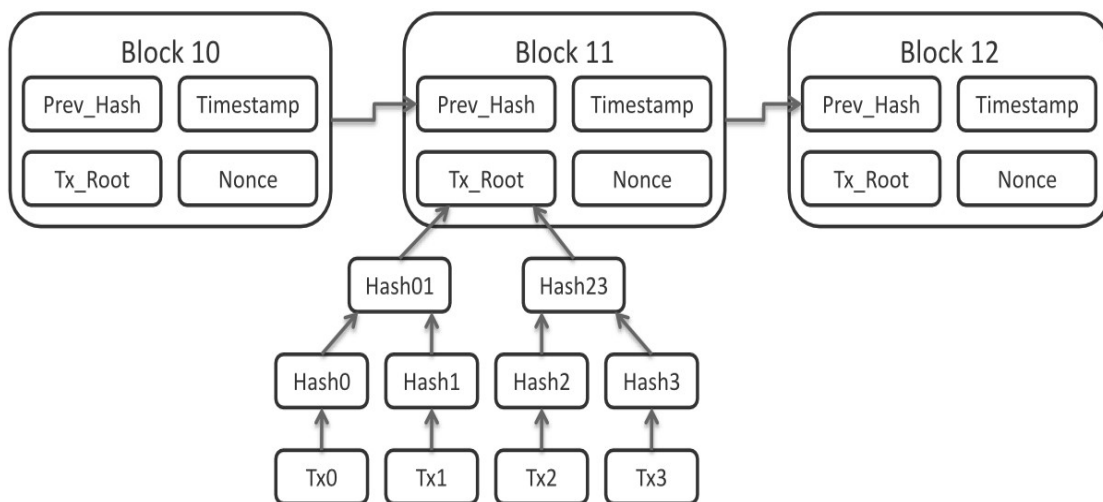
| Size (bytes)    | Field               | Description                                     |
|-----------------|---------------------|---|
| <b>4</b>        | Block Size          | The size of the block                           |
| <b>80</b>       | Block Header        | Consists of several fields, as shown in Table 5 |
| <b>1-9</b>      | Transaction Counter | Number of transactions included in the block    |
| <b>Variable</b> | Transactions        | Recorded transaction included in the block      |

*6. Table: Structure d'un bloc*

Le hachage précédent (hachage du bloc précédent) est présent pour assurer un lien entre les blocs. Chaque bloc peut être identifié de manière unique par son hachage de bloc qui est généré à partir des quatre paramètres de l'en-tête de bloc (voir diagramme). Maintenant, puisqu'il s'agit d'un réseau d'égal à égal, chaque homologue peut ne pas recevoir les transactions dans le même ordre. Donc, en fonction de leur arrangement de transactions, le hachage de bloc pourrait être différent. Un seul de ces blocs peut être ajouté à la blockchain car un seul enregistrement d'une transaction peut exister. En outre, chaque pair ne peut pas avoir sa propre version d'un bloc car cela met tout le système en jeu. Il devrait donc y avoir un moyen pour tous les pairs de s'accorder sur un bloc unique comme valide, de sorte que seul ce bloc soit ajouté à la blockchain. Pour atteindre ce consensus, le bitcoin a pour règle que le hachage de bloc doit avoir un certain nombre de zéros non significatifs. Ainsi, tous les nœuds ont trois entrées: le hachage précédent (Prev\_Hash), l'horodatage et la racine de la transaction (Tx\_Root). Ils savent également que la sortie doit être dans un certain format. Maintenant, ils continuent à ajouter une valeur de nonce aléatoire aux entrées jusqu'à ce qu'elles parviennent à un hachage cible. En raison de la nature de la fonction de hachage, il est impossible de deviner cette valeur de nonce. Le seul processus permettant de le savoir consiste à utiliser la méthode des essais et des erreurs. La difficulté est exponentiellement proportionnelle au nombre de zéros au début du bloc. Cette valeur nonce est très difficile à trouver et nécessite souvent d'effectuer des milliards de hachages. Ce processus n'est pas économe en énergie et nécessite du matériel spécial avec une énorme capacité de calcul. Une fois que cette valeur de nonce est trouvée, le nœud diffuse l'ensemble du bloc vers le réseau et il est très facile pour les autres nœuds de vérifier s'il s'agit du nonce correct. Cette valeur de nonce prouve qu'un nœud a effectué un travail considérable avant de générer un bloc, d'où le nom Preuve de travail.



11. Figure: Contenus d'un hachage de Bloc



12. Figure: Preuve de Travail

Après vérification, ce bloc est ajouté à la blockchain et tous les nœuds commencent à travailler sur le prochain ensemble de transactions.

Ce mécanisme de preuve de travail est ce qui rend la blockchain immuable. Imaginons un scénario dans lequel un pirate informatique voudrait supprimer une transaction (Tx1) du bloc 11, comme indiqué dans le diagramme ci-dessus. Ici, tous les autres nœuds travaillent sur la génération du bloc 13. Ainsi, si le pirate informatique souhaite supprimer Tx1, la racine Tx\_Root du bloc 11 change et la valeur de nonce précédente n'est plus valide. Le pirate informatique doit effectuer à nouveau ces milliards de calculs et découvrir une nouvelle valeur de nonce. Etant donné que le hachage du bloc 11 (Prev\_Hash) est également utilisé lors du calcul du hachage du bloc 12, le processus complet doit être répété pour rechercher une nouvelle valeur de nonce pour le bloc 12. Il ne s'arrête pas là car le pirate doit également créer un bloc 13 avant tout autre nœud, car tous les nœuds suivent toujours la chaîne la plus longue puisqu'il s'agit de la chaîne dans laquelle la majorité de la puissance du processeur est investie. En bref, un seul nœud doit modifier 2 blocs et créer un nouveau bloc tandis que tous les autres nœuds tentent de générer ce nouveau bloc uniquement. Ainsi, il est impossible pour un pirate informatique de modifier les données présentes sur la blockchain. L'ensemble de ce mécanisme de validation de travail et de génération de blocs maintient la chaîne de blocs sécurisée, transparente et immuable.



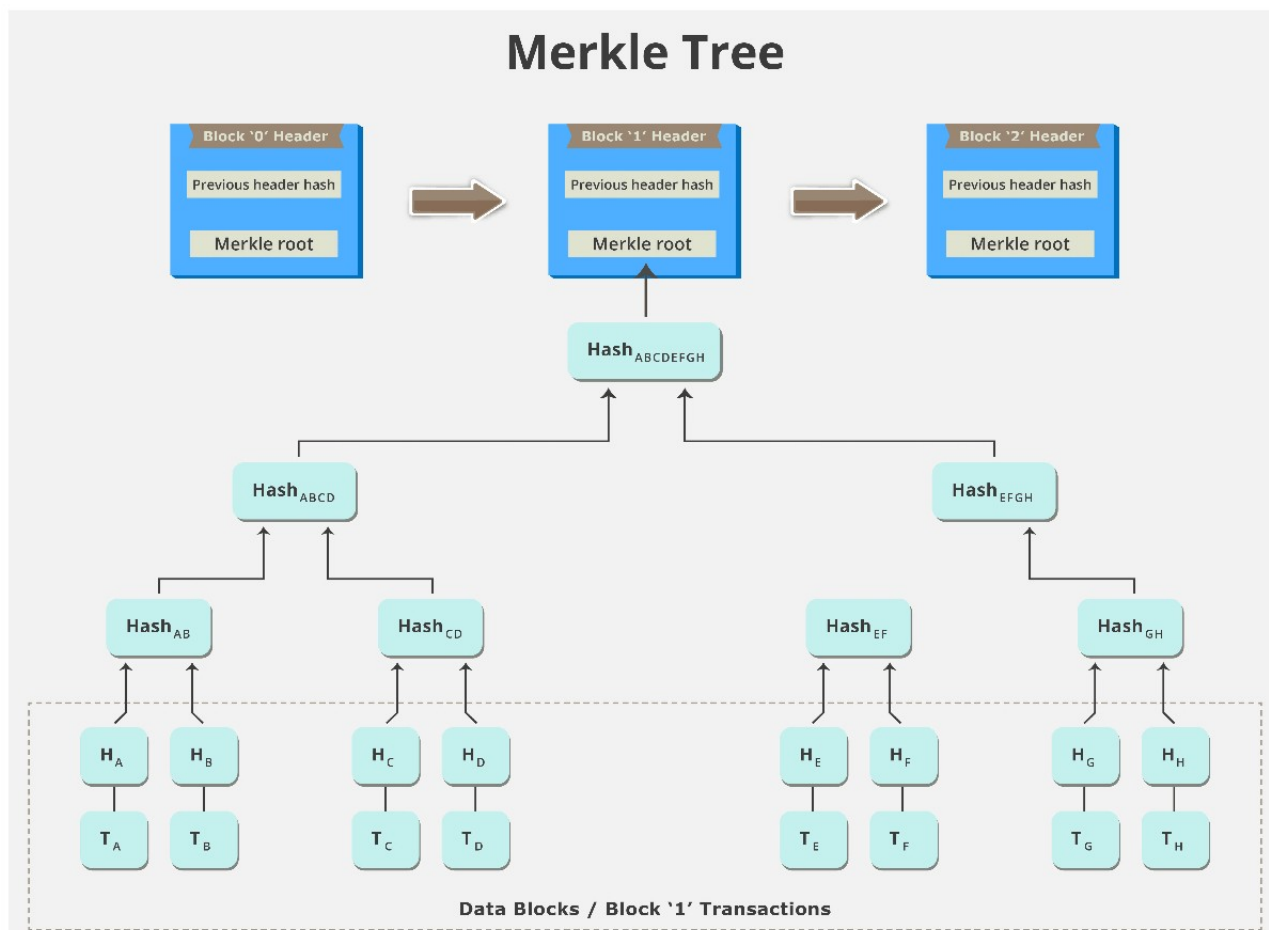
➤ **Hachage** est l'équivalent du concept d'empreinte digitale, c'est l'identifiant unique d'une personne. Lorsque deux objets sont envoyés à la Blockchain, ils ont des hachages différents. De plus, si vous avez l'objet, il est très facile de créer le hachage, mais il est presque impossible d'effectuer l'opération inverse. Pour s'assurer que l'inverse n'existe pas, la sortie est plus courte que les entrées et il y a plus de deux entrées qui donnent la même sortie. Ce mode de fonctionnement rend impossible le calcul de l'inverse.

➤ **Hachage du bloc et sa hauteur** Il y a deux façons d'identifier un bloc ; Tout d'abord, par son hash. Ce hachage est calculé par les nœuds homologues du réseau chaque fois qu'un bloc est généré. Le hachage pourrait être stocké dans une base de données incluse dans les métadonnées du bloc pour une indexation et une récupération plus rapide des blocs du disque.

La deuxième façon d'identifier un bloc serait par sa hauteur. Le bloc de genèse est à la hauteur 0. Cette méthode d'identification n'est pas absolue car deux blocs ou plus de la chaîne de blocs peuvent avoir la même hauteur et il est également possible que deux blocs de même hauteur aient le même parent.

➤ **Merkle Tree** Un type spécial de structure de stockage de données basé sur des fonctions de hachage est appelé arbre de Merkle :

- Il est structuré comme un arbre binaire ; les feuilles contiennent les valeurs à stocker et chaque nœud interne est le hachage de ses deux enfants.
- Il fournit des recherches efficaces et une protection contre la falsification, car la vérification d'une transaction est incluse dans l'arborescence. Peut être accompli en envoyant uniquement la transaction, le hachage contenu dans chaque nœud entre le nœud feuille de transaction et la racine, ainsi que les valeurs de hachage utilisées pour créer chaque hachage envoyé.
- La recherche d'une transaction dans une arborescence Merkle à trois niveaux inclut l'envoi de deux transactions (celle souhaitée et l'autre enfant de son parent) et de trois hachages (le parent de la transaction, la racine et l'autre enfant de la racine).



13. Figure: Arbre de Merkle

- **Minage** qui fait référence au processus de revue de calcul distribué effectué sur chaque "bloc" de données dans une chaîne de blocs. Cela permet de parvenir à un consensus dans un environnement où personne ne se connaît.

L'extraction minière présente deux avantages majeurs :

Le réseau de chaînes de blocs, à savoir la validation et la vérification des transactions. L'exploitation génère également de nouvelles pièces de monnaie numériques sur le réseau, car ces pièces nouvellement frappées servent de récompense au mineur qui résout le prochain bloc de la blockchain. La première étape de l'extraction consiste à calculer le niveau de difficulté de la blockchain. Tous les nœuds complets connectés au réseau de chaînes de blocs recalculent ce niveau de difficulté après certains intervalles. Le niveau peut augmenter ou diminuer en fonction du temps nécessaire pour générer un certain intervalle de blocs. Dans le cas de Bitcoins, qui était la première implémentation de blockchains, l'intervalle est de 2016 blocs. Ainsi, les nœuds complets doivent réévaluer le niveau de difficulté après chaque bloc 2016, ce qui donne un temps de consensus moyen de 10 minutes. À mesure que le nombre de mineurs augmente, le taux de création de blocs augmente également. Ceci entraîne à son tour une augmentation du niveau de difficulté, car il réduit le taux de création de blocs à 10 minutes, dans le cas de Bitcoin. Le mineur télécharge ensuite toutes les transactions et les

informations de blocage qui se sont déroulées précédemment, et construit un chemin de merkle à partir de celles-ci, générant finalement une racine de mot clé.

- **Mécanisme de consensus** est un problème fondamental dans les systèmes distribués qui nécessite que deux agents ou plus se mettent d'accord sur une valeur donnée nécessaire à des fins de calcul. Certains de ces agents peuvent être peu fiables, et le processus de consensus doit donc être dépendant. Ainsi, la nécessité de mécanismes de consensus est de faciliter la mise à jour sécurisée d'un processus ou d'un état, conformément à certaines règles de transition d'état, dans lesquelles un ensemble distribué a le droit d'effectuer les transitions d'état.

Un consensus est un processus qui permet à un ensemble de processus répartis de parvenir à un accord sur une valeur ou une action en dépit d'un certain nombre de processus défaillants (Correia, 2011). Blockchain nécessite vérification et acceptation par tous les membres du réseau, généralement appelée consensus.

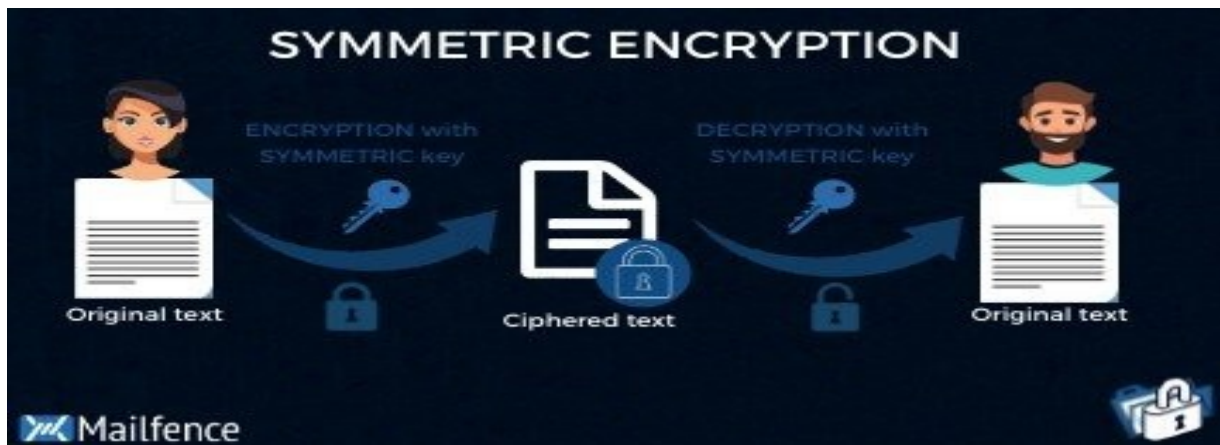
Pour parvenir à un consensus dans le mécanisme distribué, quatre algorithmes peuvent être appliqués. Dans un réseau de blockchain, un consensus est utilisé pour éviter que des acteurs mensongers ne provoquent informations potentiellement non valides dans la base de données (Swanson, 2015).

Le mécanisme de consensus concret utilisé pour une blockchain donnée dépend d'un certain nombre de y compris le degré de confiance entre les parties et l'alignement de leurs intrigues, ainsi que en tant que facteurs concernant la forme et la synchronisation du réseau (Correia, 2011).

- **Attaque 51%**

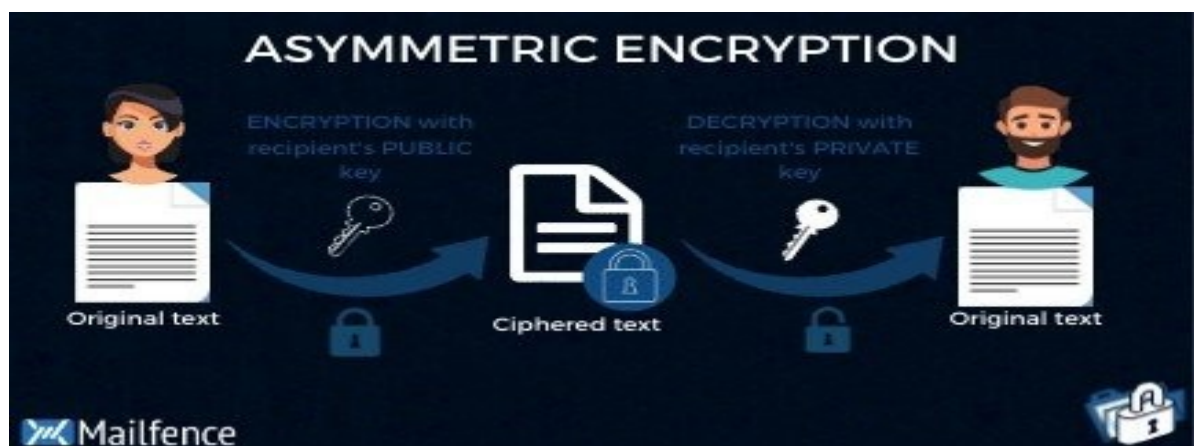
Outre le fait que l'algorithme de preuve de travail consomme beaucoup d'énergie, il présente un autre inconvénient : l'attaque à 51%. Si une seule entité devait contribuer à plus de 51% des activités du réseau Bitcoin, elle serait en mesure de contrôler totalement le réseau et de modifier le grand livre en fonction de leurs besoins. Bien que cette attaque soit théoriquement possible, elle coûterait aux mineurs une énorme somme d'argent ainsi que de la puissance de calcul

- **La cryptographie symétrique** désigne le processus d'utilisation d'une seule clé pour le chiffrement et le déchiffrement. Cela signifie que la même clé devrait être disponible pour plusieurs personnes si elles veulent échanger des messages en utilisant cette forme de cryptographie.



