



République Tunisienne

MINISTÈRE DE L'ENSEIGNEMENT
SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
Université de Sfax
Ecole Nationale d'ingénieur de Sfax



PROJET BLOCKCHAIN INDUSTRIELLE

Détection et Prévention des Dangers Maritimes

Elaboré par :

HASSINE SOUHA

FATNASSI SONIA

SARIA AMANI

SAHLI TAKWA

Encadré par : Frikha Tarek

Année Universitaire : 2024 - 2025

Remerciments

Nous tenons à remercier au début, Dieu le tout-puissant pour nous avoir protégés et guidés pour l'intelligence gratuite à pouvoir rédiger ce projet au matière Blockchain Industrielle.

Nous remercions également tous ceux qui ont participé de près ou de loin par leurs aides précieuses, leurs conseils fructueux à l'élaboration de ce travail dans la meilleure considération.

Nous adressons nos vifs remerciements à **Monsieur Tarek Frikha** pour avoir accepté d'encadrer et d'assurer le suivi de ce travail aussi pour qui nous témoignons notre profond respect et notre immense gratitude pour ses conseils précieux et judicieux.

Finalement, il est essentiel de témoigner notre sincère reconnaissance envers l'administration de l'ENIS pour sa contribution précieuse à la concrétisation de ce projet.



Table des matières

Introduction Générale	1
1 Etude Préalable	3
1.1 Introduction	4
1.2 Présentation du projet	4
1.2.1 Objectif à atteindre	5
1.2.2 Étude de l'existant	5
1.3 Conclusion	9
2 État de l'art	10
2.1 Introduction	11
2.2 Présentation des concepts et des notions théoriques utilisés	11
2.2.1 Hypothèse de solution :	11
2.2.2 L'histoire de l'internet :	12
2.2.3 Introduction à la technologie Blockchain :	15
2.2.4 La Blockchain Ethereum :	30
2.3 IA et Blockchain :	32
2.3.1 Définition Générale de l'Intelligence Artificielle (IA) :	32
2.3.2 Utilité de l'IA lorsqu'elle est intégrée à la Blockchain :	32
2.4 Conclusion	33
3 Etude Conceptuelle	34
3.1 Introduction	35
3.2 Démarche de conception et méthodologie de travail	35
3.2.1 Langage de modélisation UML	35
3.2.2 Les méthodes de conception	36
3.2.3 La méthode utilisée	38
3.2.4 Conception de la technologie de sécurité	39
3.3 Méthodologie de partie IA de travail :	42
3.3.1 Méthodologie CRISP-DM :	42
3.3.2 Justification du choix de CRISP-DM :	43
3.4 Conclusion	44

4 Implémentation de la solution proposée	45
4.1 Introduction	46
4.2 Outils et Environnements de travail	46
4.2.1 Environnement logiciel	46
4.2.2 Languages utilisés	46
4.2.3 Les applications utilisées	49
4.2.4 Les frameworks utilisées pour la programmation	52
4.3 Présentation de l'enchainement de l'application	53
4.3.1 Déploiement du smart contract sur Ganache	53
4.3.2 Description des interfaces d'application	70

Table des figures

2.1	Exemples de domaines d'application de la blockchain	11
2.2	Evolution du Web	12
2.3	L'architecture d'une application Web 2.0	13
2.4	Architecture d'une application web 3.0	16
2.5	Exemple de réseau Peer to peer	17
2.6	Exemple des applications mobiles	18
2.7	Caractéristiques de la blockchain	18
2.8	Structure d'un bloc	21
2.9	Structure d'une chaîne des blocs	21
2.10	Mécanisme de hachage dans la blockchain	21
2.11	Flux de la preuve de travail	22
2.12	Rôle de "Nonce" dans le minage	23
2.13	Minage dans la blockchain	24
2.14	Valeur de "Nonce"	24
2.15	blockchain Consortiale (fédérée)	27
2.16	Les crypto-monnaies les plus connues	29
2.17	Les blockchains les plus connues	29
2.18	Différence entre "smart contract" et "contrat traditionnel"	30
3.1	Processus de développement en Y	38
3.2	Diagramme de cas d'utilisation global	41
3.3	CRISP-DM	44
4.1	télécharger MetaMask	53
4.2	Cliquer sur Ajouter Chrome	54
4.3	Création d'un nouveau compte	55
4.4	Création du mot de passe sur le portefeuille Meta Mask	56
4.5	Phrase secrète de récupération	57
4.6	Confirmer Phrase secrète de récupération	57
4.7	Compte Meta Mask créé avec succès	58
4.8	Site officiel de Ganache	58
4.9	Lancer l'installation	59
4.10	Création d'un workspace	59

4.11 Espace de travail crée	60
4.12 Ajouter un réseau etherum	60
4.13 enregistrer le réseau ganache	61
4.14 réseau ajoutée avec succès	62
4.15 Compte ganache vide d'argent	62
4.16 récupération de l'adresse pour le compte	63
4.17 Préparer un nouveau compte ganache	63
4.18 Importation du compte depuis Ganache	64
4.19 Compte approuvé par MetaMask	64
4.20 Interface d'accueil du RemixIDE	65
4.21 Injecter Notre compte MetaMask avec remix-ide	67
4.22 RemixIDE connectée avec le compte MetaMask	67
4.23 Lancer le déploiement	68
4.24 déploiement réussi avec succès	69
4.25 Lancement de serveur	70
4.26 Présentation de l'application	71
4.27 Présentation de Dataset	71
4.28 Interface d'inscription	72
4.29 Confirmation de la transaction	72
4.30 Interface d'authentification du marin ou utilisateur	73
4.31 Useer connécté-base de donnée	73
4.32 Accès vers l'interface d'ajout	74
4.33 Ajout des features à tester	74
4.34 Liste des données normales dans phpMyAdmin	75
4.35 Résultat de test sans danger	75
4.36 Test de modèle sur features normales	76
4.37 Transaction des de données anormales détectées	76
4.38 Affichage de test de danger	77
4.39 Test de modèle avec valeurs anormales	77

Liste des tableaux

2.1 Comparaison des types de blockchain	28
3.1 Utilisation d'UML dans la conception du projet	40

Liste des abréviations

- ETH** : Ethereum (crypto-monnaie)
- BTC** : Bitcoins
- NFT** : Non-Fungible Tokens
- POW** : Proof of Work
- POS** : Proof of Stake
- PU** : Processus Unifié
- IDE** : Integrated Development Environment
- JSON** : JavaScript Object Notation
- UML** : Unified Modeling Language
- RAD** : Rapid Application Development
- PU** : Processus Unifié
- DCU** : Diagramme de Cas d'Utilisation
- HTML** : HyperText Markup Language
- CSS** : Cascading Style Sheets
- PHP** : Hypertext Preprocessor
- SQL** : Structured Query Language
- API** : Application Programming Interface
- RPC** : Remote Procedure Call
- VM** : Virtual Machine
- P2P** : Peer-to-Peer (Pair-à-pair)
- DLT** : Distributed Ledger Technology (Technologie de registres distribués)
- sBTC** : Wrapped Bitcoin (Une forme de Bitcoin utilisée sur Ethereum)
- ETH** : Ethereum (Une blockchain populaire)
- JSON-RPC** : JavaScript Object Notation - Remote Procedure Call
- 2TUP** : Two-Track Unified Process
- SaaS** : Software as a Service
- EVM** : Ethereum Virtual Machine
- DApps** : Applications décentralisées

IA : Intelligence Artificielle

Introduction Générale

La pollution marine, en particulier celle causée par les hydrocarbures, représente une menace croissante pour les écosystèmes marins et la biodiversité. Cette problématique environnementale complexe requiert une surveillance et une gestion efficaces pour minimiser les impacts négatifs sur les océans. Dans ce contexte, les technologies avancées telles que l'intelligence artificielle (IA) et la blockchain jouent un rôle crucial en offrant des solutions innovantes pour détecter, suivre et gérer la pollution marine en temps réel. La numérisation de la surveillance de l'environnement marin permet non seulement une meilleure gestion des données, mais aussi une traçabilité et une sécurité accrues, essentielles pour une prise de décision rapide et précise.

À l'heure où la pollution des mers par les hydrocarbures continue de croître, l'implémentation de technologies avancées permet d'améliorer l'identification rapide des sources de pollution et d'optimiser les réponses des parties prenantes. L'IA facilite la détection automatique des incidents, tandis que la blockchain garantit la sécurité des données collectées, assurant leur intégrité et leur transparence. Ainsi, ces technologies combinées offrent une solution efficace pour la gestion de la pollution marine, tout en garantissant une meilleure réactivité face aux incidents environnementaux.

C'est dans ce cadre que s'inscrit notre projet, qui vise à développer une plateforme utilisant l'IA et la blockchain pour la détection et la gestion en temps réel de la pollution marine. Cette plateforme permettra d'assurer la traçabilité des données collectées, d'analyser la gravité des anomalies détectées et de notifier les parties prenantes concernées de manière transparente et sécurisée.

Ce rapport se divise en quatre chapitres. Le premier chapitre présente une analyse du cadre du projet, des technologies existantes et des besoins spécifiques de l'application. Le deuxième

chapitre détaille les technologies et concepts clés utilisés, suivis d'une explication de l'approche adoptée pour la conception de la plateforme. Enfin, nous décrirons les étapes de réalisation, les outils choisis, ainsi que les résultats obtenus à travers des exemples d'interface illustrant les fonctionnalités développées.

Chapitre

1

Etude Préalable

1.1 Introduction

Dans ce premier chapitre, nous entamons notre exploration en étudiant l'existant pour identifier les lacunes et proposer des solutions adéquates. Notre objectif est de mettre en lumière les défis à surmonter et de définir les orientations à suivre pour atteindre nos objectifs. Nous commencerons par décrire l'existant, puis nous proposerons des solutions appropriées pour remédier aux lacunes identifiées. Ensuite, nous nous pencherons sur les exigences du projet, tant fonctionnelles que non fonctionnelles, en utilisant des méthodes et des techniques pour recueillir les informations nécessaires.

1.2 Présentation du projet

Dans le contexte dynamique et pressant de la lutte contre la pollution marine, la détection et l'analyse en temps réel des anomalies sont essentielles pour protéger l'écosystème marin. La pollution par hydrocarbures représente une menace majeure pour la biodiversité marine, et son suivi rigoureux devient crucial. Notre projet vise à répondre à ces défis en combinant l'intelligence artificielle (IA) et la blockchain pour une gestion optimisée des données de pollution marine. Selon une étude publiée par l'Organisation des Nations Unies, les marées noires et les fuites d'hydrocarbures sont responsables de la destruction de vastes étendues de faune et de flore sous-marines, affectant gravement les écosystèmes marins. L'IA sera utilisée pour analyser et évaluer la gravité des anomalies détectées, tandis que la blockchain assurera un enregistrement sécurisé et transparent des données critiques, garantissant leur traçabilité. Ces solutions technologiques novatrices visent non seulement à améliorer la détection et la signalisation des incidents en temps réel, mais aussi à garantir la sécurité et l'intégrité des informations collectées. La combinaison de ces technologies représente un pas décisif vers la protection de notre environnement marin tout en assurant une gestion efficace et sécurisée des données.

Notre projet se concentre sur la création d'une application innovante utilisant à la fois la Blockchain et l'IA pour gérer la pollution marine. L'IA analysera en temps réel les anomalies liées à la pollution par hydrocarbures, évaluant leur gravité et facilitant une réponse rapide. La Blockchain, quant à elle, assurera la sécurisation et la traçabilité des données critiques relatives aux incidents de pollution. Cette combinaison de technologies permettra une gestion

1.2. PRÉSENTATION DU PROJET

transparente, sécurisée, and efficace des informations environnementales, contribuant ainsi à la protection des écosystèmes marins.

Des fonctionnalités conviviales seront intégrées à l'application pour répondre aux besoins des acteurs de l'environnement marin, tels que les agences de gestion des risques et les chercheurs. L'adoption de notre plateforme Blockchain dans la gestion de la pollution marine offers de nombreux avantages, notamment une sécurité renforcée, une transparence accrue, and une efficacité optimale in la gestion des incidents. Nous nous engageons à développer et à améliorer continuellement cette application pour suivre les évolutions technologiques et garantir qu'elle reste à la pointe de l'innovation, tout en répondant aux défis croissants de la protection des écosystèmes marins.

1.2.1 Objectif à atteindre

Nous présentons ci-après les principaux objectifs visant à mettre les avancées technologiques au service de la lutte contre la pollution du milieu marin, avec trois axes principaux :

- Développement d'applications auto-configurables sur la blockchain pour assurer la sécurité et la traçabilité des données liées à la pollution marine.
- Mettre en place un système décentralisé garantissant des échanges sécurisés de données environnementales entre les parties impliquées dans la gestion de la pollution marine.
- Utiliser des contrats intelligents pour implémenter une stratégie proactive dans la gestion des données liées aux incidents de pollution marine.

1.2.2 Étude de l'existant

Cette section a pour objectif d'explorer le paysage actuel de la gestion de la pollution marine et des technologies associées. Cette analyse nous permettra de dégager les limites du système actuel et de mettre en évidence les défis rencontrés dans ce domaine. Nous examinerons les insuffisances observées dans la gestion des données environnementales et la surveillance de la pollution marine en temps réel :

A- Descriptif de l'existant :

Le paysage actuel de la gestion de la pollution marine est marqué par une dispersion des informations entre différents systèmes et acteurs, ce qui rend difficile une coordination efficace.

1.2. PRÉSENTATION DU PROJET

Bien que des initiatives telles que Aquaplus, DeepSense AI, Seabin Project, Earthchain, et SmartBuoy utilisent des technologies avancées pour surveiller et détecter la pollution marine, plusieurs défis subsistent. Ces systèmes, qui combinent l'intelligence artificielle et la blockchain, permettent de détecter et d'identifier rapidement les sources de pollution et d'envoyer des notifications aux parties prenantes concernées. Cependant, des vulnérabilités persistent dans la gestion et la sécurisation des données collectées en temps réel, nécessitant une surveillance continue.

La traçabilité et la sécurité des données restent des préoccupations majeures, car la diversité des technologies utilisées et des plateformes impliquées complique l'intégrité des informations. Malgré des efforts pour garantir la sécurité des données, il existe encore des lacunes concernant la vérification de l'authenticité des informations et la traçabilité des incidents de pollution. C'est dans ce contexte que l'intégration de la blockchain pourrait représenter une avancée significative pour améliorer la gestion des données environnementales, garantir la transparence et assurer la sécurité des informations collectées, tout en permettant une réponse plus rapide et plus coordonnée face à la pollution marine.

B- Critique de l'existant :

Cette section a pour objectif de critiquer l'état actuel de la gestion de la pollution marine et des technologies de surveillance associées, en mettant en évidence les limites et les défis rencontrés. Il est essentiel d'examiner attentivement les systèmes et les pratiques en place, ainsi que les études et recherches pertinentes, afin de comprendre les lacunes et d'identifier les opportunités d'amélioration.

Bien que des technologies avancées telles que l'intelligence artificielle et la blockchain soient utilisées pour surveiller la pollution marine, plusieurs limitations subsistent :

Cette section a pour objectif de critiquer l'état actuel de la gestion de la pollution marine et des technologies de surveillance associées, en mettant en évidence les limites et les défis rencontrés. Il est essentiel d'examiner attentivement les systèmes et les pratiques en place, ainsi que les études et recherches pertinentes, afin de comprendre les lacunes et d'identifier les opportunités d'amélioration. Bien que des technologies avancées telles que l'intelligence artificielle et la blockchain soient utilisées pour surveiller la pollution marine, plusieurs limitations subsistent :

1.2. PRÉSENTATION DU PROJET

- Coûts élevés : Les systèmes de surveillance en temps réel de la pollution marine, notamment les bouées intelligentes et les capteurs, peuvent être coûteux à déployer et à maintenir, ce qui limite leur accessibilité et leur adoption à grande échelle.
- Faible précision dans des conditions météorologiques extrêmes : Les technologies actuelles rencontrent des difficultés pour fonctionner avec précision dans des conditions météorologiques difficiles, telles que des nuages, des tempêtes ou des vagues importantes, ce qui peut affecter la fiabilité des données collectées.
- Couverture limitée : Malgré les efforts pour déployer des capteurs et des bouées, la couverture géographique reste limitée, particulièrement dans les zones maritimes éloignées ou difficiles d'accès, rendant difficile la surveillance complète des océans.
- Manque de précision pour détecter spécifiquement les déversements d'hydrocarbures : Les systèmes de détection actuels manquent parfois de précision pour identifier les déversements d'hydrocarbures de manière fiable, ce qui peut entraîner des retards dans la prise de décision et la réponse aux incidents.
- Complexité d'intégration : L'intégration de ces technologies dans des systèmes existants de gestion de la pollution marine est souvent complexe et nécessite des ajustements techniques importants pour garantir leur compatibilité et leur efficacité.
- Fiabilité : Les systèmes actuels souffrent parfois de problèmes de fiabilité, avec des erreurs ou des défaillances pouvant affecter la précision des données ou la réactivité face aux incidents de pollution.
- Limites techniques : Les technologies actuelles présentent des limites techniques, notamment en termes de capacité à analyser de grandes quantités de données en temps réel ou à prédire les incidents de pollution avant qu'ils ne se produisent. Dans ce contexte, il devient évident qu'il existe des opportunités pour améliorer les systèmes de gestion de la pollution marine, en particulier par l'intégration de solutions plus fiables et précises, telles que la blockchain et des techniques d'IA plus avancées. Ces technologies pourraient offrir des solutions pour surmonter certaines des limitations actuelles, notamment en termes de sécurité des données, de traçabilité et de réactivité.