14. Figure: Chiffrement symétrique

- **La cryptographie asymétrique** désigne le processus d'utilisation de deux clés pour le chiffrement et le déchiffrement. Toute clé peut être utilisée pour le chiffrement et le déchiffrement. Le chiffrement des messages à l'aide d'une clé publique peut être déchiffré à l'aide d'une clé privée et les messages chiffrés par une clé privée peuvent être déchiffrés à l'aide d'une clé publique. Comprenons cela à l'aide d'un exemple. Tom utilise la clé publique d'Alice pour crypter les messages et l'envoie à Alice. Alice peut utiliser sa clé privée pour décrypter le message et en extraire le contenu. Les messages cryptés avec la clé publique d'Alice ne peuvent être décryptés que par Alice car elle est la seule à détenir sa clé privée et personne d'autre. C'est le cas général des clés asymétriques. Il y a une autre utilisation que nous verrons en discutant des signatures numériques.



15. Figure: Chiffrement symétrique

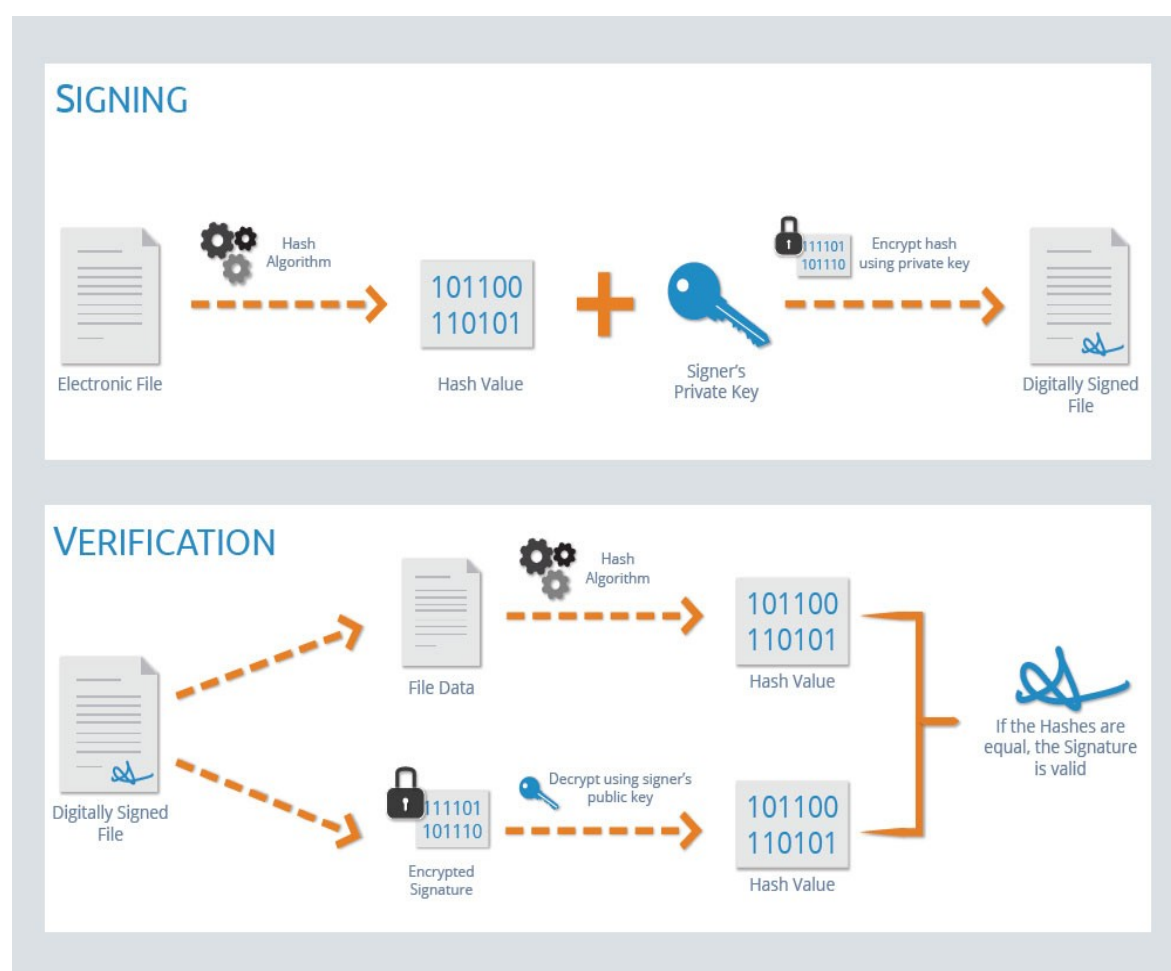
➤ **Signature numérique**

Plus tôt, nous avons discuté de la cryptographie à l'aide de clés asymétriques. L'un des cas importants pour l'utilisation de clés asymétriques est la création et la vérification d'une signature numérique. Les signatures numériques sont très semblables à une signature faite

par une personne sur une feuille de papier. Semblable à une signature papier, une signature numérique aide à identifier une personne. Il aide également à s'assurer que les messages ne sont pas altérés en transit. Comprenons les signatures numériques à l'aide d'un exemple.

Alice veut envoyer un message à Tom. Comment Tom peut-il identifier et s'assurer que le message vient d'Alice seulement et que le message n'a pas été modifié ou altéré en transit? Au lieu d'envoyer un message brut / transaction, Alice crée un hash de la charge utile entière et crypte le hash avec sa clé privée. Elle ajoute la signature numérique résultante au hachage et la transmet à Tom. Lorsque la transaction atteint Tom, il extrait la signature numérique et la décrypte en utilisant la clé publique d'Alice pour trouver le hash original. Il extrait également le hachage original du reste du message et compare les deux hachages. Si les hachages correspondent, cela signifie qu'ils proviennent en fait d'Alice et qu'ils n'ont pas été altérés.

Les signatures numériques sont utilisées pour signer les données de transaction par le propriétaire de l'actif ou de la cryptomonnaie, comme Ether.





16. Figure: Signature numérique

- **Un nonce** est un nombre aléatoire ou semi-aléatoire qui est généré pour une utilisation spécifique. Il est lié à la communication cryptographique et à la technologie de l'information

(TI). Le terme signifie "nombre utilisé une fois" ou "nombre une fois" et est communément appelé nonce cryptographique.

En règle générale, un nonce est une valeur qui varie avec le temps pour vérifier que des valeurs spécifiques ne sont pas réutilisées. Un nonce peut être un horodatage, un compteur de visite sur une page Web ou un marqueur spécial destiné à limiter ou empêcher la reproduction non autorisée d'un fichier.

- **Ethereum** est une plateforme basée sur la technologie blockchain qui permet aux développeurs de réaliser et déployer des applications décentralisées ou DApps (pour decentralized applications). Alors que le rôle principal de Bitcoin est de transférer de la monnaie virtuelle, celui d'Ethereum est de faire fonctionner le programme de n'importe quelle application décentralisée. Au lieu d'investir dans des serveurs, les développeurs vont mettre l'application sur le réseau Ethereum.

|  VS  |  | Bitcoin             | Ethereum                            |
|--|--|---------------------|-------------------------------------|
| Founder  |  | Satoshi Nakamoto    | Vitalik Buterin                     |
| Release Date   |  | 9 Jan 2008          | 30 July 2015                        |
| Release Method   |  | Genesis Block Mined | Presale                             |
| Blockchain   |  | Proof of work       | Proof of work (Planning for POS)    |
| Useage   |  | Digital Currency    | Smart Contracts<br>Digital Currency |
| Cryptocurrency Used  |  | Bitcoin(Satoshi)    | Ether                               |
| Algorithm  |  | SHA-256             | Ethash                              |
| Blocks Time  |  | 10 Mintues          | 12-14 Seconds                       |
| Mining   |  | ASIC miners         | GPUs                                |
| Scalable   |  | Not now             | Yes                                 |

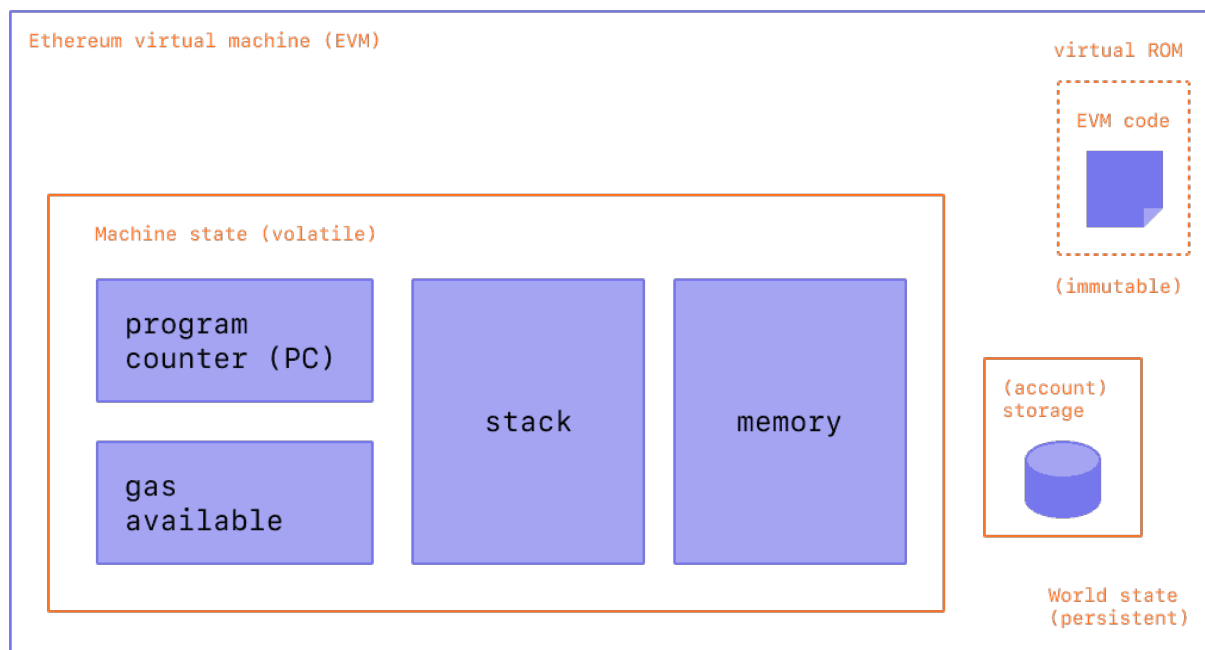
7. Table: difference entre ethereum et bitcoin

- **L'Ethereum Virtual Machine (EVM)** est en quelque sorte le cerveau de la blockchain Ethereum. Cette entité permet au réseau d'être continuellement mis à jour selon les transactions et interactions incluses dans les blocs.

Sans l'EVM, la blockchain Ethereum ne serait qu'un réseau épars dans lequel chaque nœud pourrait avoir une version différente de la blockchain, avec des transactions et des comptes disparates.

L'EVM est un système ou un environnement dit « Turing complet », signifiant que celui-ci possède la puissance de calcul de la machine Turing. Autrement dit, l'EVM est capable de réaliser des opérations telles que des calculs simples, des récursions, des comparaisons, des modifications de variables, etc.

L'EVM est également ce qui permet à la blockchain Ethereum de traiter des smart contracts, et donc d'héberger des applications décentralisées complexes. C'est ce qui différencie les blockchains d'infrastructure comme Ethereum des autres blockchains comme Bitcoin ou Monero.



17. Figure: Ethereum Virtual machine (EVM)

➤ **Contrat Intelligent** est simplement un programme exécuté sur la blockchain d'Ethereum. C'est un ensemble de code (ses fonctions) et de données (son état) qui réside à une adresse spécifique sur la blockchain Ethereum.

Le contrat intelligent est un type de compte Ethereum. Cela signifie qu'il dispose d'un solde et peut envoyer des transactions sur le réseau. Cependant, il n'est pas contrôlé par un utilisateur, mais est plutôt déployé et exécuté comme un programme. Les comptes des utilisateurs peuvent ensuite interagir avec un contrat intelligent en soumettant des transactions qui exécutent une fonction définie sur le contrat intelligent. Un contrat intelligent peut définir des règles, comme un contrat normal, et les appliquer automatiquement via le code. Les contrats intelligents ne peuvent pas être supprimés par défaut et les interactions avec eux sont irréversibles.

- **Ether** est la devise utilisée dans le réseau Ethereum. Contrairement aux bitcoins, les éthers n'ont pas été créés pour devenir une monnaie numérique mondiale décentralisée et leurs aspirations vont au-delà de l'envoi ou du transfert d'argent.
  
- **Gas** est l'unité qui mesure la quantité d'efforts de calculs requis pour exécuter des opérations spécifiques sur le réseau Ethereum.  
  
Comme chaque transaction Ethereum nécessite des ressources informatiques pour s'exécuter, cela implique des frais. Le gaz correspond aux frais requis pour effectuer correctement une transaction sur Ethereum.
  
- **Dapp (application décentralisée)** est une application qui fournit une interface aux contrats intelligents. Un exemple de Dapp pourrait être une application crypto-devise.
  
- **L'ERC-20** introduit une norme pour les jetons fongibles. En d'autres termes, ils disposent d'une propriété qui fait que chaque jeton est exactement le même (en termes de type et de valeur) qu'un autre jeton. Par exemple, un jeton ERC-20 agit exactement comme de l'ETH, ce qui signifie que 1 jeton est et sera toujours égal à tous les autres jetons.
  
- **L'ERC-721** introduit une norme pour les NFT. En d'autres termes, ce type de jeton est unique et peut avoir une valeur différente de celle d'un autre jeton du même contrat intelligent, peut-être en raison de son âge, de sa rareté ou du visuel qui lui est associé.

## 2.8 Types de la Blockchain

La blockchain peut être "avec permission" (privée) ou "sans permission" (publique). La première catégorie impose des restrictions aux contributeurs du consensus. Seul ceux de confiance et choisis qui ont le droit de valider des transactions. Elle ne nécessite pas beaucoup de calcul pour atteindre un consensus, ainsi, elle est économique en termes de temps d'exécution et en énergie. Généralement les transactions sont privées et ne sont accessibles que par les objets autorisés. La deuxième catégorie (blockchain publique) utilise un nombre illimité d'objets anonymes. En se basant sur la cryptographie, chaque acteur peut communiquer d'une manière sécurisée. Chaque objet est représenté par une paire de clés (publique/privée), et a le droit de lire, d'écrire et de valider des transactions dans la blockchain. La blockchain est sûre si 51% des objets (ou plus) sont honnêtes et lorsque le consensus du réseau est atteint. Généralement, les blockchains sans permission consomment beaucoup d'énergie et de temps, car elles exigent un montant de calcul pour renforcer la sécurité du système (ex. en utilisant la PoW).



Il existe trois types de Blockchain, selon leur mode de fonctionnement :

- **Blockchain public** : Tout le monde peut lire ou écrire des données et la seule condition est de disposer d'un ordinateur et d'une connexion Internet. Une partie de ce type de réseau restreint l'accès uniquement en lecture ou en écriture. Ethereum et Bitcoin sont des exemples qui utilisent une approche où tout le monde peut écrire.
- **Blockchain privée** : N'est pas ouvert au public, mais est accessible uniquement sur invitation et tous les membres participants se connaissent et se font confiance. Ceci est très utile lorsque la Blockchain est utilisée entre entreprises appartenant à la même branche. Parmi les plus célèbres, citons Hyperledger (de Linux Foundation) et Ripple (protocole permettant les transferts internationaux).
- **Blockchain permissionnée** : Aussi connu sous le nom de Consortium Blockchain, est un hybride entre Blockchain publique et privée. Dans ce type, seuls quelques nœuds sélectionnés sont prédéterminés et les nœuds participants sont invités, mais toutes les transactions sont publiques. Cela signifie que les nœuds participent à la maintenance et à la sécurité de ce réseau, mais que toutes les transactions sont visibles pour les utilisateurs du monde entier. Le droit de lecture peut être public ou limité aux participants. Les Blockchains du consortium préservent la confidentialité des données, comme les Blockchains privés. BigchainDB est un exemple de consortium Blockchain.

## 2.9 Avantages de la technologie Blockchain

Les chaînes de blocs peuvent renforcer la sécurité principalement sur trois aspects : le blocage du vol d'identité, la prévention de la manipulation des données et l'arrêt des attaques par déni de service .

### ➤ **Blocage de vol d'identité**

La structure de la preuve de travail du mineur de réseau de Blockchain et son grand livre distribué de transactions de données réduisent les risques de vol et de corruption des données.

### ➤ **Prévenir la manipulation et la fraude des données**

Dans la technologie Blockchain, la cryptographie, le hachage et une structure décentralisée empêchent quasiment tout membre de modifier les données du grand livre. Cela empêche et détecte toute forme de manipulation et permet aux organisations de maintenir la protection des informations. Une solution importante qui a été développée pour éviter la fraude et la manipulation est KSI (Keyless Signature Infrastructure), qui assure la protection des réseaux ainsi que la sécurité et la confidentialité des données.

Avec KSI, personne ne peut manipuler les données et l'authenticité des données électroniques peut être prouvée mathématiquement. KSI stocke les signatures numériques des fichiers originaux dans une Blockchain, puis vérifie les copies en revérifiant les signatures des copies par rapport à celles stockées dans la Blockchain. Si une quelconque

manipulation est effectuée, elle est détectée très rapidement car les hachages stockés dans la chaîne de caractères résident dans des milliers de nœuds. KSI Technology est utilisée activement dans les secteurs de l'aérospatiale et de la défense, ainsi que dans le secteur de la santé, afin de mieux contrôler le dossier médical du patient.

### ➤ **Prévention des attaques par déni de service distribué**

Il existe un grand nombre d'infrastructures critiques à protéger. Blockchain peut aider avec DNS (Domain Name System) qui fournit un accès à des sites Web utilisant des noms de domaine plutôt que des adresses IP. Le système DNS est dangereusement centralisé dans quelques serveurs racine sous le contrôle de l'ICANN (Internet Corporation for Assigned Names and Numbers), qui est responsable des adresses de protocole IP, des identificateurs de protocole, des fonctions de gestion de système de domaine et de la gestion de système du serveur racine. Blockchain pourrait créer un DNS distribué, beaucoup plus transparent, rendant pratiquement impossible la manipulation des enregistrements par une seule entité.

Il existe certaines différences entre les réseaux Blockchain et le paradigme du Cloud Computing. Dans le modèle en nuage, les périphériques IoT sont identifiés, authentifiés et connectés via des serveurs en nuage, où le traitement et le stockage sont souvent effectués. Les réseaux IoT ayant des coûts élevés sont concernés par le modèle de cloud centralisé. Les appareils IoT sont vulnérables aux attaques DDoS, au vol de données, au piratage et au piratage à distance. Si un périphérique IoT connecté à un serveur fait l'objet d'une violation, toutes les personnes connectées au serveur pourraient être affectées. En outre, le modèle de nuage centralisé est sujet à la manipulation. Les données collectées ne garantissent pas que les informations sont utilisées de manière appropriée. Blockchain peut éliminer ces problèmes de Cloud Computing. Dans Blockchain, les échanges de messages entre périphériques peuvent être traités de la même manière que les transactions financières dans un réseau bitcoin. Les appareils reposent sur des contrats intelligents qui garantissent plus de sécurité. Le fait que Blockchain vérifie de manière cryptographique les transactions signées, elle élimine la possibilité d'attaque par interférence, de rejeu ou d'autres attaques.

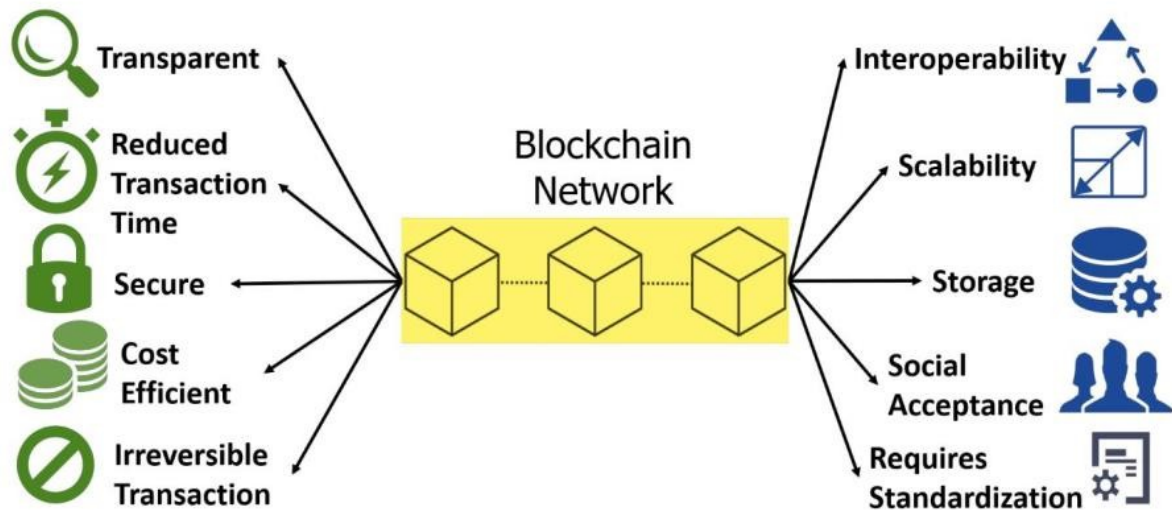
## **2.10 Défis de la Blockchain**

La blockchain est une technologie émergente qui se répand dans divers secteurs et qui présente un grand nombre d'avantages et d'opportunités. Cependant, cette technologie présente son propre ensemble de défis à relever (voir figure 43). Quelques-uns de ces défis majeurs sont abordés dans cette section.

## OPPORTUNITIES

VS

## CHALLENGES



18. Figure: Opportunités et défis des blockchains

### ➤ Sécurité et confidentialité des données

Le premier et le plus important défi concerne la sécurité et la confidentialité des données. Avec la mise en œuvre d'applications basées sur la technologie de la blockchain, la nécessité pour un tiers d'effectuer une transaction est éliminée. Étant donné que le mécanisme de blockchain permet à l'ensemble de la communauté, plutôt qu'à un seul tiers de confiance, de vérifier les enregistrements dans une architecture de blockchain, les données sont exposées à des risques potentiels en matière de sécurité et de confidentialité. Étant donné que tous les nœuds peuvent accéder aux données transmises par un nœud, la confidentialité des données ne sera pas active. En cas d'absence d'une tierce partie pour autorisation, le patient doit sélectionner un ou plusieurs représentants qui peuvent accéder à ses informations et / ou à ses antécédents médicaux en son nom, en cas d'urgence. Désormais, ce représentant peut également autoriser un ensemble de personnes à accéder aux enregistrements du même patient, ce qui peut créer une menace énorme pour la sécurité et la confidentialité des données. L'implication de mécanismes de haute sécurité dans les données entraînera à son tour des obstacles pour le transfert des données d'un bloc à un autre et, par conséquent, les destinataires auront accès à des données limitées ou incomplètes. En outre, les réseaux blockchain sont sujets à une sorte de violation de la sécurité connu sous le nom d'attaque 51%. Cette attaque implique une équipe de mineurs qui possèdent plus de 50% des blocs d'un réseau blockchain. Les mineurs obtiennent une autorité du réseau et pourraient empêcher toute nouvelle transaction en ne leur donnant pas leur consentement. Cinq crypto-monnaies ont récemment été victimes de cette attaque. En outre, un dossier patient peut contenir des données sensibles qui ne conviennent pas pour figurer dans la chaîne de blocs.

### ➤ Gestion de la capacité de stockage

Un autre défi qui apparaît sur ce front est la gestion de la capacité de stockage. La Blockchain a été conçue pour enregistrer et traiter les données de transaction, qui ont une portée limitée, de sorte qu'elle n'a pas besoin de beaucoup de stockage. Avec le temps, au fur et à mesure de son expansion dans le domaine de la santé, les défis du stockage devinrent évidents. Le secteur de la santé contient une grande quantité de données qui doivent être traitées quotidiennement. Des dossiers des patients aux antécédents médicaux, en passant par les rapports de test, en passant par les analyses IRM, les rayons X et autres images médicales, toutes les données du scénario de la blockchain seront disponibles pour tous les nœuds de la chaîne, ce qui nécessite un espace de stockage considérable. De plus, les applications de la blockchain étant basées sur des transactions, les bases de données utilisées pour cette technologie ont tendance à se développer rapidement. En raison de la taille croissante des bases de données, la vitesse de recherche et d'accès à l'enregistrement devient lente, ce qui est tout à fait inadéquate pour les types de transactions pour lesquels la rapidité est essentielle. Par conséquent, une solution de chaîne de blocs doit être évolutive et résiliente.

➤ **Problèmes d'interopérabilité**

La blockchain souffre également du problème de l'interopérabilité, c'est-à-dire que les chaînes de blocs de divers fournisseurs et services de communication communiquent entre elles de manière transparente et appropriée. Ce défi crée des obstacles au partage efficace des données.

➤ **Défis de la normalisation**

La technologie de la blockchain en est encore à ses balbutiements et elle sera donc certainement confrontée à des problèmes de standardisation en vue de son application pratique en médecine et en soins de santé. Un certain nombre de normes bien authentifiées et certifiées seraient exigées des autorités internationales de normalisation. Ces normes prédéfinies seraient utiles pour évaluer la taille, la nature des données et le format des informations échangées dans les applications blockchain. Ces normes examineront non seulement les données partagées, mais devront également servir de mesures de sécurité préventives.

➤ **Défis sociaux**

La technologie des chaînes de blocs évolue toujours et fait donc face à des défis sociaux, tels que le changement de culture, en plus des défis techniques susmentionnés. Accepter et adopter une technologie complètement différente des méthodes de travail traditionnelles n'est jamais chose facile. Bien que l'industrie médicale s'achemine lentement vers la numérisation, il lui reste encore beaucoup à faire pour passer complètement à cette technologie, en particulier celle comme la blockchain, qui n'a pas encore été validée sur le plan clinique. Il faudra du temps et des efforts pour convaincre les médecins de passer de la paperasserie à la technologie. En raison de son faible taux d'adoption dans le secteur de la santé, la technologie et les politiques proposées sont relativement peu fiables. En raison de

tous ces défis et menaces, nous ne pouvons pas, à ce jour, le qualifier de solution viable et universelle pour tous les problèmes de santé.

## **Conclusion**

La technologie Blockchain est révolutionnaire. Elle va rendre la vie plus simple et plus sûre, en changeant la façon dont les informations personnelles sont stockées et dont les transactions de biens et de services sont effectuées.

## **Chapitre 3 :Analyse et Conception**

Ce chapitre présente le diagramme de cas d'utilisation suivie des diagrammes de séquences qui explique en détails chaque cas d'utilisation et finalement le diagramme de classe pour une vue globale des classes du système.

# 1. Uml

**UML**, c'est l'acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ». La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc .



## 2. Analyse des besoins

### 2.1. Acteurs principaux du système

Les acteurs présentent les entités externes au système qui interagissent directement avec lui. L'acteur peut être humain, matériel...

Les acteurs de notre système et leurs rôles :

- **Client**
- **Vendeur E-commerce**
- **Administrateur**

### 2.2. Identification des Rôle

| Client             |  |
|--------------------|--|
| description        | La personne qui l'intention d'acheter en ligne |
| Rôle               | ➔ Paiement                                     |
| Vendeur E-commerce |  |
| description        | La personne qui possède un site e-commerce     |

|                       |  |
|-----------------------|--|
| <b>Rôle</b>           | → Inscription<br>→ Gestion des intégrations<br>→ Consultation des paiements<br>→ Consultation de documentation<br>→ Création du portefeuille<br>→ Réalisation des transactions |
| <b>Administrateur</b> |  |
| <b>description</b>    | C'est l'individu qui contrôle et gère le système   |
| <b>Rôle</b>           | → Gestion des Clients<br>→ Consultation  |

8. Table: Identification des Rôle

## 2.3. Diagramme de cas d'utilisation

### 2.3.1. Définition

Diagramme de cas d'utilisation montre le comportement attendu d'un système, il est le premier diagramme construit lors du développement d'un projet pour clarifier et structurer les besoins des utilisateurs et les objectifs à atteindre pour le système.

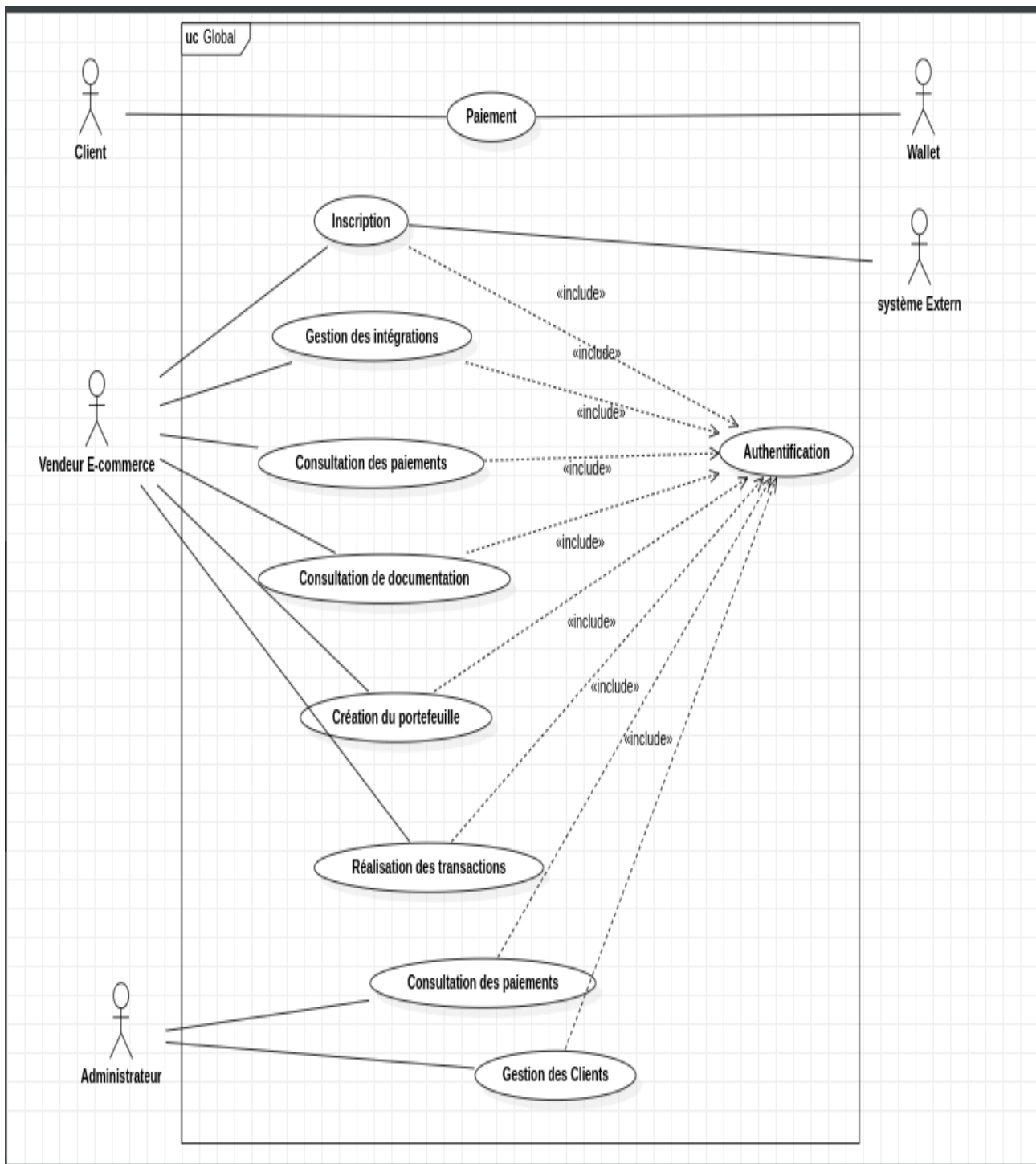
Les utilités des diagrammes de cas d'utilisation :

- Déterminer les interactions du système.
- Déterminer les fonctionnalités du système.
- Déterminer les dépendances en entre les fonctionnalités.

La figure 14 sise ci-dessous présente le digramme globale des cas d'utilisation de notre projet.

### 2.3.2 Diagramme des cas d'utilisations globale

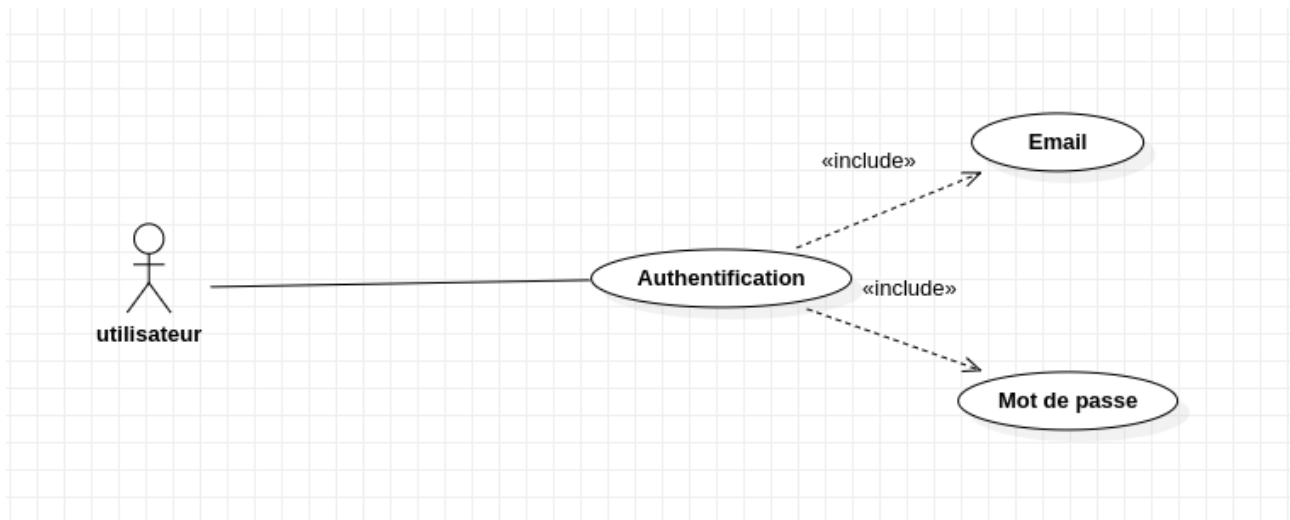




19. Figure: Diagramme des cas d'utilisation globale

A l'issue de l'expression des besoins à l'aide du diagramme des cas d'utilisation globale, dans ce qui suit, nous détaillons chacun des cas d'utilisations présenté dans **la figure 15** jusqu'à **la figure 19**, en donnant sa description textuelle.