Dans ce contexte, il devient évident qu'il existe des opportunités pour améliorer les systèmes de gestion de la pollution marine, en particulier par l'intégration de solutions plus fiables et

1.2. PRÉSENTATION DU PROJET

précises, telles que la blockchain et des techniques d'IA plus avancées. Ces technologies pourraient offrir des solutions pour surmonter certaines des limitations actuelles, notamment en termes de sécurité des données, de traçabilité et de réactivité.

C- Solution Proposée :

Après une analyse approfondie de l'existant, plusieurs limites des systèmes actuels de surveillance de la pollution marine ont été identifiées. Cela met en évidence la nécessité de développer une solution complète capable de relever ces défis. Dans cette optique, nous avons proposé la conception et l'implémentation d'une plateforme dédiée à la détection et à la gestion des déversements d'hydrocarbures. Cette plateforme vise à :

Détecter précisément les déversements d'hydrocarbures grâce à l'intelligence artificielle, réduisant ainsi les risques de confusion avec d'autres types de pollution et permettant une intervention rapide et ciblée.

Notifier instantanément les parties prenantes via une application Flutter, leur offrant une interface intuitive pour réagir efficacement aux incidents signalés.

Sécuriser les données critiques collectées en utilisant la technologie blockchain, garantissant ainsi la traçabilité et l'intégrité des informations tout au long de leur cycle de vie, et renforçant la confiance des utilisateurs.

Fournir une accessibilité simplifiée grâce à une interface multiplateforme combinée à une intégration optimisée de l'intelligence artificielle et de la blockchain, ce qui contribue à réduire les coûts opérationnels tout en améliorant l'efficacité globale du système.

En intégrant ces fonctionnalités essentielles, notre plateforme se distingue par sa capacité à répondre de manière efficace, sécurisée et innovante aux besoins spécifiques de la surveillance de la pollution marine, offrant ainsi une solution moderne et adaptée aux défis actuels.

1.3 Conclusion

Dans ce premier chapitre, nous avons défini le champ de notre étude, suivi d'une analyse de l'existant afin de préciser nos objectifs à atteindre. En effet, cette analyse nous a permis de préparer une conception solide pour les améliorations que nous allons ajouter à la solution proposée afin de répondre à nos besoins. Dans le chapitre 2, nous présenterons les démarches de développement ainsi que l'étude de l'état de l'art de notre solution.

Chapitre **2**

État de l'art

2.1 Introduction

Après avoir défini les objectifs et réalisé l’analyse nécessaire pour élaborer la solution finale dans le premier chapitre, une étape préliminaire à la réalisation de l’application, nous permet de présenter les concepts clés et les technologies nécessaires à utiliser tout au long de notre implémentation. Ensuite, le chapitre 2 offre une synthèse succincte de notre exploration du domaine, fournissant ainsi une base solide pour la mise en œuvre de notre application.

2.2 Présentation des concepts et des notions théoriques utilisés

2.2.1 Hypothèse de solution :

Dans le domaine de la surveillance de la pollution marine, la technologie Blockchain constitue une solution innovante pour relever certains défis critiques, notamment ceux liés à la sécurisation des données environnementales, au suivi des incidents de pollution et à la traçabilité des interventions, comme suggéré dans la figure 2.1. Sa nature décentralisée et immuable fournit un cadre sécurisé pour protéger les données critiques, garantir leur intégrité et atténuer les vulnérabilités souvent rencontrées dans les systèmes traditionnels de gestion de données environnementales.

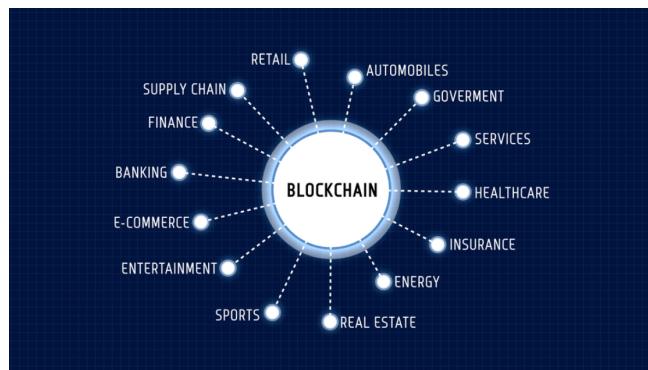


FIGURE 2.1 – Exemples de domaines d’application de la blockchain

Alors, nous avons pris la décision stratégique de migrer vers Ethereum. Les fonctionnalités avancées et les avantages spécifiques qu’offre cette plateforme correspondent parfaitement à nos objectifs de surveillance de la pollution marine. Ainsi, notre équipe s’est engagée à travailler

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

avec cette technologie pour concevoir et mettre en œuvre une solution innovante qui intègre des données sécurisées, traçables et fiables, en réponse aux défis critiques du suivi et de la gestion des incidents environnementaux en milieu marin.

2.2.2 L'histoire de l'internet :

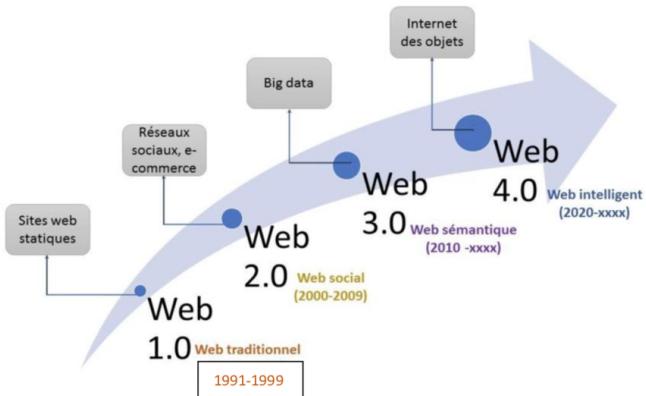


FIGURE 2.2 – Evolution du Web

Pour que nous puissions comprendre ce que nous entendons par «**Web 3.0/Blockchain**», nous devons revenir au Web 1.0 comme indiqué au figure 2.2 et commencer par là :

A- Phase 1 (première phase) Web 1.0

Le Web 1.0, aussi appelé Web classique, représentait une étape statique dans l'évolution d'Internet, s'étendant de la fin des années 1980 jusqu'environ 2005. Durant cette période, les sites web étaient principalement axés sur la diffusion d'informations et manquaient d'interactivité. Les utilisateurs jouaient un rôle de spectateurs passifs, les sites se concentrant sur la présentation de produits avec une interaction limitée. L'avènement des premiers sites d'e-commerce a marqué cette ère, mais les coûts élevés des logiciels propriétaires et l'éclatement de la bulle dot.com en 2000 ont remis en question cette approche. En somme, le Web 1.0 se caractérisait par un accès à Internet principalement orienté vers la consultation de contenu plutôt que vers des interactions dynamiques.

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

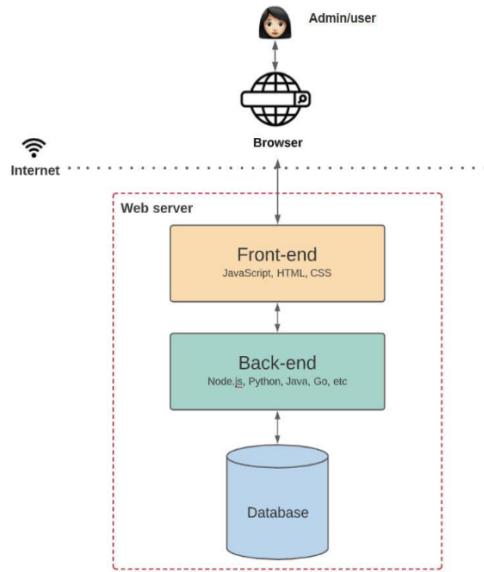


FIGURE 2.3 – L'architecture d'une application Web 2.0

B- Phase 2 (deuxième phase) Web 2.0

Le Web 2.0, qui correspond au Web actuel, se distingue par une interaction dynamique [2.3](#) entre les utilisateurs et le contenu en ligne. Par exemple, Medium est une plateforme de blogs qui permet aux utilisateurs de publier et d'interagir avec du contenu. Pour fonctionner, Medium nécessite une architecture complexe comprenant :

1. **Une base de données** pour stocker les informations telles que les utilisateurs, les publications, les commentaires, etc.
2. **Un code back-end** qui définit la logique métier de la plateforme, comme l'inscription d'un nouvel utilisateur, la publication d'un blog ou les commentaires sur un blog existant.
3. **Un code front-end** qui définit la logique de l'interface utilisateur de Medium, comme l'apparence du site et les actions lorsqu'un utilisateur interagit avec chaque élément de la page.

Cette architecture est typique des applications Web 2.0, qui dépendent de serveurs centralisés pour stocker les données et exécuter le code back-end.

Mais tout cela change. La technologie Blockchain a ouvert une nouvelle direction passionnante pour les applications décentralisées/Web 3.0

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

C- Phase 3 (troisième phase) Web 3.0

Le terme "Web 3.0" a évolué pour signifier bien plus que le simple Web sémantique. Les passionnés de blockchain qui développent des DApps (applications décentralisées) utilisent ce terme pour décrire l'idée de créer des applications sur une architecture ouverte et décentralisée. L'objectif principal du Web 3.0 est d'améliorer l'intelligence, l'autonomie et l'ouverture d'Internet en utilisant des technologies telles que l'intelligence artificielle (IA), la blockchain, les graphiques 3D, la réalité virtuelle et la réalité augmentée.

L'IA peut aider à mieux comprendre et interpréter le contenu sur le Web, tandis que les blockchains peuvent être utilisées pour créer des applications sur des protocoles décentralisés, éliminant ainsi les intermédiaires de confiance et offrant une vérifiabilité publique. La cryptomonnaie peut transformer complètement le modèle commercial des applications sur Internet, en remplaçant le modèle basé sur la publicité actuel. En bref, une explication intéressante de ces phases est celle donnée par Tim Berner-Lee, le fondateur du World Wide Web, a donné une explication intéressante de ces phases :

- Le Web 1.0 est la phase "lisible" du Web, avec une interaction limitée entre les utilisateurs.
- Le Web 2.0 est la phase "inscriptible" du Web, où les utilisateurs peuvent interagir avec le site et entre eux.
- Le Web 3.0 est la phase "exécutable" du Web, où les ordinateurs peuvent interpréter les informations comme les humains, pour ensuite générer un contenu personnalisé pour les utilisateurs. Cette dernière phase vise à permettre aux ordinateurs de comprendre et d'interpréter les informations comme le font les humains, afin de proposer un contenu adapté aux utilisateurs.

Alors, en quoi le Web 3.0 est-il différent d'un point de vue approfondi ?

Le Web 3.0 se distingue des applications Web 2.0 telles que Medium en éliminant les intermédiaires. Il n'y a pas de base de données centralisée pour stocker les données de l'application, ni de serveur Web centralisé pour la logique back-end. Au lieu de cela, la blockchain est utilisée pour créer des applications sur une machine d'état décentralisée, gérée par des nœuds anonymes sur Internet. Les blockchains sont des machines d'état qui sont instanciées avec un état de genèse et ont des règles strictes pour définir comment l'état peut évoluer. Aucune entité ne contrôle cette machine d'état décentralisée, qui est dirigée collectivement par tous les membres

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

du réseau. En résumé, le Web 3.0 utilise la blockchain pour créer des applications décentralisées qui éliminent les intermédiaires et sont gérées collectivement par les membres du réseau.

C-1) Le Back-end d'une application Web 3.0 :

Au lieu de la façon dont le back-end de "Medium" était contrôlé, le Web 3.0, nous permet à écrire des contrats intelligents qui définissent la logique de nos applications et les déploient sur la machine d'état décentralisée. Cela signifie que chaque personne qui souhaite créer une application blockchain déploie son code sur cette machine d'états partagée. Un contrat intelligent peut remplacer un composant côté serveur (alias "back-end") dans une application standard. Par conséquent, le back-end d'une DApp repose sur un contrat intelligent qui définit sa logique. Ces contrats intelligents peuvent être programmés dans des langages de haut niveau tels que Solidity ou Vyper.

C-2) Le Front-end d'une application Web 3.0 :

Le côté client d'une DApp définit la logique de l'interface utilisateur et nous utilisons simplement des technologies Web standard telles que HTML, CSS, JavaScript, etc. Cela nous permet comme un développeur Web traditionnel d'utiliser des outils, des bibliothèques et des Framework familiers. Cela reste à peu près le même que le front-end en Web 2.0, à quelques exceptions près, sur lesquelles nous reviendrons plus tard. Les interactions avec la blockchain Ethereum, telles que la signature numérique de messages, l'envoi de transactions et la gestion de clés publiques et privées, se font souvent via le navigateur Web, via le portefeuille électronique MetaMask. Le front-end est généralement lié à la blockchain Ethereum via son API JavaScript, le web3.js, qui est fournie avec les packages d'application et les dépendances. Voici à quoi ressemble l'architecture [2.4](#) :

2.2.3 Introduction à la technologie Blockchain :

A- Définition et historique :

La blockchain, également connue sous le nom de chaîne de blocs, est une technologie de registres distribués (DLT, ou "Distributed Ledger Technology") utilisée pour créer un système de stockage de données de manière distribuée et immuable. Cela permet un enregistrement chronologique et sécurisé des transactions réalisées entre les membres d'un réseau. Initialement

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

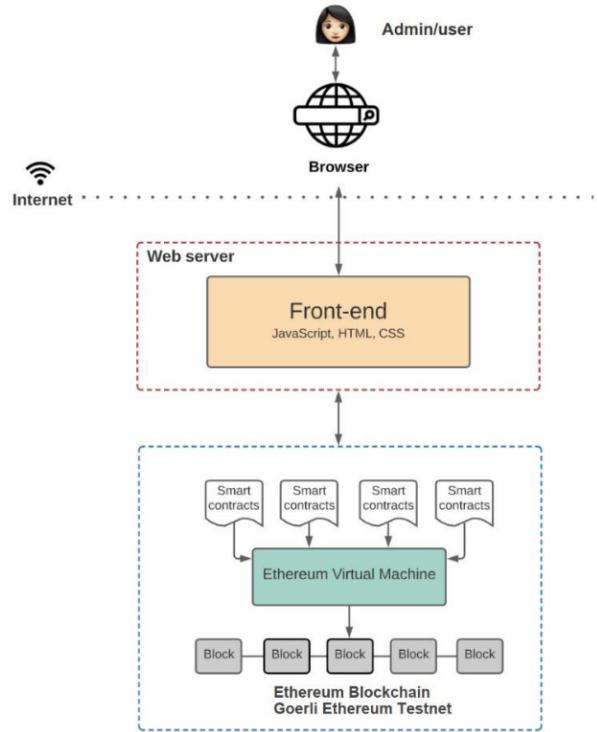


FIGURE 2.4 – Architecture d'une application web 3.0

conçue comme un élément central du Bitcoin, une cryptomonnaie révolutionnaire présentée dans le livre blanc de Satoshi Nakamoto, publié en 2008 et suivi en 2009 par l'implémentation de référence du logiciel. La blockchain a depuis élargi son champ d'application bien au-delà des cryptomonnaies. Nakamoto est reconnu comme le concepteur de la première base de données construite avec une chaîne de blocs, résolvant ainsi le problème de la double dépense pour une monnaie numérique via un réseau pair-à-pair. Son potentiel disruptif réside dans sa capacité à établir un consensus de confiance dans des scénarios où plusieurs parties, souvent anonymes et peu fiables, doivent stocker un état (transaction). En exploitant la vérifiabilité publique pour garantir la sécurité cryptographique des transactions, la blockchain offre une traçabilité vérifiable, ouvrant ainsi la voie à diverses applications dans les services financiers, tels que les actifs numériques, les transferts de fonds et les paiements en ligne. Un exemple emblématique de blockchain est Ethereum, une base de données partagée entre tous les membres d'un réseau qui s'accroît sans interruption et facilite les transactions entre ses utilisateurs.

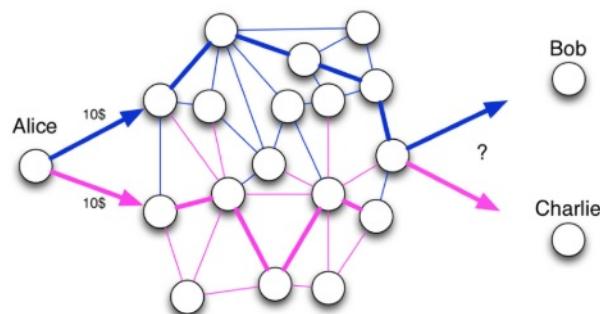
2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

B- Principe et fonctionnement de la technologie Blockchain :

La technologie de la blockchain repose sur trois principes fondamentaux : la décentralisation, l'ouverture et l'immuabilité. Elle offre un registre partagé où aucune personne ou entité n'a le contrôle absolu, permettant ainsi à tout le monde d'y participer. Une fois que les données sont enregistrées dans la blockchain, elles deviennent immuables, impossible à altérer ou supprimer. Ces caractéristiques assurent une sécurité inhérente au système, en particulier lorsqu'il s'agit de transactions effectuées sans passer par une tierce partie, comme dans le cas de Bitcoin.

Prenons **un exemple** présenté au figure 2.5 pour illustrer son fonctionnement : sur **un réseau peer-to-peer (P2P)** comme celui de la blockchain, des transactions sont effectuées entre participants sans l'intermédiation d'une autorité centrale.

En résolvant le défi de la réconciliation des transactions dans un réseau décentralisé, un consensus est atteint sur l'ordre des événements, garantissant ainsi la validité des transactions sans nécessiter une tierce partie de confiance. Cela signifie que si Alice envoie 10 € à la fois à Bob et à Charlie, le réseau détermine qui recevra finalement les fonds en fonction de l'ordre dans lequel les transactions ont été validées.



2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

contrôle accroît la confiance des utilisateurs dans la sécurité de leurs données, ouvrant ainsi la voie à des services plus avancés et sécurisés.