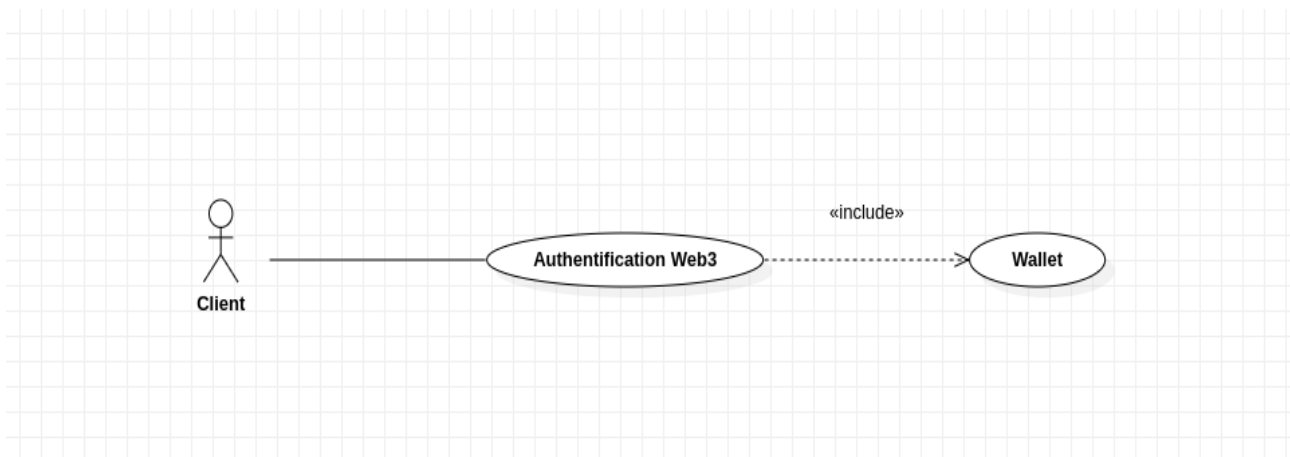
### 2.3.3 Cas d'utilisation Authentification



20. Figure: Diagramme de cas d'utilisation s'authentifier

|                     |  |
|---------------------|--|
| Acteur principal    | <ul style="list-style-type: none"> <li>➤ Vendeur e-commerce</li> <li>➤ Administrateur</li> </ul>   |
| Objectif            | S'assurer que l'utilisateur est bien celui qui prétant être  |
| Scenario nominal    | <ol style="list-style-type: none"> <li>1. L'utilisateur saisit son e-mail et son mot de passe .</li> <li>2. Le système vérifie l'e-mail et le mot de passe .</li> <li>3. Le système affiche l'espace approprié pour chaque utilisateurs</li> </ol> |
| Scenario alternatif | <ol style="list-style-type: none"> <li>1. E-mail et mot de passe sont incorrects, un retour vers la page d'authentification sera effectué avec un message d'erreur</li> </ol>  |

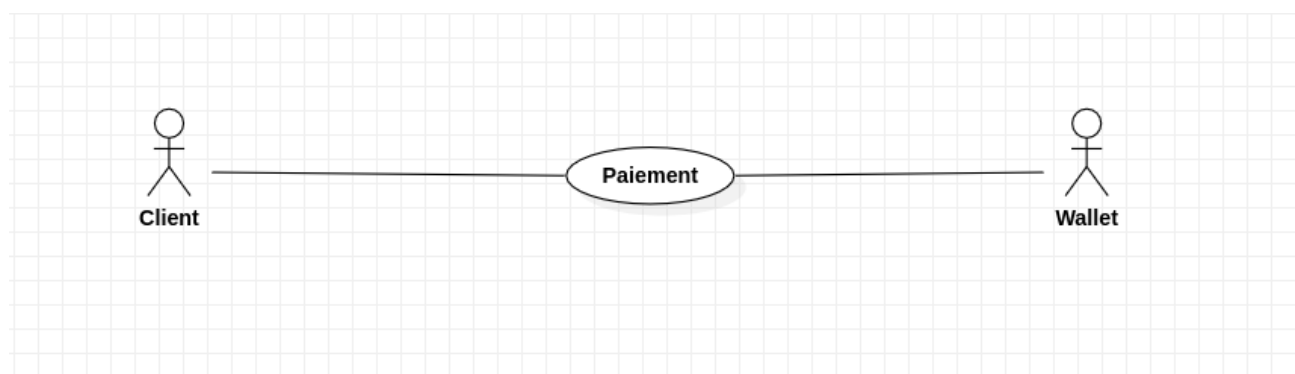
### 2.3.4 Cas d'utilisation Authentification Web3



21. Figure: Diagramme de cas d'utilisation authentification Web3

|                            |   |
|----------------------------|---|
| <b>Acteur principal</b>    | ➤ Client  |
| <b>Objectif</b>            | Confirmer l'authentification  |
| <b>Scenario nominal</b>    | <ol style="list-style-type: none"> <li>1. Wallet demande la confirmation de l'opération.</li> <li>2. L'utilisateur confirme l'authentification</li> <li>3. Le système affiche le modal de paiement</li> </ol> |
| <b>Scenario alternatif</b> | <ol style="list-style-type: none"> <li>1. L'utilisateur refuse l'authentification</li> </ol>  |

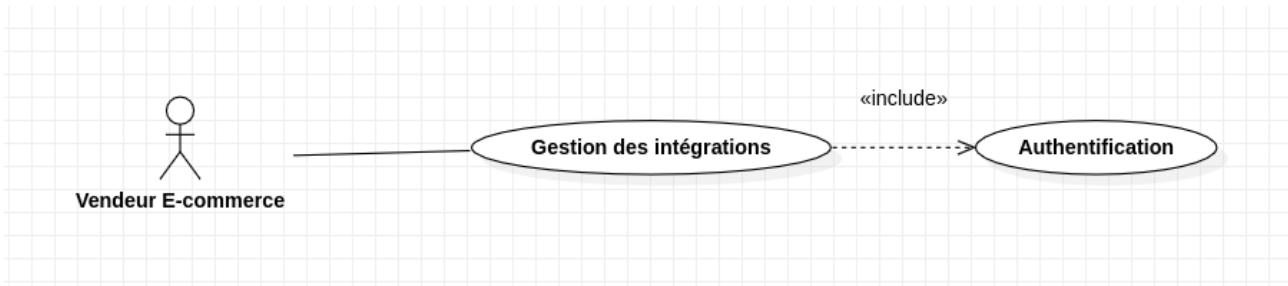
### 2.3.5 Cas d'utilisation Paiement



22. Figure: Diagramme de cas d'utilisation Paiement

|                            |  |
|----------------------------|--|
| <b>Acteur principal</b>    | ➤ Client   |
| <b>Objectif</b>            | Effectuer des paiements  |
| <b>Scenario nominal</b>    | <ol style="list-style-type: none"> <li>1. L'utilisateur choisit à faire le paiement</li> <li>2. Le système affiche le modal de paiement</li> <li>3. L'utilisateur choisit et valide l'opération</li> <li>4. le système effectuer la transaction</li> <li>5. Le système actualise sa BDD et affiche le message de succès .</li> </ol> |
| <b>Scenario alternatif</b> | <ol style="list-style-type: none"> <li>1. L'utilisateur annuler l'opération</li> <li>2. L'utilisateur n'a pas le solde suffisant</li> </ol>  |

## 2.3.6 Cas d'utilisation Gestion des intégrations



23. Figure: Diagramme de cas d'utilisation Gestion des intégrations

|                  |   |
|------------------|---|
| Acteur principal | ➤ Vendeur E-commerce  |
| Objectif         | Pouvoir ajouter, modifier, supprimer une intégration  |
| Scenario nominal | <p>➤ <b>Cas 1 : Ajouter une intégration</b></p> <ol style="list-style-type: none"> <li>1. Vendeur E-commerce choisit d'ajouter une intégration.</li> <li>2. Le système affiche le formulaire à remplir.</li> <li>3. Vendeur E-commerce remplit et valide le formulaire.</li> <li>4. Le système ajoute les informations dans la BDD.</li> <li>5. Le système actualise la liste des intégrations et il l'affiche.</li> </ol> <p>➤ <b>Cas 2 : Modifier une intégration</b></p> <ol style="list-style-type: none"> <li>1. Vendeur E-commerce choisit une intégration à modifier.</li> <li>2. Le système affiche le formulaire de modification.</li> <li>3. Il modifie les champs voulus.</li> <li>4. Le système met à jour les informations dans la BDD.</li> <li>5. Le système actualise la liste des intégrations et il l'affiche.</li> </ol> <p>➤ <b>Cas 3 : Supprimer une intégration</b></p> <ol style="list-style-type: none"> <li>1. Vendeur E-commerce choisit une intégration à supprimer.</li> <li>2. Le système demande une confirmation.</li> <li>1. Vendeur E-commerce confirme ou annule la suppression.</li> </ol> |

|                     |   |
|---------------------|---|
|                     | <ol style="list-style-type: none"> <li>3. Le système supprime l'opération de la BDD.</li> <li>4. Le système actualise la liste des intégrations et il l'affiche.</li> </ol> |
| Scenario alternatif | <ol style="list-style-type: none"> <li>1. Modification avec des champs vides, champs non conforme aux types : un message d'erreur sera affiché.</li> </ol>                  |

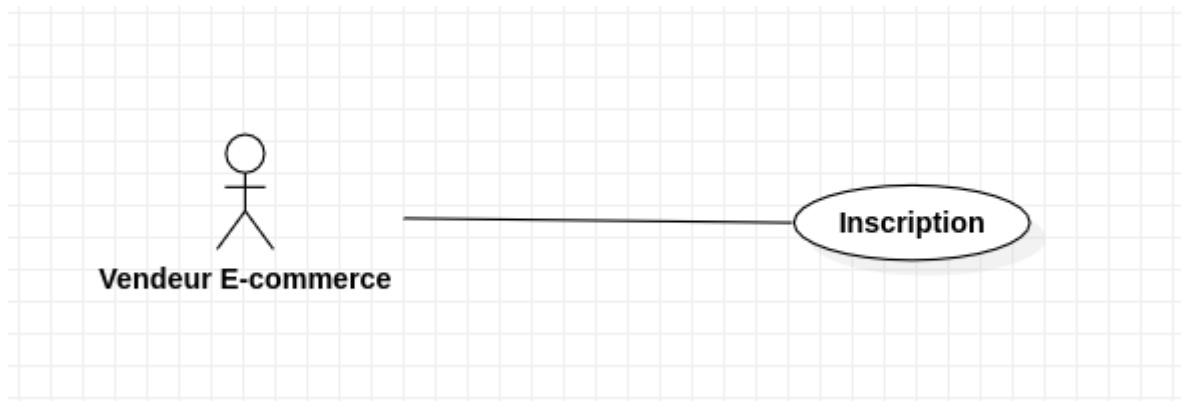
## 2.3.7 Cas d'utilisation Réalisation des transactions



24. Figure: Diagramme de cas d'utilisation Réalisation des transactions

|                     |   |
|---------------------|---|
| Acteur principal    | ➤ Vendeur E-commerce  |
| Objectif            | Pouvoir effectuer des transaction   |
| Scenario nominal    | <ol style="list-style-type: none"> <li>1. Vendeur E-commerce choisit d'envoyer des crypto-monnaie</li> <li>2. Le système affiche le formulaire à remplir.</li> <li>3. Vendeur E-commerce remplit et valide le formulaire.</li> <li>4. la portefeuille s'ouvre pour confirmer <b>Gaz Fees</b></li> </ol> |
| Scenario alternatif | <ol style="list-style-type: none"> <li>1. les champs non conforme aux types, formulaire vide : un message d'erreur sera affiché.</li> </ol>   |

## 2.3.8 Cas d'utilisation Inscription



25. Figure: Diagramme de cas d'utilisation Inscription

|                            |   |
|----------------------------|---|
| <b>Acteur principal</b>    | ➤ Vendeur E-commerce  |
| <b>Objectif</b>            | Demande d'inscription   |
| <b>Scénario nominal</b>    | <ol style="list-style-type: none"> <li>1. L'utilisateur choisit à faire l'inscription .</li> <li>2. Le système affiche le formulaire à remplir .</li> <li>3. L'utilisateur remplit et valide le formulaire.</li> <li>4. Le système ajoute l'utilisateur à la liste des demandes .</li> <li>5. Le système actualise sa BDD et affiche la page d'attente .</li> </ol> |
| <b>Scénario alternatif</b> | <ol style="list-style-type: none"> <li>1. les champs non conforme aux types, formulaire vide : un message d'erreur sera affiché.</li> </ol>   |

### 3 Diagramme de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language (UML). On représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets .

Pour réaliser les diagrammes des séquences nous avons utilisé des opérateurs d'interactions. Un opérateur d'interaction définit le type d'un fragment composé.

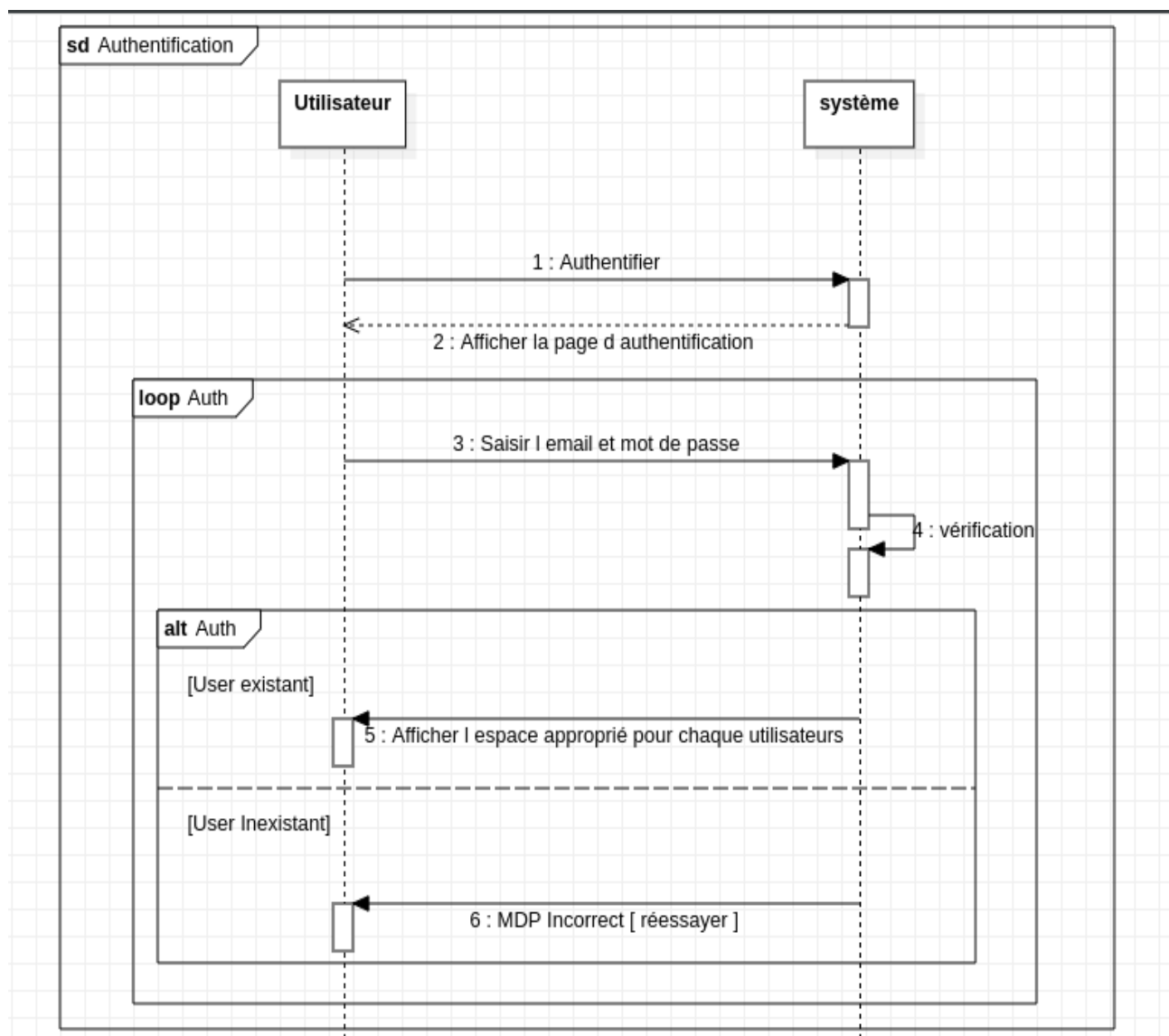
Les opérateurs d'interaction que nous avons utilisés dans les diagrammes de séquences sont :

1. **Référence (ref)** : cet opérateur désigne que le fragment fait référence à un cas vu précédemment.

2. **Alternative(Alt)** : cet opérateur désigne que le fragment composé représente un choix de comportement. Une opérande d'interaction au maximum sera choisie. L'opérande choisi doit avoir une expression de garde implicite ou explicite qui a la valeur "true" à ce point de l'interaction.
3. **Boucle (Loop)** : cet opérateur désigne que le fragment composé représente une boucle. L'opérande "Loop" sera répétée plusieurs fois .