Ci-dessous est présenté un schéma très général [2.6] correspondant à l'ajout d'un bloc de données transactionnelles sur la blockchain :

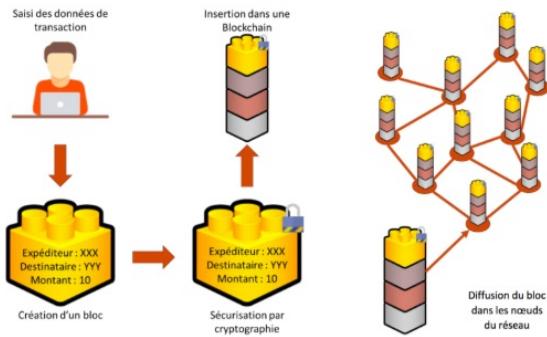


FIGURE 2.6 – Exemple des applications mobiles

C- Caractéristiques des systèmes Blockchain :

De manière générale et comme présenté dans la figure 2.7, la Blockchain présente les caractéristiques clés suivantes :

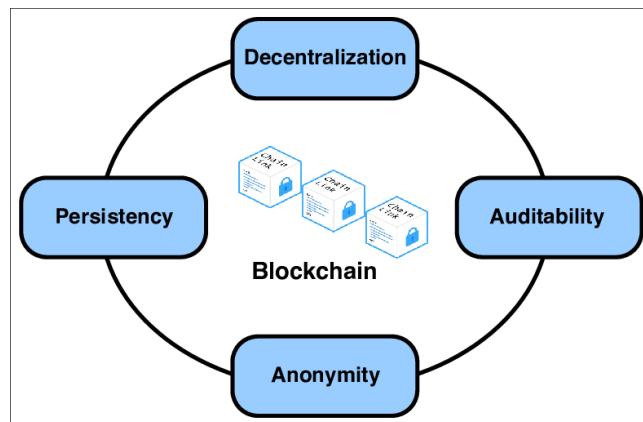


FIGURE 2.7 – Caractéristiques de la blockchain

- Décentralisation : Dans les systèmes de transactions centralisés conventionnels, chaque transaction doit être validée par l'agence centrale de confiance (par exemple, la banque centrale), ce qui entraîne inévitablement des goulots d'étranglement en termes de coûts et de performances au niveau des serveurs centraux. D'une manière différente, une transaction dans le

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

réseau Blockchain peut être effectuée entre deux pairs (P2P) sans authentification par l'agence centrale. De cette manière, Blockchain peut réduire considérablement les coûts du serveur et atténuer les goulets d'étranglement des performances du serveur central.

- Persistance : Étant donné que chacune des transactions réparties sur le réseau doit être confirmée et enregistrée dans des blocs répartis sur l'ensemble du réseau, il est presque impossible de la falsifier.
- Un registre numérique : est une structure qui héberge toutes ces transactions. En clair, le grand livre numérique fonctionne comme une feuille de calcul qui contient chaque nœud d'un réseau et enregistre chaque achat effectué par ce nœud. La signature numérique protège les informations du grand livre numérique contre la falsification et garantit qu'elles sont exceptionnellement sécurisées. La caractéristique la plus intrigante de ce grand livre est que même si tout le monde peut consulter les données, elles ne peuvent pas être modifiées. De plus, chaque bloc diffusé serait validé par d'autres nœuds et les transactions seraient vérifiées. Ainsi, toute falsification pourrait être facilement détectée.
- Anonymat : Chaque utilisateur peut interagir avec le réseau Blockchain avec une adresse générée. De plus, un utilisateur pourrait générer de nombreuses adresses pour éviter de révéler son identité. Il n'existe plus de partie centrale qui conserve les informations privées des utilisateurs. Ce mécanisme préserve une certaine confidentialité sur les transactions incluses dans la Blockchain. Il est important de noter que la Blockchain ne peut garantir la parfaite préservation de la vie privée en raison de la contrainte intrinsèque.
- Auditabilité : Étant donné que chacune des transactions sur la Blockchain est validée et enregistrée avec un horodatage, les utilisateurs peuvent facilement vérifier et retracer les enregistrements précédents en accédant à n'importe quel nœud du réseau distribué. Dans Bitcoin, chaque transaction peut être retracée de manière itérative aux transactions précédentes. Il améliore la traçabilité et la transparence des données stockées dans la Blockchain.

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

D- Composants de base de l'architecture Blockchain :

Voici la liste des composants de base de l'architecture blockchain :

-Nœud : Un nœud est un périphérique qui est connecté à un réseau peer-to-peer. Ils ont répandu les transactions et les blocs partout. Il existe de nombreux types de nœuds différents dans une blockchain et chaque nœud a un rôle dans le réseau.

-Transaction : Les transactions sont les unités de travail individuelles effectuées pour mettre à jour la blockchain. Ils contiennent l'adresse de l'expéditeur, l'adresse du destinataire, le nombre de "pièces de crypto-monnaie" transférées ou une adresse de contrat intelligent sur une blockchain comme Ethereum, et combien de frais de gaz sont mesurés pour enregistrer cette transaction. Lorsqu'une nouvelle transaction est créée, elle est ajoutée à un pool de transactions non confirmées. Les mineurs récupèrent ensuite ces transactions et les incluent dans un nouveau bloc. Dès qu'un mineur crée un bloc, les transactions de ce bloc sont considérées comme confirmées.

-FRAIS DE GAZ : Les frais de gaz mesurent exactement la puissance de calcul requise par une transaction donnée avant de pouvoir être enregistrée sur un réseau blockchain.

Nous avons besoin d'ETH pour payer les frais de gaz lors de transactions sur Ethereum, sBTC lors de transactions sur Bitcoin, etc. Plus la demande de calcul requise par notre transaction est élevée, plus les frais de gaz que nous devons payer aux mineurs du réseau sont élevés.

-Bloc : Les blocs sont des lots de transactions qui entrent dans la blockchain. Un bloc est créé lorsque les transactions sont vérifiées et ajoutées à la blockchain. Le bloc contient un hachage cryptographique du bloc précédent, qui crée une chaîne de blocs (d'où "blockchain"). Cela empêche quiconque de falsifier les données d'un bloc sans invalider tous les blocs suivants. Chaque bloc stocke un blob de données (qui est généralement une liste de transactions), son numéro de bloc et le hachage du bloc précédent. Chaque bloc est représenté par un objet JSON[2.8] , comme illustré ci-dessous[2.9] :

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

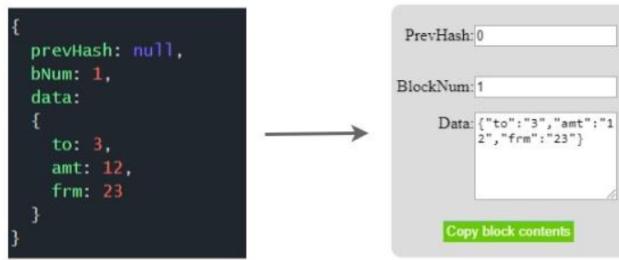


FIGURE 2.8 – Structure d'un bloc

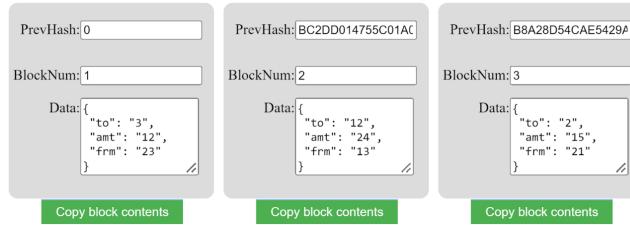


FIGURE 2.9 – Structure d'une chaîne des blocs

Nous avons imaginé qu'il y ait trois autres transactions à enregistrer et qu'un nouveau bloc doive être ajouté à l'exemple de blockchain présenté ci-dessus. Voici comment la valeur de PrevHash sera calculée afin d'être ajoutée dans le nouveau BlockNum 4[2.10] :

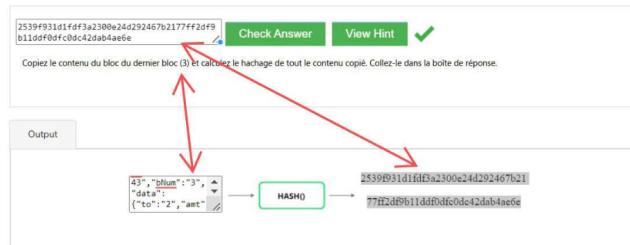


FIGURE 2.10 – Mécanisme de hachage dans la blockchain

-Chaîne : Lorsque plusieurs blocs sont successivement reliés dans un ordre spécifique, cela forme une chaîne.

-Mineurs : Ce sont des nœuds du réseau qui examinent et valident les blocs avant de permettre leur ajout à la structure de la blockchain.

-Consensus (protocole de consensus) : Le protocole de consensus garantit que chaque nouveau bloc ajouté à la blockchain est accepté par tous les nœuds du réseau. Cet algorithme vise à parvenir à un accord mutuel bénéfique à l'ensemble du réseau.

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

E- Mécanismes de consensus pour les blockchains publiques :

Les deux mécanismes de consensus les plus notables pour les “blockchains” publiques sont la preuve de travail ou POW (Proof of Work) et la preuve de participation ou POS (Proof of Stake) :

E-1) La preuve de travail (POW) :

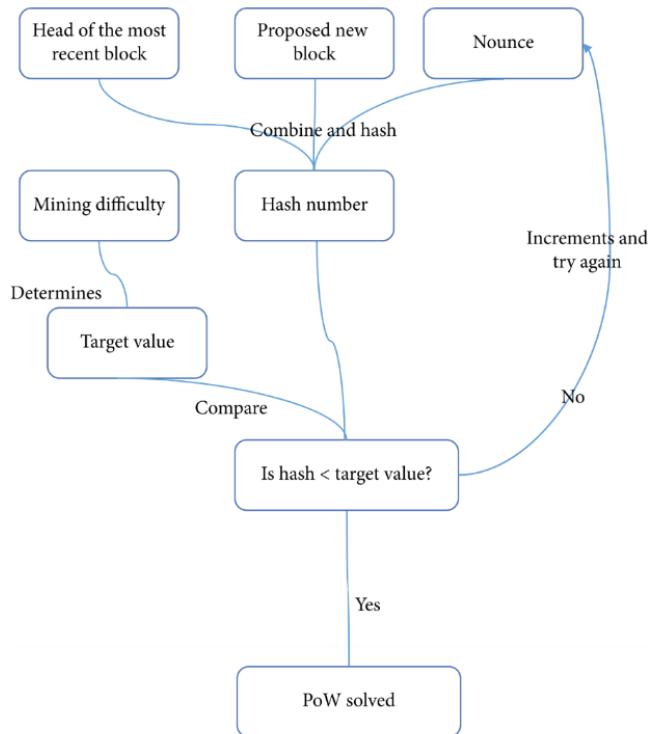


FIGURE 2.11 – Flux de la preuve de travail

La preuve de travail, comme présentée dans la figure [2.11] , est l'un des mécanismes de consensus les plus anciens qui a vu le jour dans les années 1990. Il s'agit d'un algorithme de consensus pour confirmer les transactions et créer un nouveau bloc sur la Blockchain. Il était et est toujours utilisé par divers projets de crypto-monnaie comme Bitcoin, Litecoin et Ethereum 1.0. La preuve de travail utilise la puissance de calcul des ordinateurs du réseau pour créer de nouvelles pièces virtuelles (ou “Coins”). Dans cet algorithme, les mineurs (nœuds spécialisés impliqués dans la création et l'ajout des blocs sur Blockchain) s'affrontent pour résoudre des problèmes mathématiques complexes, c'est-à-dire trouver la valeur nonce afin de terminer la transaction sur le réseau. Le nonce est chargé de générer une valeur de hachage valide pour le

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

bloc pour le rendre valide et de l'ajouter à la blockchain existante. Le processus de validation des transactions et de leur ajout à la Blockchain est appelé minage (ou “Mining”). Dès qu'un mineur crée avec succès un bloc valide, il est récompensé par de nouvelles pièces virtuelles (ou “Coins”) d'une valeur des frais de gaz calculés en fonction des transactions de ce bloc, par exemple des Bitcoins (BTC) dans le réseau Bitcoin, ou des ETH dans le réseau Ethereum. La preuve de travail (POW) est adoptée par Bitcoin, Ethereum, etc.

- Extraction (ou “Mining”) de crypto-monnaie :

Dans le processus d'extraction de la crypto-monnaie d'une blockchain spécifique, par exemple, dans Bitcoin ou Ethereum, un champ supplémentaire est ajouté dans chaque bloc, appelé nonce, comme mentionné dans la figure [2.12].

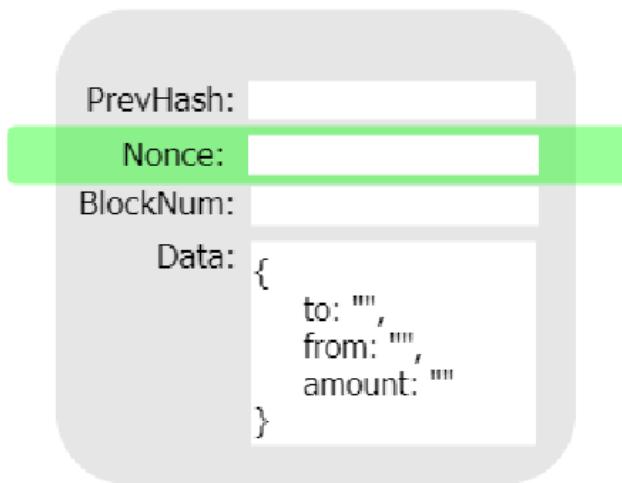


FIGURE 2.12 – Rôle de ”Nonce” dans le minage

Les mineurs reçoivent en continu de nouvelles transactions à valider sur la chaîne. Pour qu'une transaction soit confirmée, elle doit être incluse dans un bloc qui doit être créé. Le processus est relativement rapide : il suffit d'ajouter les transactions au bloc, de calculer le hachage du bloc précédent, d'incrémenter le numéro de bloc, d'ajouter la liste des transactions et de calculer le hachage du bloc. Cependant, pour encourager les mineurs à participer au réseau, ils reçoivent des récompenses en crypto-monnaie pour la création de nouveaux blocs. Par exemple, les mineurs de Bitcoin reçoivent des Bitcoins nouvellement créés, tandis que ceux d'Ethereum reçoivent des Ethers[2.13] .

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

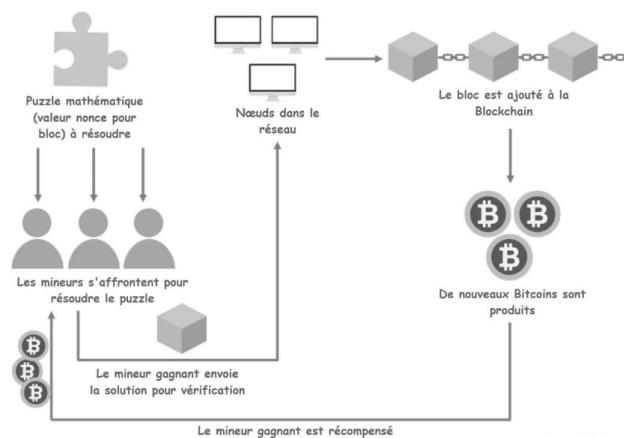


FIGURE 2.13 – Minage dans la blockchain

-Niveau de difficulté et valeur Nonce :

Un hachage valide pour un bloc est un hachage qui respecte un certain niveau de difficulté et qui est inférieur ou égal à la cible de hachage (ou "cible de hachage"). Par exemple, si la cible de hachage est 000032c2d3e96 comme présenté au figure 2.14 , tout hachage inférieur ou égal à la cible de hachage est un hachage valide pour le bloc. De plus, la difficulté détermine le nombre de zéros prédéfinis au début du hachage valide. Dans ce cas, le hachage valide doit être inférieur ou égal à la cible de hachage (ou "cible de hachage") et doit commencer par 0000. Plus de zéros au début du hachage signifient une difficulté plus élevée.



FIGURE 2.14 – Valeur de "Nonce"

Le hachage valide pour le bloc contient une valeur inférieure ou égale à la cible de hachage. Maintenant, puisque les blocs Blockchain sont hachés par des fonctions de hachage de

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

cryptographie, le nonce ne peut pas vraiment être "calculé" car les fonctions de hachage de cryptographie sont des fonctions à sens unique. Ainsi, les mineurs effectuent une sorte d'algorithm d'essai et d'essai pour voir s'ils peuvent trouver un nonce qui satisfait à l'exigence (X zéros non significatifs) pour le bloc suivant.

E-2) La preuve de participation (POS) :

La preuve de participation (POS) a été conçue en tant qu'alternative à la preuve de travail (POW), visant à réduire la quantité de calcul nécessaire pour vérifier les blocs et les transactions sur la blockchain, augmentant ainsi sa sécurité. Cette méthode modifie le processus de vérification des blocs en impliquant les détenteurs de crypto-monnaie, qui offrent leurs avoirs comme garantie pour avoir le droit de valider des blocs. Ces détenteurs deviennent alors des validateurs, sélectionnés de manière aléatoire pour valider les blocs, contrairement à la preuve de travail qui repose sur la compétition. Cela transforme le processus de validation en un partage des responsabilités plutôt qu'en une compétition.

Pour devenir validateur, un détenteur de crypto-monnaie doit mettre en jeu une certaine quantité de pièces. Par exemple, dans la future version d'Ethereum, la 2.0, un utilisateur devra bloquer 32 ETH pour devenir validateur. Les blocs sont validés par plusieurs validateurs, et une fois qu'un nombre spécifique de validateurs confirme l'exactitude du bloc, celui-ci est finalisé et scellé.