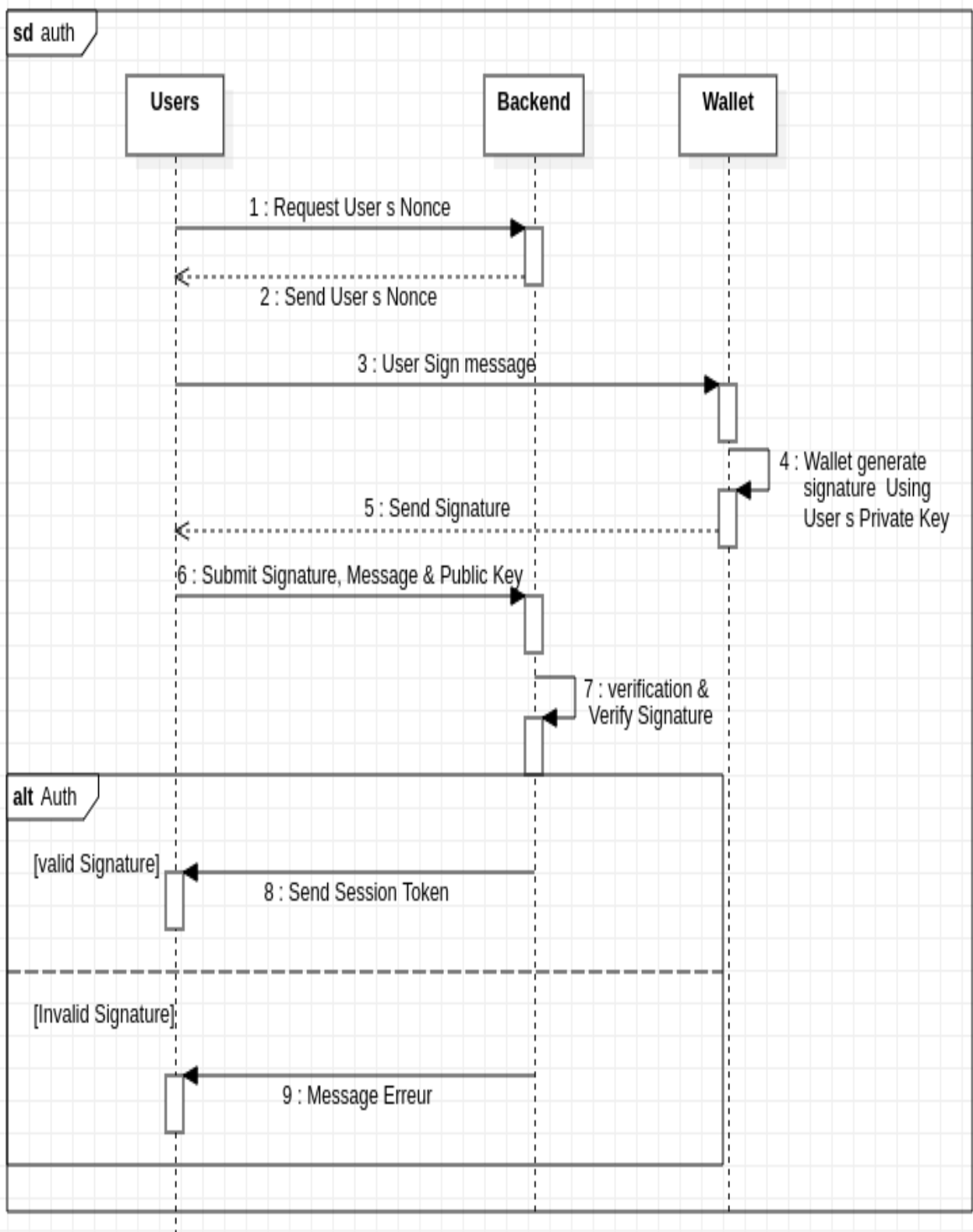
Les figures depuis **la figure 21** jusqu'à **la figure 24** présenterons respectivement les diagrammes des séquences : Authentification, Paiement, Réalisation des transactions .

### 3.1 Diagramme de séquence d'authentification :



26. Figure: Diagramme séquence système lié au cas d'utilisation Authentification

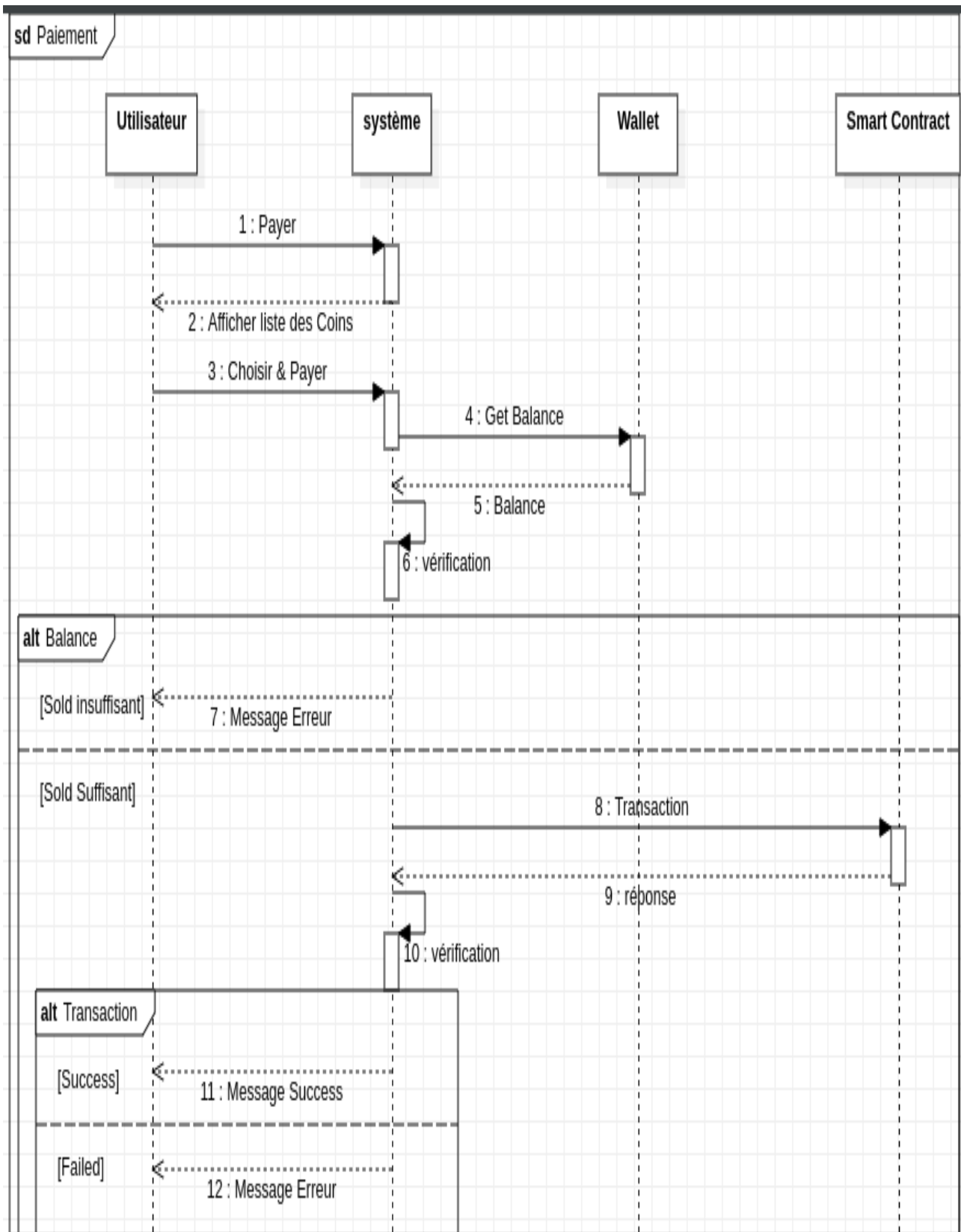
### 3.2 Diagramme de séquence d'authentification Web3:



27. Figure: Diagramme séquence système lié au cas d'utilisation Authentification Web3

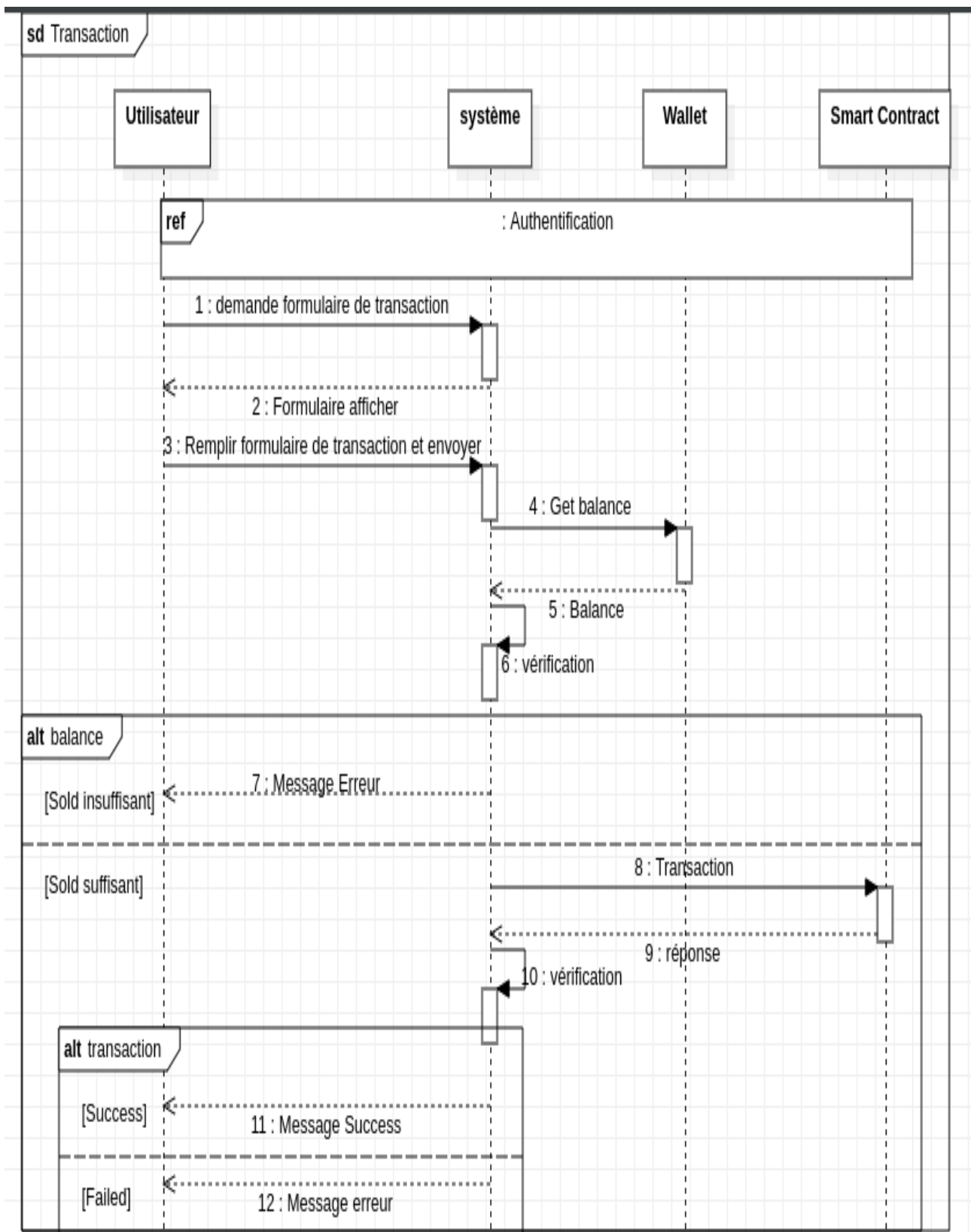


### 3.3 Diagramme de séquence Paiement :



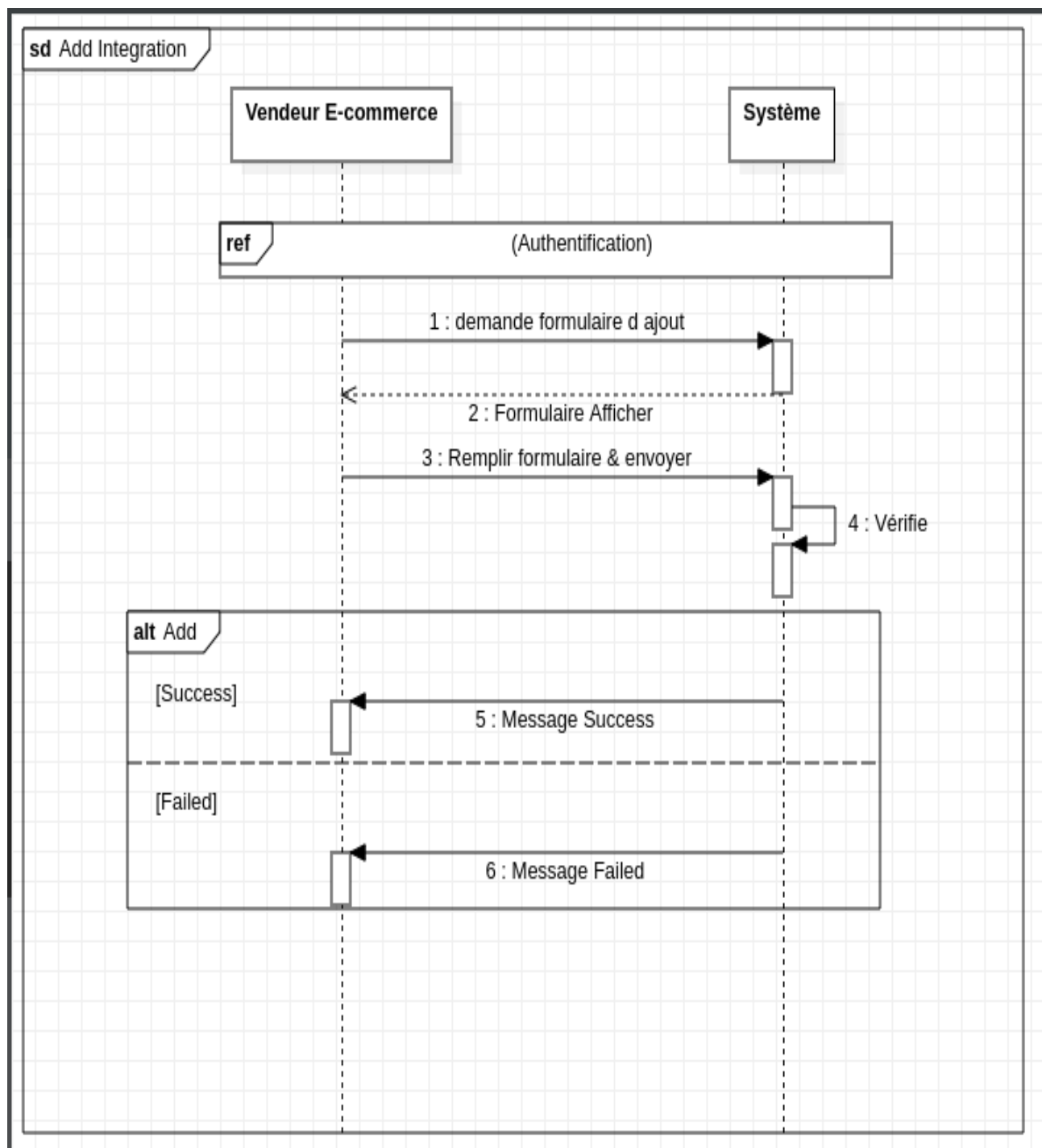
28. Figure: Diagramme séquence système lié au cas d'utilisation Paiement

### 3.4 Diagramme de séquence Réalisation des transactions



29. Figure: Diagramme séquence système lié au cas d'utilisation Transaction

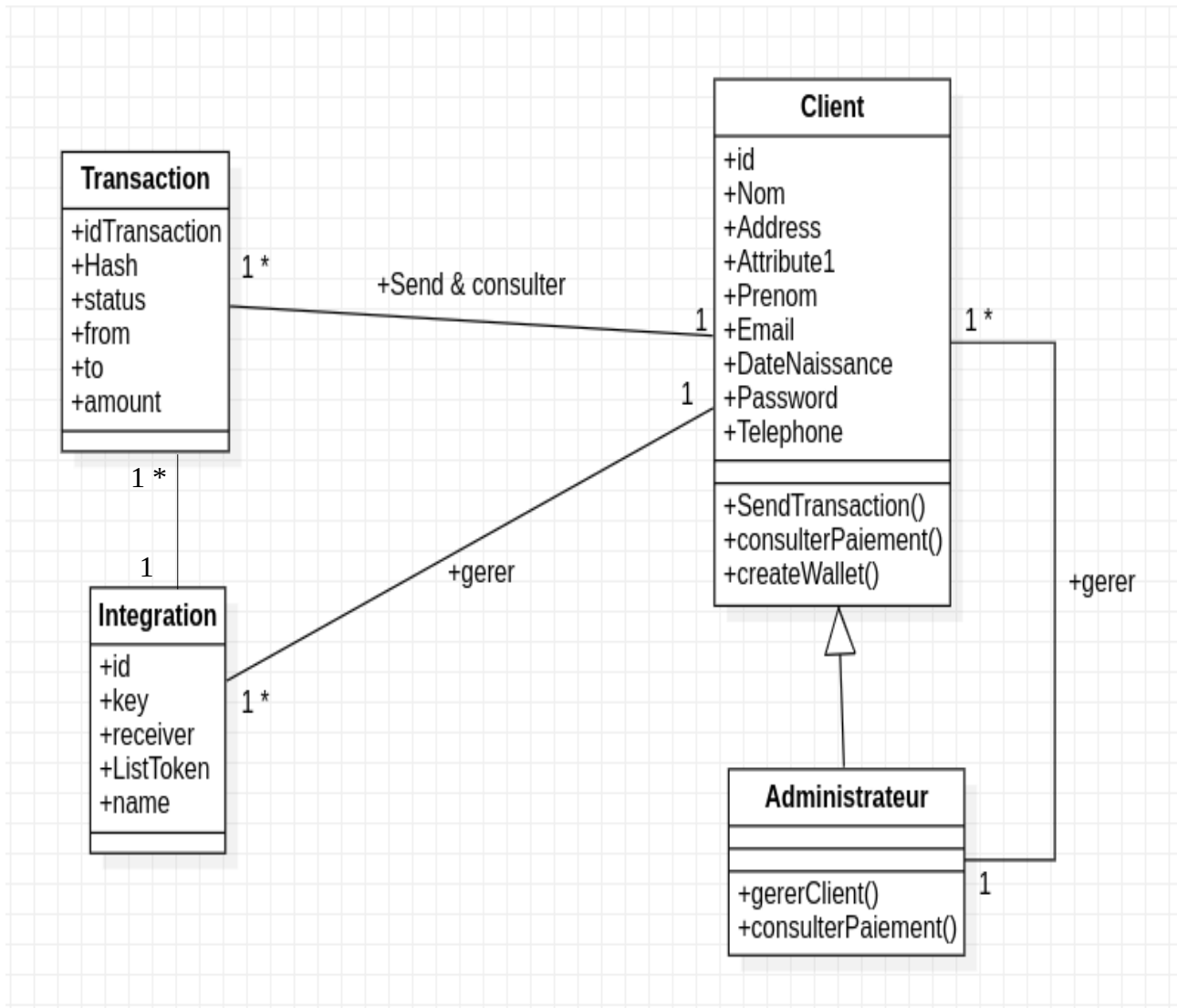
### 3.4 Diagramme de séquence d'ajout une Intégration



30. Figure: Diagramme de séquence d'ajout une Intégration

## 3 Diagramme de classes

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.



31. Figure: Diagramme des Classes

## Conclusion

Dans ce chapitre, nous avons décrit la phase d'analyse et conception de notre projet. Et nous allons présenter et définir quelques diagrammes du formalisme UML, relatifs à notre projet afin d'illustrer son fonctionnement. Le chapitre suivant est dédié à la phase de réalisation de notre projet.

## Chapitre 4 :Réalisation

Dans ce dernier chapitre comporte en premier lieu les outils, les technologies utilisés pour répondre au besoin du projet, en deuxième lieu la mise en œuvre de l'application web.

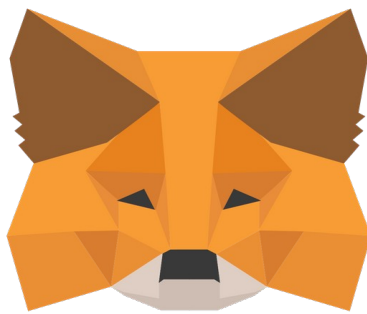
# 1. développement de la bibliothèque JavaScript

## 1.1 Choix des outils

### ➤ Metamask

Metamask est un wallet (portefeuille) de cryptodevises **basé sur la machine virtuelle Ethereum (Ethereum Virtual Machine)**. Il est donc compatible avec n'importe quelle Blockchain s'appuyant sur EVM à l'instar du BAT (Basic Attention Token) et du BNB (Binance Coin).

Metamask est également connu comme étant une extension de navigateur permettant d'interagir avec des Dapp (applications décentralisées). Il est compatible avec Google Chrome, Firefox et Brave, sachant que les risques d'hameçonnage sont réduits grâce à un anti-phishing intégré.



### ➤ WalletConnect

WalletConnect est un protocole ouvert conçu pour faciliter une connexion sécurisée entre les portefeuilles mobiles de crypto-monnaie et les dapps.

Les transactions sont effectuées via une connexion cryptée en scannant un code QR et sont confirmées sur l'appareil mobile. Comme la clé privée ne quitte jamais l'appareil de l'utilisateur, ses fonds ne sont jamais menacés et la possibilité d'un détournement est très faible.



### ➤ Etherscan

Etherscan est connu comme le principal explorateur de blocs Ethereum. Il s'agit essentiellement d'un moteur de recherche permettant aux utilisateurs de rechercher, de

confirmer et de valider des transactions sur la plateforme décentralisée de contrats intelligents Ethereum. En entrant une adresse dans la boîte de recherche, vous pouvez visualiser le solde, la valeur et toutes les ventes ou achats passés via cette adresse.



### ➤ **JSON-RPC**

JSON-RPC est un protocole allégé de procédure à distance (RPC). En premier lieu, la spécification définit plusieurs structures de données et les règles autour de leur traitement. C'est un système de transport agnostique en ce sens que les concepts peuvent être utilisés dans le même processus, via les sockets et HTTP, ou dans de nombreux environnements de passage de messages. Il utilise JSON (RFC 4627) comme format de données.

### ➤ **Rollup**

Rollup est un bundler JavaScript. C'est à dire qu'il lit votre code et bundle l'ensemble des modules importés (via import ou require) en un fichier unique. Il est capable d'exporter ce module dans les principaux formats de modules (CJS, ESM, AMD, IIFE) correspondant à tous les cas d'usage. En outre, via quelques plugins bien utiles, il se chargera aussi de transpiler et de minifier le code.



### ➤ **Babel**

Babel est un Transpileur, un type de compilateur qui sert compiler du code source d'un certain langage de programmation en du code source d'un autre langage de programmation. En ce qui concerne Babel, il permet de convertir du Javascript en du Javascript. Pour être plus précis, il permet de convertir du code javascript récent en du code javascript capable d'être interprété par des vieux navigateurs.



### ➤ **Ether.Js**

La bibliothèque ethers.js vise à être une bibliothèque complète et compacte pour interagir avec la Blockchain Ethereum et son écosystème. Il a été conçu à l'origine pour être utilisé avec ethers.io et s'est depuis développé en une bibliothèque plus générale.



### ➤ **HTML**

“HyperText Markup Language” est un langage dit de “marquage” (de “structuration” ou de “balisage”) dont le rôle est de formaliser l'écriture d'un document avec des balises de formatage. Les balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents. La version 5.0 du langage HTML définit deux syntaxes de DOM : HTML5 et XHTML5. Cette version apporte de nouvelles possibilités en terme de création d'“applications Web riches” bénéficiant de l'intégration d'éléments multimédias et d'interactivité, à l'image de ce que permettent Adobe Flash ou Microsoft Silverlight.





## ➤ CSS

Cascading Style Sheets (feuilles de styles en cascade), servent à mettre en forme des documents web, type page HTML ou XML. Par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus-dessous, etc.), le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web. Les feuilles de styles ont d'ailleurs pour objectif principal de dissocier le contenu de la page de son apparence visuelle.



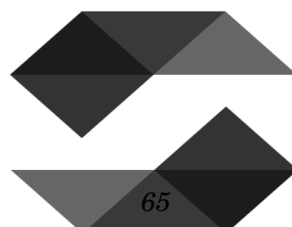
## ➤ Javascript

Javascript est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web.



- **Solidity** est un langage orienté objet et de haut niveau pour la mise en œuvre de contrats intelligents. Les contrats intelligents sont des programmes qui régissent le comportement des comptes dans l'état Ethereum.

Solidity est un langage d'accolades. Il est influencé par le C++, le Python et le JavaScript, et est conçu pour cibler la machine virtuelle Ethereum (EVM).



- **Uniswap** Écrit dans le langage de contrat intelligent Vyper, Uniswap est un protocole de liquidité automatisé et open-source sur Ethereum qui permet de trader et de lister facilement des jetons ERC20.

Construit autour des valeurs de décentralisation, de résistance à la censure, de sécurité et de fonctionnement sans autorisation, Uniswap est devenu l'échange automated market maker (AMM) le plus populaire d'Ethereum depuis le lancement de la version 1 d'Uniswap en novembre 2018.



### 3. développement de l'application Web

#### 2.1 Backend

- **PHP** (Hypertext Preprocessor) est langage de scripts généraliste et Open Source spécialement conçu pour créer des pages Web dynamiques. C'est un langage qui est facile à apprendre, qui permet de communiquer avec les bases de données et qui est multi plateforme.



- **Laravel** : l'un des frameworks web PHP les plus populaires. Il est gratuit, et son code source est ouvert et basé sur le modèle architectural MVC ( Model View Controller ) et entièrement développé en programmation orientée objet .



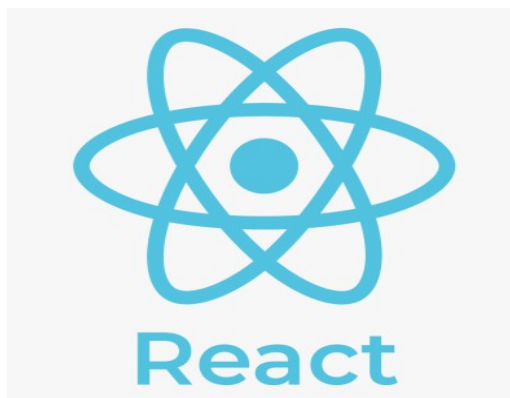
- **MongoDB** est une base de données de documents, ce qui signifie qu'elle stocke les données dans des documents de type JSON. C'est la façon la plus naturelle de penser aux données, et qu'elle est beaucoup plus expressive et puissante que le modèle traditionnel de ligne/colonne.

Le modèle de document de MongoDB est simple à apprendre et à utiliser pour les développeurs, tout en offrant toutes les fonctionnalités nécessaires pour répondre aux exigences les plus complexes à n'importe quelle échelle. Il fournit des pilotes pour plus de 10 langues, et la communauté en a construit des dizaines d'autres.



## 2.2 Frontend

- **React.Js** React est une bibliothèque JavaScript open-source qui est utilisée pour construire des interfaces utilisateur spécifiquement pour des applications d'une seule page. Elle est utilisée pour gérer la couche d'affichage des applications web et mobiles. React nous permet également de créer des composants d'interface utilisateur réutilisables. React a été créé par Jordan Walke, un ingénieur logiciel travaillant pour Facebook. React a été déployé pour la première fois sur le flux d'informations de Facebook en 2011 et sur Instagram.com en 2012.



- **Material-ui** un ensemble de règles de design proposées par Google. Il est parfois traduit par « conception matérielle » ou par « design contextuel ». C'était d'abord un design destiné aux applications mobiles (mobile first). Il est ensuite devenu l'une des grandes tendances du design d'interface. L'approche de Google rappelle le Flat Design (ou design plat).

Material-ui donne Des composants React pour un développement web plus rapide et plus simple.



- **Tailwind CSS** est un framework utility-first CSS (feuilles de style en cascade) avec des classes prédéfinies que vous pouvez utiliser pour construire et concevoir des pages web directement dans votre balisage. Il vous permet d'écrire du CSS dans votre HTML sous la forme de classes prédéfinies.



## 3. Les interfaces

### 3.1 Bibliothèque JavaScript

#### 3.1.1 Exemple de l'intégration

Un exemple d'intégration de la librairie dans un site de commerce électronique

## Checkout

Home > Electronics > Headphones > Cart > Checkout

Logitech K251  
\$20.00

Pay With Crypto

or pay with card

Email

Card details

0000 1234 6549 15151

MM/YY CVC

Name on card

Name on card

32. Figure: Checkout Page

### 3.1.2 Authentication Web3

Chaque utilisateur doit s'authentifier avec son Wallet

## Checkout

Home > Electronics > Headphones > Cart > Checkout

Logitech K251  
\$20.00

Connect wallet

Connect with one of our available wallet providers or create a new one.

MetaMask Popular

WalletConnect

Why do I need to connect with my wallet?