F- Rôle de la cryptographie dans la blockchain :

Les fonctions de hachage, les paires de clés publiques-privées et les signatures numériques constituent ensemble la base de la blockchain. Ces fonctionnalités cryptographiques permettent aux blocs d'être liés en toute sécurité à d'autres blocs, et garantissent également la fiabilité et l'immuabilité des données stockées sur la blockchain.

Les blockchains sont concrètement utilisées dans le domaine des crypto-monnaies, où des paires de clés publiques-privées sont employées pour enregistrer les adresses des utilisateurs sur la blockchain. La cryptographie dans la blockchain garantit que la clé publique fait office d'adresse de l'utilisateur, visible par tous les participants au réseau. En revanche, la clé privée est une information confidentielle utilisée pour accéder aux données d'adresse et autoriser des actions telles que des transactions. Les signatures numériques sont également largement utilisées dans

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

les crypto-monnaies. Elles permettent d'approuver les transactions en les signant de manière sécurisée et hors ligne, et sont également utilisées pour les contrats multisignatures et les portefeuilles numériques sur la blockchain. Cela permet d'effectuer des actions à partir de ces contrats et portefeuilles, à condition que les signatures numériques de plusieurs clés privées.

G- Classification des systèmes Blockchain :

Il existe trois types de Blockchain : publique, privée (autorisée) et consortium (fédérée). Pour cela dans notre projet nous avons choisi d'utiliser une blockchain privée ou autorisée. Cette décision découle de notre besoin d'assurer le contrôler strictement l'accès aux données médicales sensibles et de maintenir une certaine confidentialité. Ainsi elle permet à un groupe sélectionné d'entités ou d'organisations de gérer le réseau, ce qui peut être bénéfique pour assurer la conformité aux réglementations en matière de confidentialité des données.

G-1) La blockchain publique : La blockchain publique est entièrement décentralisée et accessible à tous via une connexion Internet. Elle permet à quiconque de consulter, enregistrer ou vérifier les transactions, offrant un accès ouvert et transparent, indépendamment de la localisation géographique.

G-2) Les blockchains Privées : La blockchain avec autorisation est un système fermé où une autorité centrale régule l'accès, le consensus et les activités du réseau. Les participants, dont les identités sont surveillées, bénéficient de contrôles d'accès spécifiques, ce qui rend le système centralisé malgré l'utilisation de registres distribués.

G-3) Consortium ou blockchain fédérée :

Le troisième type de blockchain est la blockchain de consortium [2.15] , qui vise à éliminer l'autonomie unique accordée à une seule entité comme dans une blockchain privée. Dans le cas d'une blockchain de consortium, contrairement à une blockchain privée, plusieurs entités sont présentes sur le réseau. Comme il n'y a pas d'autorité unique gouvernant le contrôle, elle conserve un caractère décentralisé. Un groupe d'entreprises ou d'individus prenant des décisions dans le meilleur intérêt de l'ensemble du réseau est présent, appelé consortiums ou fédérations, d'où le nom de blockchain de consortium ou fédérée. Ce type de blockchain est particulièrement intéressant pour ses nombreux avantages pour les membres du réseau, notamment la simplification des processus clients et la facilitation des échanges de données intra-industrie sécurisés

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

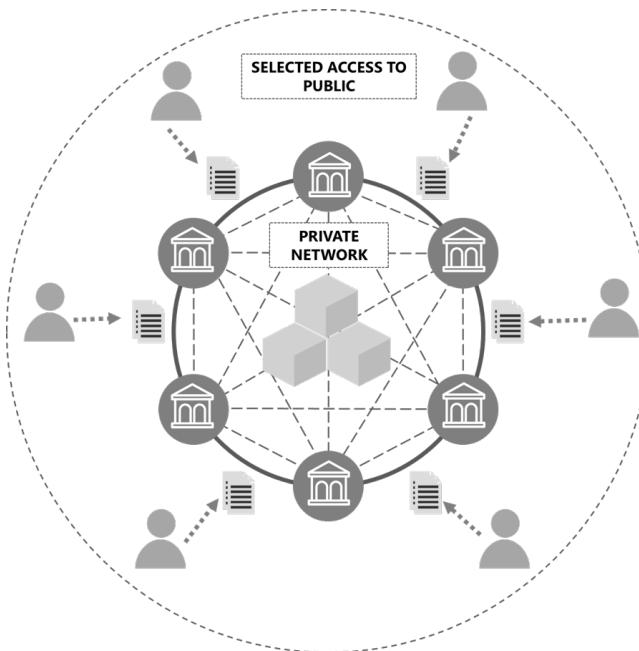


FIGURE 2.15 – blockchain Consortiale (fédérée)

et confidentiels.

Mais bien sur il y a plusieurs autres types de blockchain mais nous avons énumérer les plus remarquables. Pour faire cloturer nous avons fait un tableau comparatif des différents types de la Blockchain comme indique le tableau ci-dessous :

H- Applications de la blockchain :

Les applications des blockchains sont infinies. Il peut être utilisé pour presque tout. Voici quelques-unes des applications les plus remarquables et fonctionnelles de la blockchain.

H-1) Les crypto-monnaies :

Les crypto-monnaies [2.16] sont des actifs numériques virtuels qui reposent sur la technologie de la blockchain, une chaîne de blocs décentralisée et cryptée. Contrairement aux monnaies traditionnelles, leur valeur est déterminée par l'offre et la demande et elles ne dépendent pas d'une autorité centrale comme une banque centrale.

H-2) Principales applications Blockchain au-delà des crypto-monnaies à connaître :

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

TABLE 2.1 – Comparaison des types de blockchain

Caractéristiques	Blockchain Publique	Blockchain Privée	Blockchain Consortium
Accès	Public, ouvert à tous	Restreint, contrôlé par une seule entité ou un groupe restreint d'entités	Restreint, partagé entre les membres du consortium
Confidentialité	Faible (toutes les transactions sont publiques)	Élevée (accès restreint aux participants autorisés)	Moyenne (accès limité aux membres du consortium)
Vitesse de transaction	Variable, généralement plus lente en raison de la preuve de travail ou d'autres mécanismes de consensus	Rapide, car moins de nœuds de validation sont nécessaires	Variable, généralement plus rapide qu'une blockchain publique mais plus lente qu'une blockchain privée
Exemples	Bitcoin, Ethereum	Hyperledger Fabric, Corda	R3 Corda, Hyperledger Fabric (avec plusieurs membres)

- Partage sécurisé des données médicales.
- Marchés NFT.
- Suivi des redevances musicales.
- Paiements transfrontaliers.
- Systèmes d'exploitation IoT en temps réel.
- Sécurité de l'identité personnelle.
- Système de suivi anti-blanchiment d'argent.
- Suivi de la chaîne d'approvisionnement et de la logistique.
- Mécanisme de vote.
- Création de contenu original.
- Échange de crypto-monnaie.
- Plateforme de traitement immobilier.

La blockchain peut enregistrer tout ce qui doit être horodaté et authentifié par n'importe qui, pas seulement les transactions financières, comme le but de l'invention de Bitcoin. Les gens utilisent déjà la technologie blockchain pour sécuriser les votes lors des élections politiques, suivre les transactions peer-to-peer et certifier les médias numériques en utilisant le concept de NFT (Non-Fungible Tokens le sujet brûlant de nos jours) comme les textes, la musique, l'art,

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

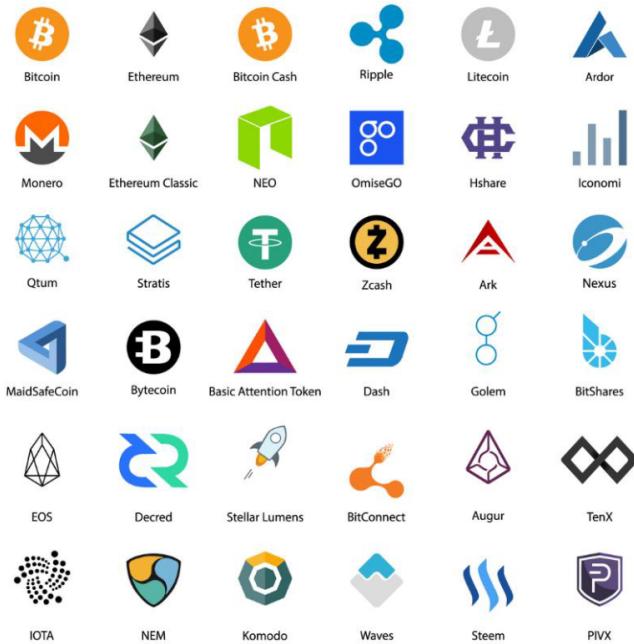


FIGURE 2.16 – Les crypto-monnaies les plus connues

les jeux vidéo et des films. Jusqu'à présent, il existe de nombreuses Blockchains différentes, dont certaines sont développées par des sociétés géantes connues telles qu'IBM, Oracle[2.17] .