Connect Wallet

or pay with card

Email

Card details

0000 1234 6549 15151

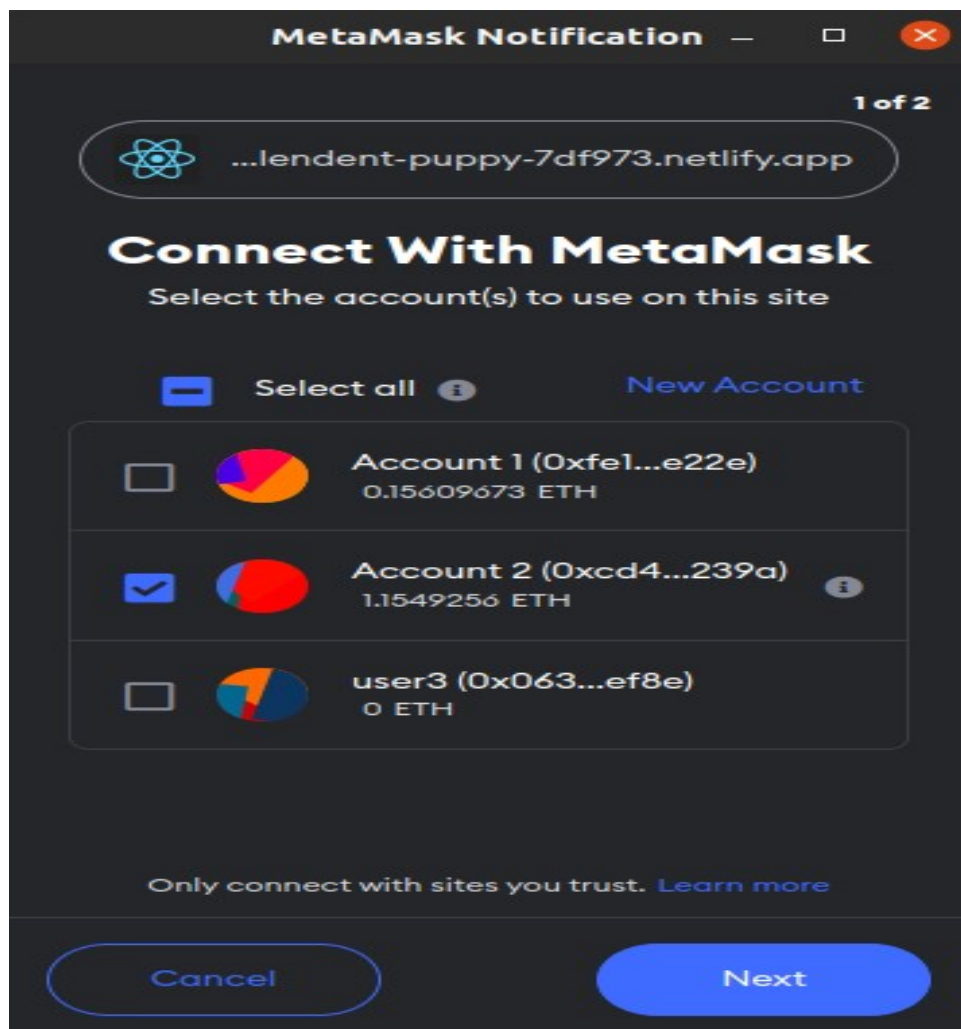
MM/YY CVC

Name on card

Name on card

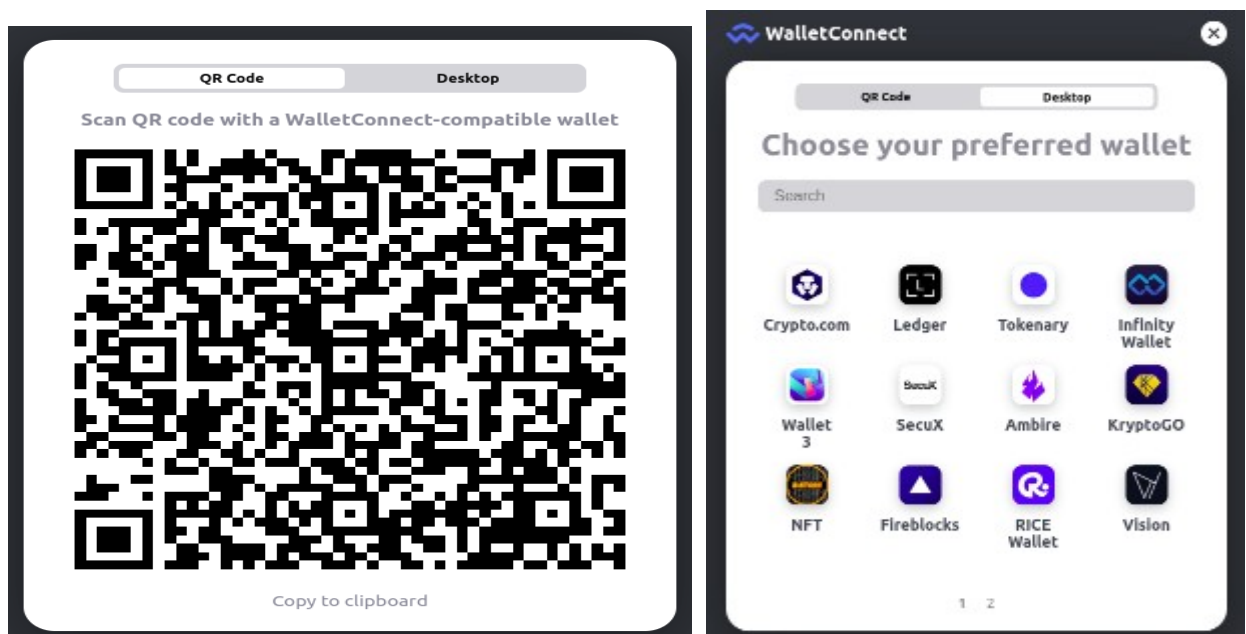
33. Figure: d'authentification web3

#### ➤ Metamask



34. Figure: Metamask

## ➤ WalletConnect

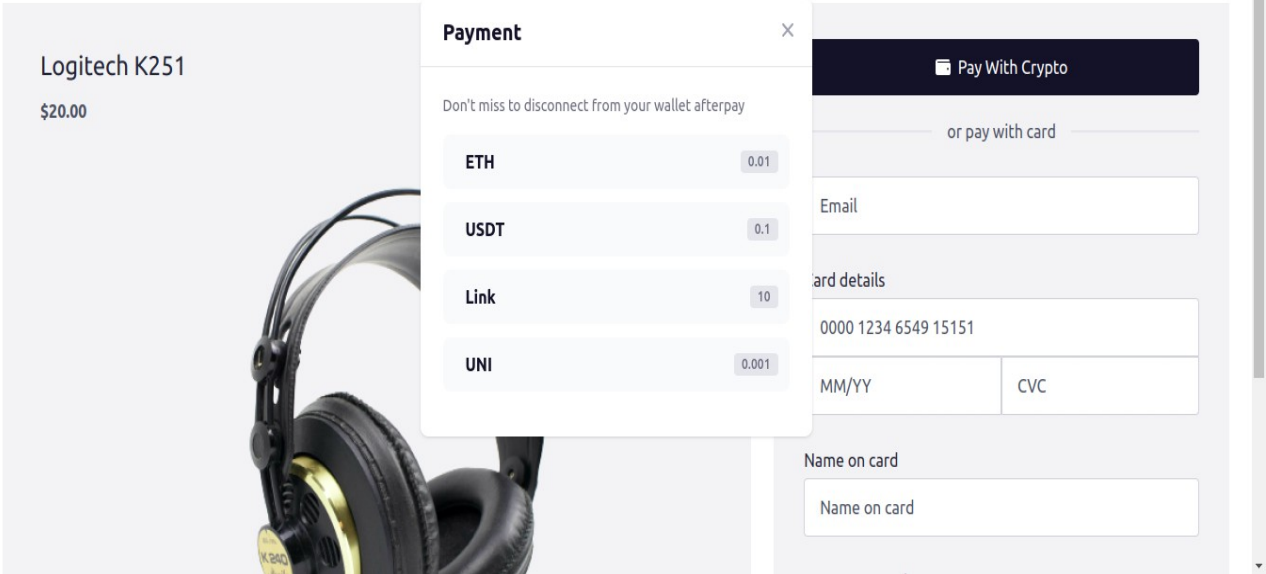


35. Figure: WalletConnect

### 3.1.2 Paiement

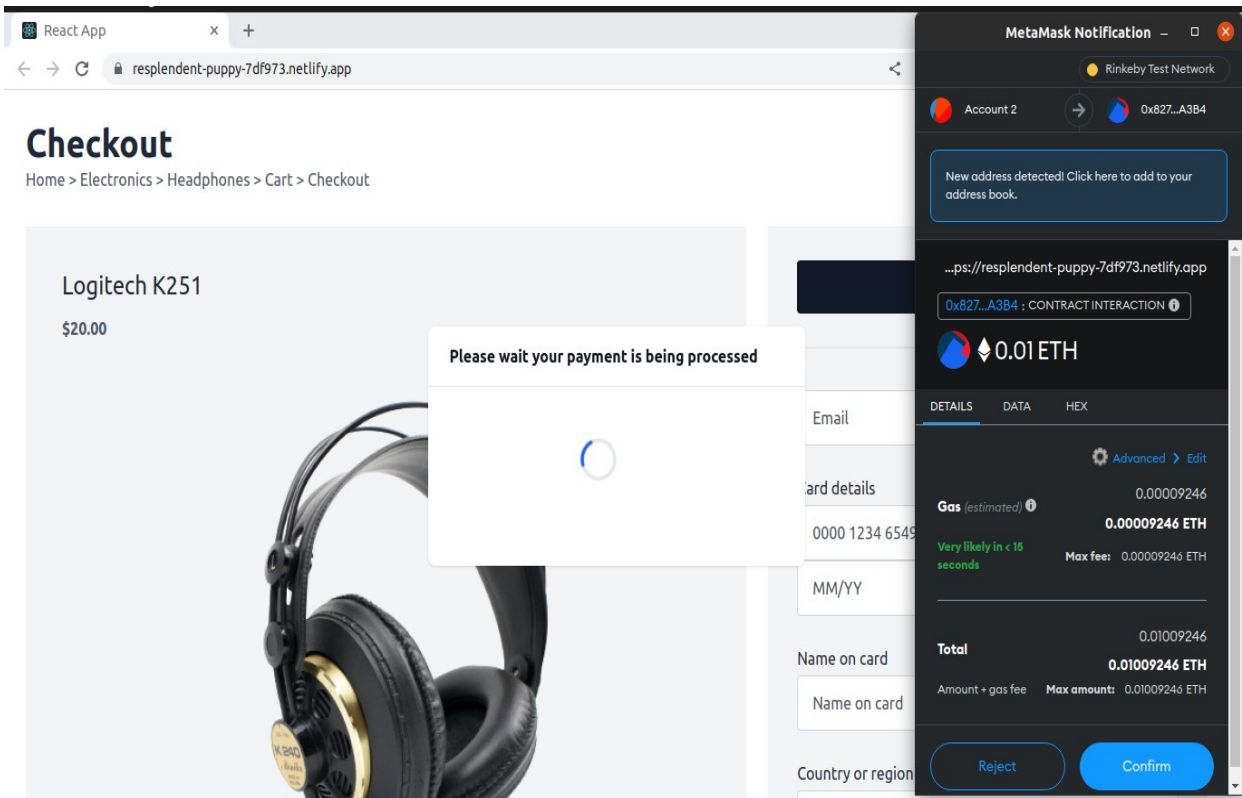
#### Checkout

Home > Electronics > Headphones > Cart > Checkout



36. Figure: Paiement

#### ➤ ETH



37. Figure: Paiement avec ETH

## les détails de la transaction dans Etherscan

Overview

Internal Txns

State

[ This is a Rinkeby Testnet transaction only ]

Transaction Hash:

0x290ca829bc30fe37e034a2b5edf4994728610cdd004fb71046305ca150f5386a

Status:

Success

Block:

11322629

15 Block Confirmations

Timestamp:

3 mins ago (Sep-04-2022 04:26:23 PM +UTC)

From:

0xcd40213965257586d3bbe616990c92db9259239a

To:

Contract 0x82703a9f3618dce7ce840f45704ed0160066a3b4

TRANSFER 0.01 Ether From 0x82703a9f3618dce7ce840f457... To 0xcd40213965257586d3bbe61...

Value:

0.01 Ether (\$0.00)

Transaction Fee:

0.000085701 Ether (\$0.00)

Gas Price:

0.000000003 Ether (3 Gwei)

Click to see More

38. Figure: Les détails de la transaction dans Etherscan ETH

## ➤ UNI

React App


resplendent-puppy-7df973.netlify.app

# Checkout

Home > Electronics > Headphones > Cart > Checkout

Logitech K251

\$20.00



Please wait your payment is being processed

Email

Card details

0000 1234 6549

MM/YY

Name on card

Name on card

Country or region

MetaMask Notification

Rinkeby Test Network

Account 2 → 0x827...A3B4

New address detected! Click here to add to your address book.

https://resplendent-puppy-7df973.netlify.app

0x827...A3B4 : CONTRACT INTERACTION

DETAILS DATA HEX

Gas (estimated) 0.00225 0.00225 ETH  
Very likely in < 15 seconds Max fee: 0.00225 ETH

Total 0.00225 0.00225 ETH  
Amount + gas fee Max amount: 0.00225 ETH

Reject Confirm

39. Figure: Paiement avec UNI



## Les détails de la transaction dans Etherscan

OverviewInternal TxnsLogs (9)State

[ This is a Rinkeby Testnet transaction only ]

Transaction Hash:

0x24e4f2989b9ce5a9e1fa9c9cf0810ba597cc332e1346cbc5513908f1c23d87c

Status:

Success

Block:

113225221 Block Confirmation

Timestamp:

25 secs ago (Sep-04-2022 03:50:26 PM +UTC)

From:

0xcd40213965257586d3bbe610990c92db9259239a

Interacted With (To):

Contract 0x82703a9f3618dce7ce84045704ed0160066a3b4

TRANSFER 0.000724838780897103 Ether From 0xc778417e0631411396c0109... To 0x7a250c9630b4cfc529739d3c...

TRANSFER 0.000724838780897103 Ether From 0x7a250c9630b4cfc529739d3c... To 0xcd40213965257586d3bbe61...

Tokens Transferred: 3

From 0xcd40213965257... To 0x82703a9f3618d... For 0.001 Uniswap (UNI)

From 0x82703a9f3618d... To 0x4e99615101c0b... For 0.001 Uniswap (UNI)

From 0x4e99615101c0b... To 0x7a250c9630b4c... For 0.000724838780897103 Wrapped Ethe... (WETH)

Value:

0 Ether (\$0.00)

Transaction Fee:

0.000521109 Ether (\$0.00)

Gas Price:

0.000000003 Ether (3 Gwei)

Gas Limit & Usage by Txn:

750,000 | 173,703 (23.16%)

Gas Fees:

Base: 0.000000008 Gwei | Max: 3 Gwei | Max Priority: 3 Gwei

Burnt & Txn Savings Fees:

Burnt: 0.000000000001389624 Ether (\$0.00)

Txn Savings: 0 Ether (\$0.00)

Others:

Txn Type: 2 (GIP-1559)Nonce: 568Position: 2

Input Data:

| # | Name        | Type    | Data                                       |
|---|-------------|---------|--|
| 0 | tokenAmount | uint256 | 1000000000000000                           |
| 1 | token       | address | 0x1f9840a85d5aF5b101762F9258040dC4201F984  |
| 2 | to          | address | 0xcd40213965257586d3bbe610990c92db9259239a |

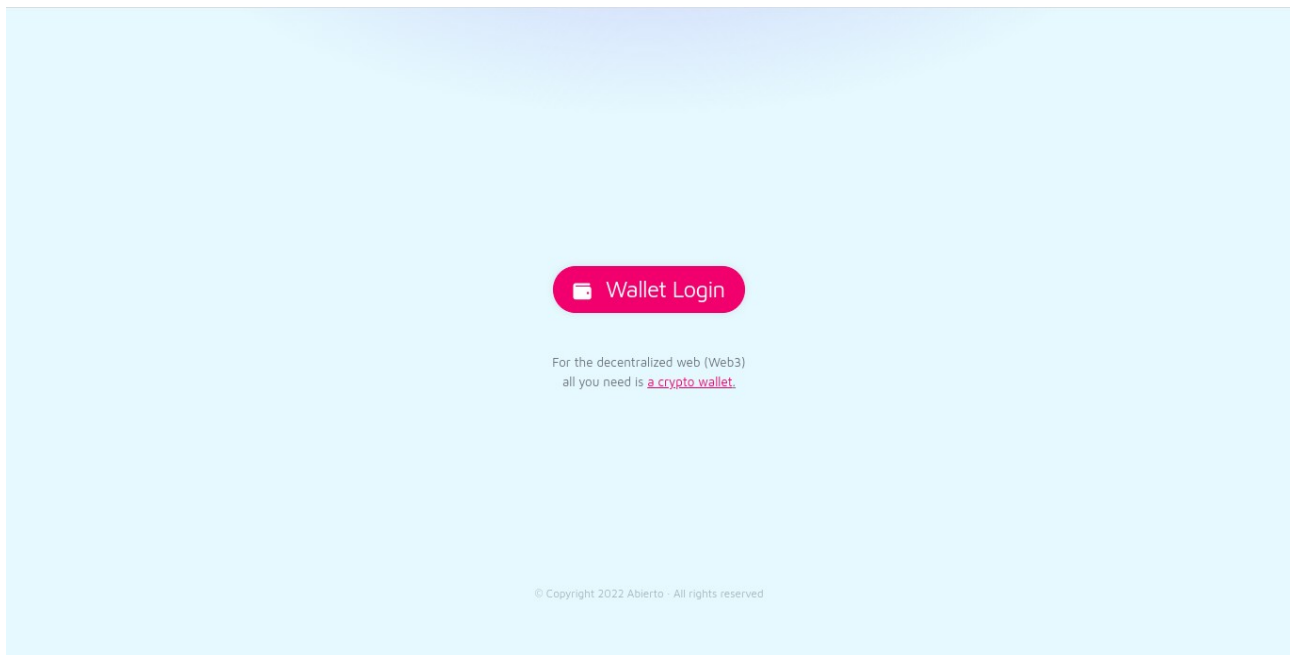
Switch Back

Click to see Less

40. Figure: Les détails de la transaction dans Etherscan UNI

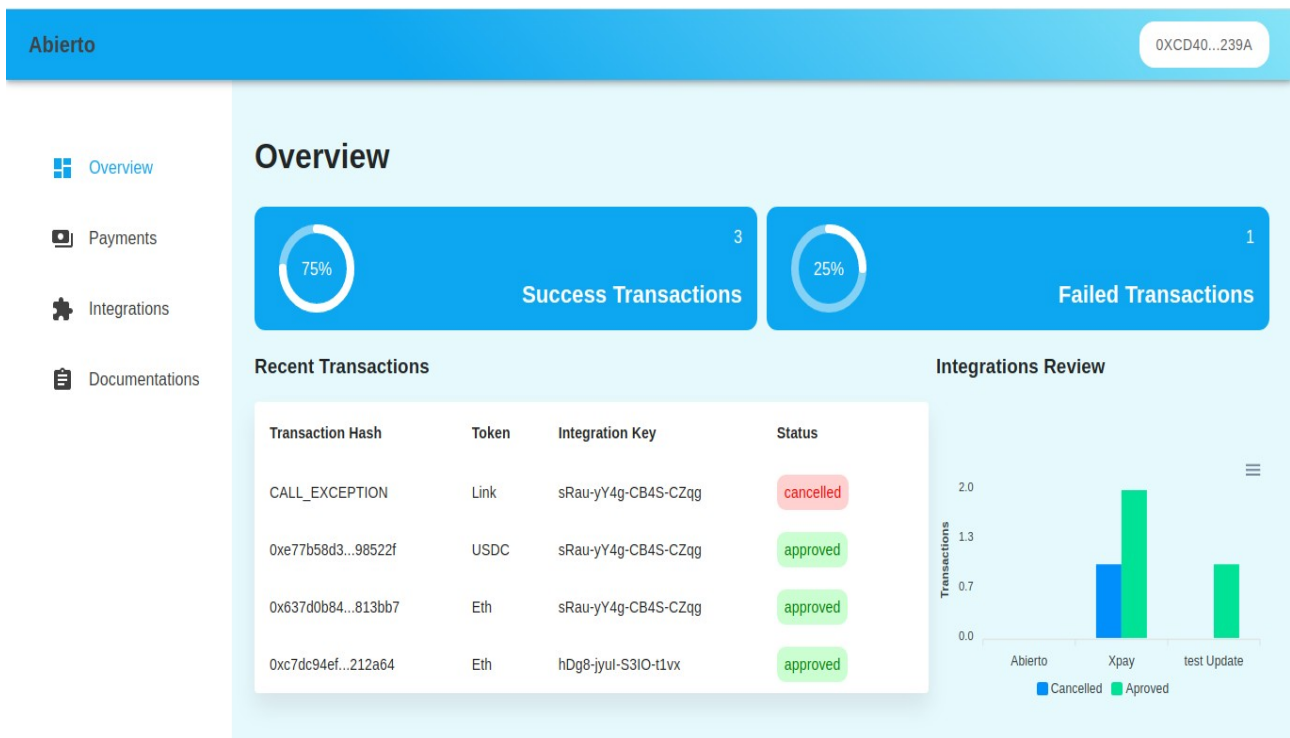
## 3.2 Application Web

### 3.2.1 Login



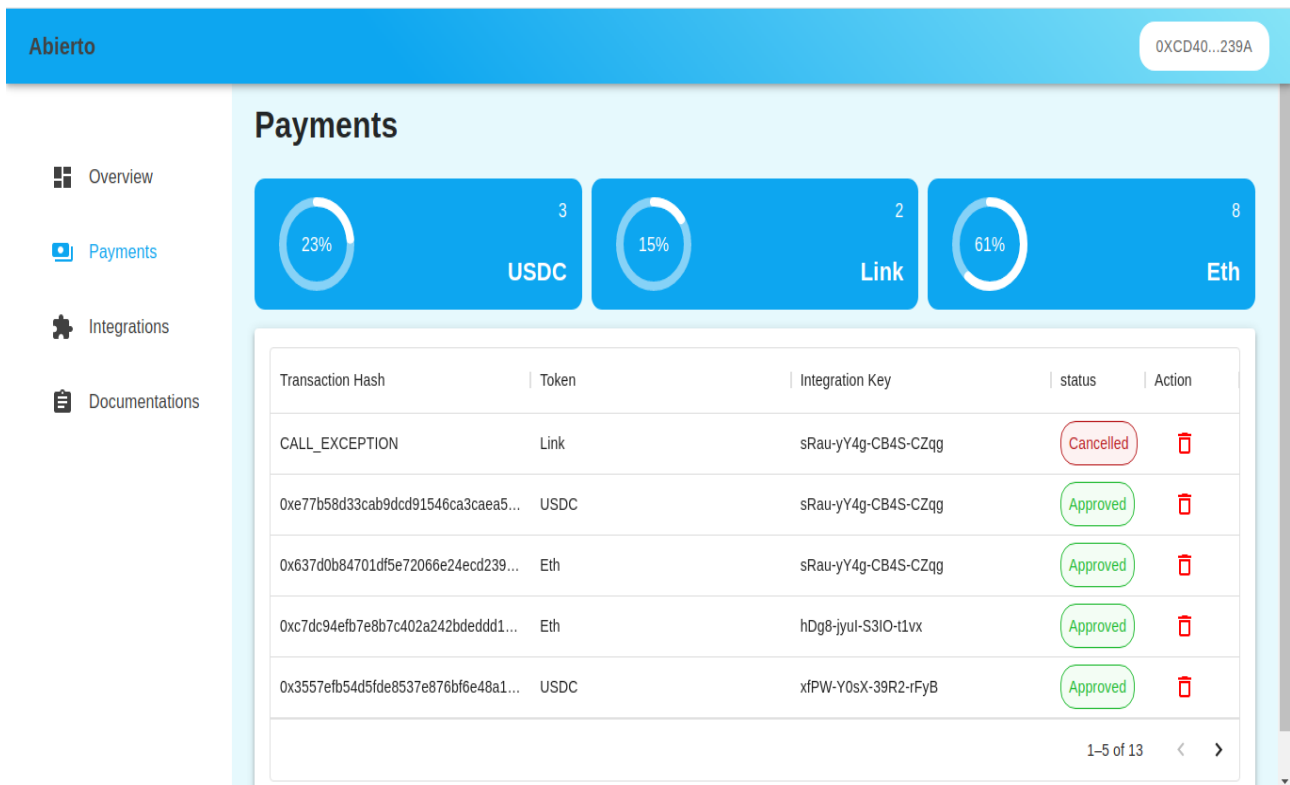
41. Figure: Login

### 3.2.2 Overview



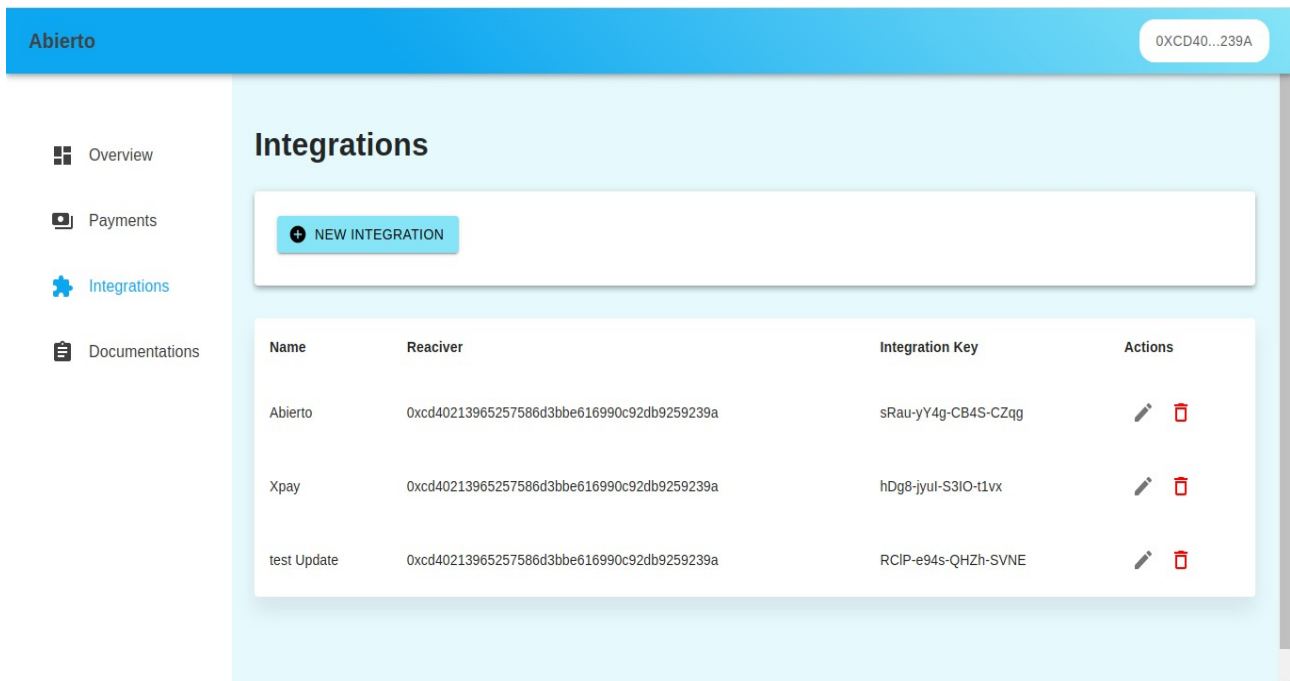
42. Figure: OverView

### 3.2.3 Paiements



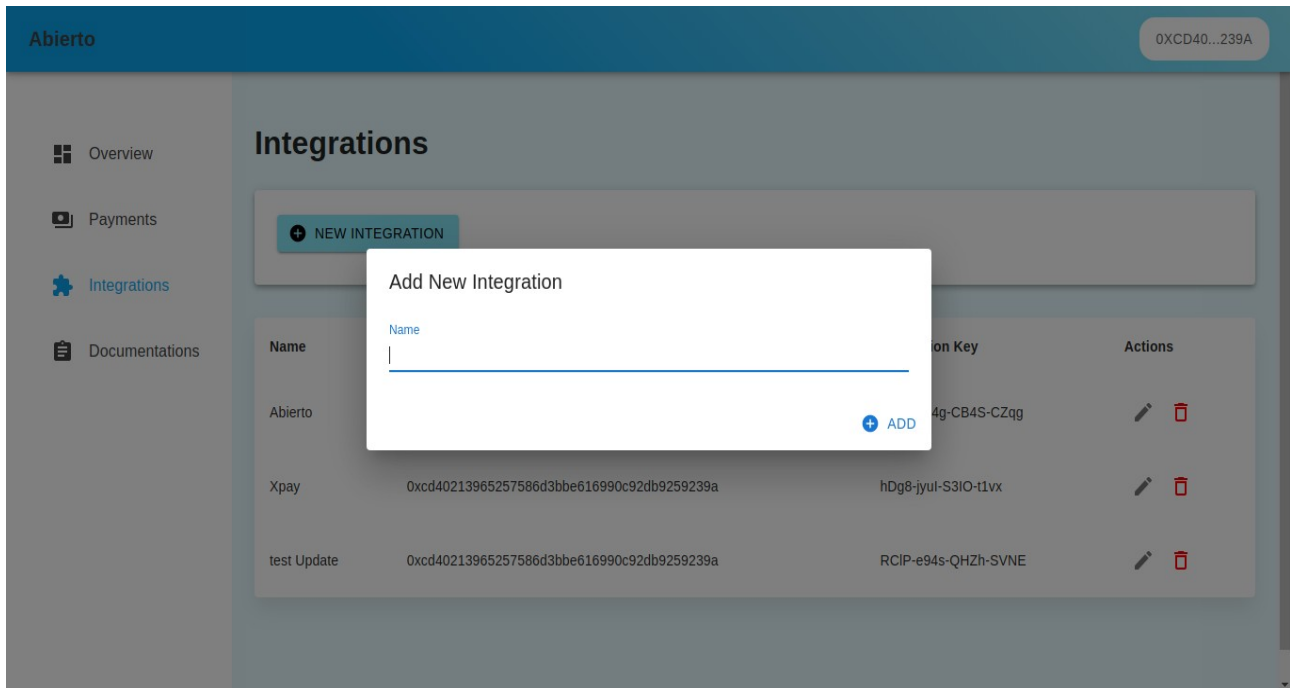
43. Figure: Paiement

### 3.2.4 Intégration



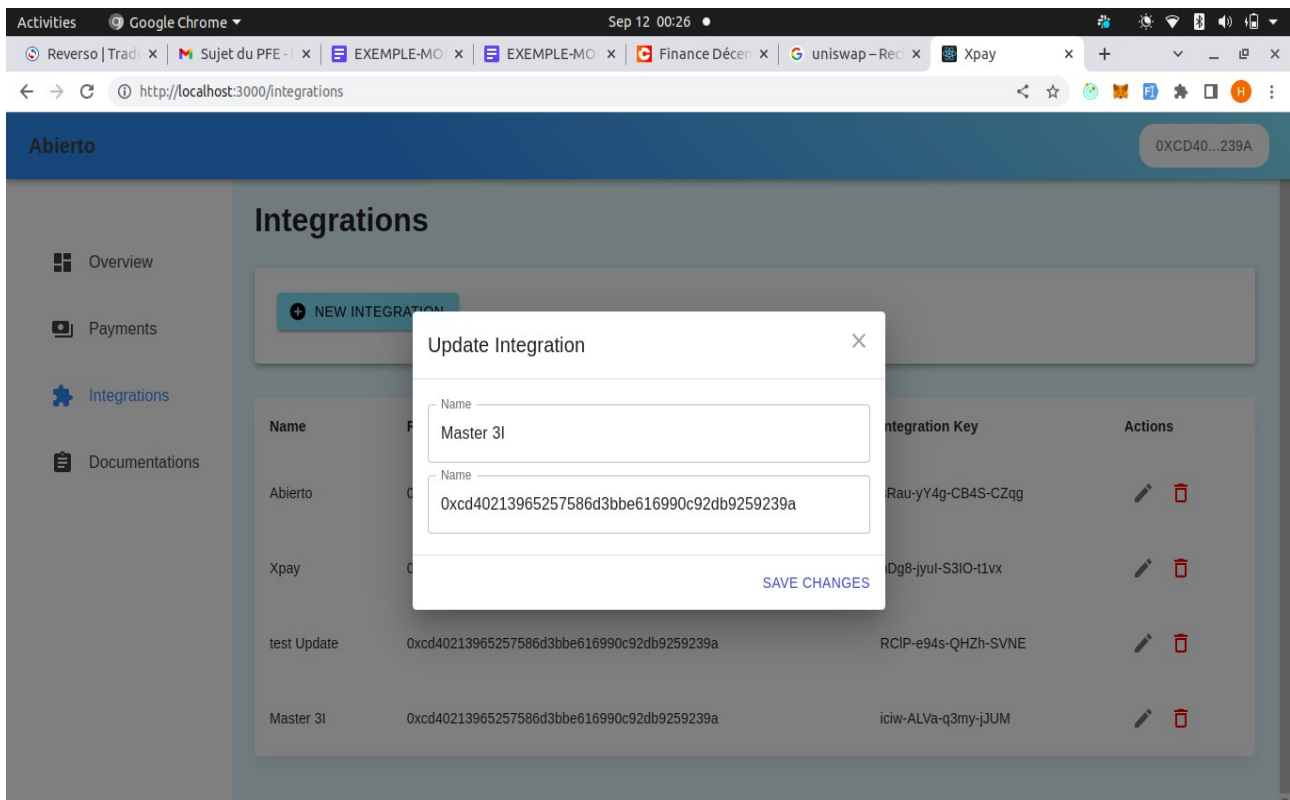
44. Figure: Integration

### 3.2.5 Ajouter une Intégration



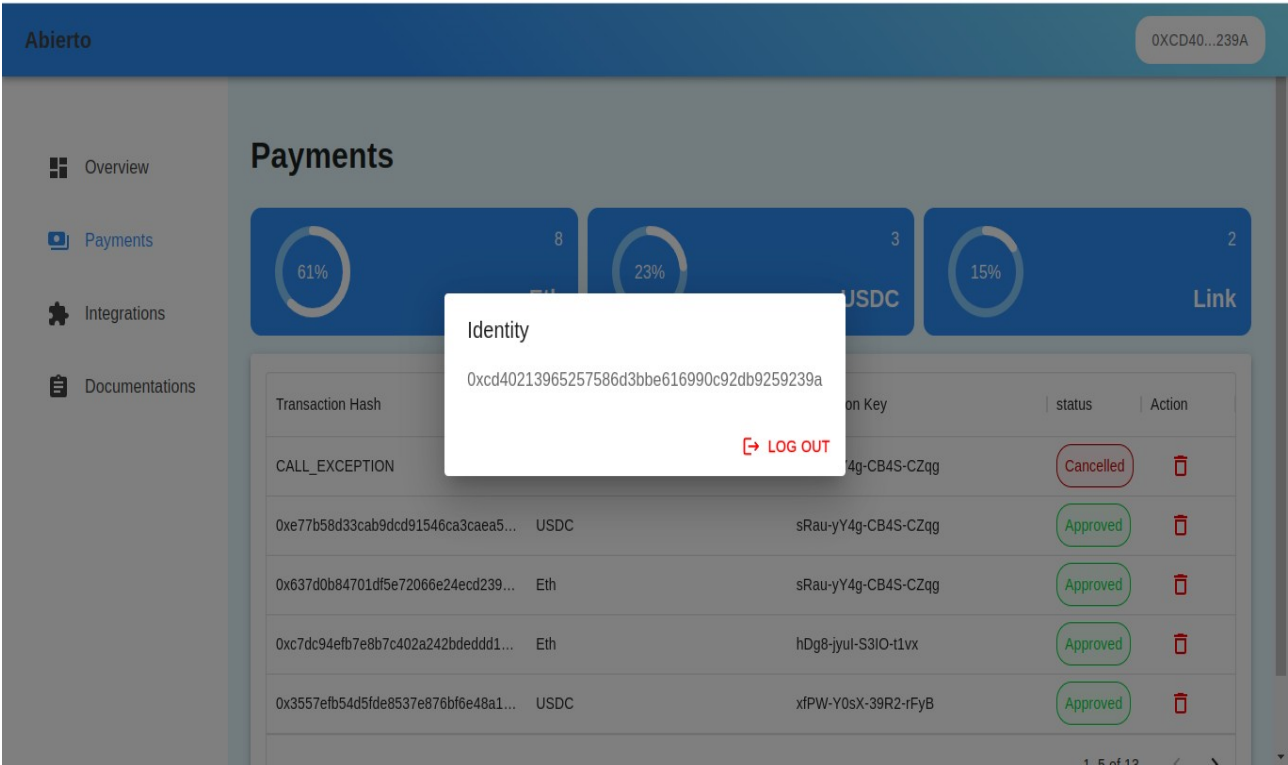
45. Figure: Add Integration

### 3.2.6 Modifier une Intégration



46. Figure: Update Integration

### 3.2.8 LogOut



47. Figure: LogOut

## Conclusion

J'ai présenté dans ce chapitre les différentes fonctionnalités qui ont été développées .

## ***Conclusion Générale et Perspectives***

Le projet de fin d'études consistait à développer un système de paiement de portefeuille à portefeuille (P2P), qui donnera la possibilité pour les commerçants de recevoir tous les jetons préférés avec la conversion d'actifs basée sur la blockchain des paiements entrants en temps réel.

Pour mettre en oeuvre ce projet, nous Étions amènes, dans un premier lieu, à établir une étude conceptuelle du sujet afin de dégager les différents modules et fonctionnalités de ce système ainsi qu'une étude des outils et technologies susceptibles de convenir à sa réalisation. Ensuite, nous avons procédé à l'analyse et à la conception du projet sur la base du formalisme UML. Un certain nombre de diagrammes ont été élaborés afin de mieux découper le projet, ce qui a facilité sa mise en oeuvre. Enfin, nous avons mis en oeuvre les divers modules de cette application. Le résultat de cette phase finale respecte les exigences et les besoins déjà définis dans le présent rapport.

L'achèvement de ce projet de graduation a été une opportunité intéressante pour moi d'acquérir de nouvelles connaissances dans le domaine du développement Blockchain.

Finalement, je ne prétends pas avoir fait un travail parfait et il reste à noter que ce projet peut être amélioré en ajoutant plus de fonctionnalités, à savoir :

- **Vérification d'identité dans l'inscription.**
- **Possibilité de mettre en place des portefeuilles.**
- **réaliser des opérations sur la plate-forme.**
- **...**

# ***Bibliographie***

- [1] *Ethereum Projects for Beginners*, Kenny Vaneetvelde, 2018
- [2] *Hands-On Blockchain with Hyperledger*, [Nitin Gaur, Luc Desrosiers, Venkatraman Ramakrishna, Petr Novotny, Dr. Salman A. Baset, Anthony O'Dowd], 2018
- [3] *Solidity Programming Essentials*, Ritesh Modi, 2018
- [4] *Beginning Ethereum Smart Contracts Programming*, Wei-Meng Lee, 2019
- [5] *Ethereum Smart Contract Development in Solidity*, [Gavin Zheng, Longxiang Gao, Liqun Huang, Jian Guan], 2021
- [6] *BLOCKCHAIN Novice to Expert*, Keizer Söze, 2017
- [7] *Blockchain Blueprint for a New Economy*, Melanie Swan, 2015
- [8] *Blockchain Enabled Applications*, [Vikram Dhillon, David Metcalf, Max Hooper], 2017

# *Webographie*

- [1] <https://ethereum.org/fr/>
- [2] <https://solidity-fr.readthedocs.io/fr/latest/>
- [3] <https://www.lemagit.fr/definition/Web-30>
- [4] <https://medium.com/0xcode/hashing-functions-in-solidity-using-keccak256-70779ea55bb0>
- [5] <https://remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.7+commit.e28d00a7.js>
- [6] [https://www.asprom.com/application/blockchain\\_1.pdf](https://www.asprom.com/application/blockchain_1.pdf)
- [7] [https://www.senat.fr/rap/r17-584/r17-584\\_mono.html](https://www.senat.fr/rap/r17-584/r17-584_mono.html)
- [8] <https://www.f5.com/services/resources/glossary/ssl-tls-encryption#:~:text=SSL%20FTLS%20uses%20both%20asymmetric,data%20within%20the%20secured%20session.>
- [9] <https://dev.to/anggapur/user-authentication-with-web3-4he8>
- [10] <https://docs.alchemy.com/>