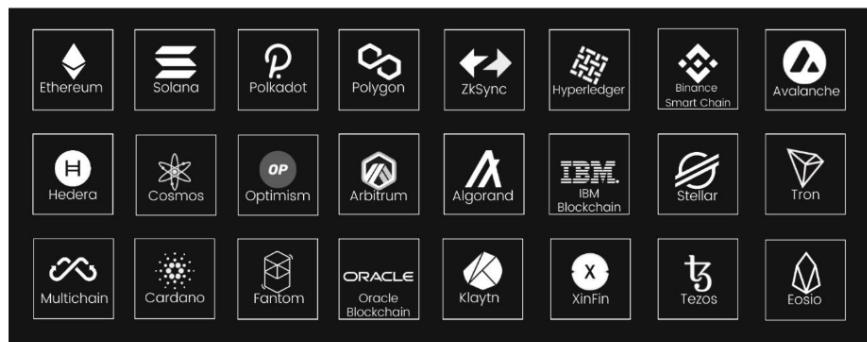


FIGURE 2.17 – Les blockchains les plus connues

Mais, le premier et le plus populaire est Ethereum, c'est pourquoi nous l'avons choisi comme notre infrastructure Blockchain pour construire notre DApp dessus.

2.2.4 La Blockchain Ethereum :

1- Définition :

La blockchain Ethereum, souvent surnommée "ordinateur mondial", est une plateforme de programmation de dApps qui fonctionne comme une machine d'état déterministe. Accessible à l'échelle mondiale et entretenue par un réseau décentralisé de nœuds pair à pair, elle permet des changements d'état régis par des règles de consensus. Cette machine d'état est accessible et ouverte à tous les utilisateurs du réseau, et n'appartient donc à aucune entité unique, mais plutôt est partagée collectivement par l'ensemble des participants.

2- Contrats intelligents :

L'essor de la technologie Blockchain a favorisé le développement de concepts tels que les « **Smart Contracts** ». Ces contrats auto-exécutables [2.18] intègrent les termes d'un accord directement dans des lignes de code. Déployés sur des plateformes Blockchain via des langages de programmation spécifiques, ils automatisent l'exécution des contrats grâce à des protocoles informatiques et des interfaces utilisateur. Devenus populaires grâce à la Blockchain, ils simplifient leur utilisation par rapport aux technologies disponibles lors de leur invention il y a 20 ans, et pourraient potentiellement remplacer des intermédiaires tels que les avocats et les banques.

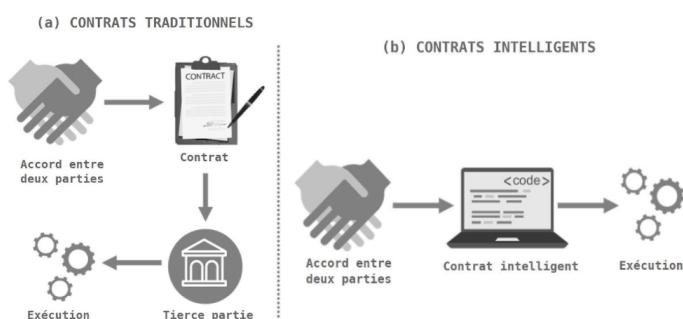


FIGURE 2.18 – Différence entre "smart contract" et "contrat traditionnel"

a) Les principales caractéristiques des contrats intelligents :

Parmi les **avantages** que les smarts contracts possèdent :

- **Autonomie** : Les contrats intelligents éliminent le besoin d'intermédiaires, tels que des avocats ou des courtiers, car ils appliquent automatiquement les termes de l'accord sans

2.2. PRÉSENTATION DES CONCEPTS ET DES NOTIONS THÉORIQUES UTILISÉS

faire appel à des tiers.

- **Transparence** : Le code et l'exécution d'un contrat intelligent sont enregistrés sur la Blockchain, la rendant transparente et visible pour tous les participants du réseau.
- **Sécurité** : Les contrats intelligents sont construits sur la Blockchain, qui fournit des fonctionnalités de sécurité inhérentes telles que l'immuabilité, des algorithmes cryptographiques et des mécanismes de consensus qui garantissent l'intégrité et la sécurité du contrat.
- **Efficacité** : En automatisant l'exécution des contrats, les contrats intelligents réduisent le besoin de processus manuels, de paperasse et de rapprochement, conduisant à des transactions plus rapides et plus efficaces.

D'autre part, il y a aussi des **inconvénients** liés à l'utilisation des contrats intelligents :

- **La possibilité de failles** : Une réalité inhérente à tout programme informatique. Plus les contrats intelligents sont complexes, plus le risque de vulnérabilités augmente.
- **L'absence totale de régulation** : Ce qui soulève des questions tant sur le plan moral que juridique. Cette carence en encadrement légal peut engendrer des dilemmes éthiques et des litiges juridiques.

3- Ethereum Virtual Machine (EVM) :

Les contrats intelligents sur Ethereum Blockchain s'exécutent sur la machine virtuelle Ethereum ou EVM, qui est présente sur chaque nœud du réseau. Le but d'EVM est d'exécuter la logique définie dans les contrats intelligents et traite les changements d'état qui se produisent sur cette machine d'état accessible globalement (La Blockchain Ethereum). L'EVM ne comprend pas les langages de haut niveau comme Solidity et Vyper, qui sont utilisés pour écrire des contrats intelligents. Au lieu de cela, nous devons compiler le langage de haut niveau en code d'octet, que l'EVM peut ensuite exécuter.

4- Qu'est-ce qu'une application décentralisée (DApps) ?

Une DApp est une application logicielle qui fonctionne partiellement ou totalement grâce à un ensemble d'acteurs différents et à des smart contracts exécutés sur une ou plusieurs blockchains. Les DApp sont souvent open source et utilisent des réseaux décentralisés, offrant une

2.3. IA ET BLOCKCHAIN :

plus grande autonomie et un meilleur contrôle des utilisateurs, ainsi qu'une sécurité et une confiance accrues.

Une application décentralisée est typiquement composée de plusieurs éléments, dont des contrats autonomes (smart contracts) opérant sur une blockchain, une interface utilisateur transparente, un modèle de stockage de données distribué, un protocole de communication pair à pair, ainsi qu'un système décentralisé de résolution de noms.

2.3 IA et Blockchain :

2.3.1 Définition Générale de l'Intelligence Artificielle (IA) :

L'intelligence artificielle (IA) désigne un ensemble de techniques permettant aux machines de simuler des comportements intelligents, tels que l'apprentissage, le raisonnement, la prise de décision et la résolution de problèmes. Ces capacités sont souvent rendues possibles grâce à des algorithmes d'apprentissage automatique, d'apprentissage profond et des systèmes experts, permettant aux machines de traiter des données complexes et d'exécuter des tâches qui nécessitent normalement une intervention humaine.

2.3.2 Utilité de l'IA lorsqu'elle est intégrée à la Blockchain :

L'intégration de l'IA avec la blockchain ouvre de nouvelles perspectives en combinant les capacités analytiques et adaptatives de l'IA avec les propriétés sécurisées, transparentes et immuables de la blockchain.

Analyse et traitement automatisés des données : L'IA peut traiter rapidement les données enregistrées dans une blockchain, détecter des modèles, identifier des anomalies ou prévoir des tendances, renforçant ainsi la prise de décision en temps réel.

Traçabilité et sécurité des données :

La blockchain garantit que les données traitées par l'IA restent sécurisées et traçables, offrant une fiabilité renforcée dans des domaines sensibles tels que la santé, la finance ou la gestion environnementale.

Optimisation des coûts et des performances :

L'IA peut optimiser l'utilisation des ressources blockchain, comme la gestion des contrats in-

2.4. CONCLUSION

telligents ou l'optimisation énergétique des réseaux.

Applications spécifiques : Dans des cas comme la surveillance environnementale, l'IA détecte rapidement des événements critiques (par exemple, les déversements d'hydrocarbures), tandis que la blockchain conserve un enregistrement immuable des incidents, garantissant la transparence et facilitant la coordination entre les parties prenantes.

Cette combinaison technologique offre une synergie unique, permettant de relever des défis complexes avec des solutions robustes et efficaces.

Alors si nous cloturons pourquoi l'IA est essentielle dans notre sujet ?

L'intelligence artificielle (IA) est cruciale pour relever les défis complexes liés à la détection et à la gestion des déversements d'hydrocarbures en milieu maritime. Grâce à sa capacité à automatiser des tâches complexes et à analyser des volumes massifs de données, l'IA offre une détection précise des déversements, en évitant les confusions avec d'autres types de pollution, comme les algues ou les débris marins. Cependant, malgré ces avantages, certaines limitations persistent : une faible précision dans des conditions météorologiques extrêmes (nuages, tempêtes), une couverture géographique limitée, et des défis techniques dans l'identification spécifique des hydrocarbures. Notre solution propose de dépasser ces limites grâce à une combinaison innovante de technologies. Une application multiplateforme intuitive, développée sous Flutter, garantit des notifications instantanées pour une intervention rapide. Les données critiques sont sécurisées au sein d'une blockchain, offrant une traçabilité et une intégrité sans faille. En intégrant efficacement l'IA et la blockchain, cette solution réduit les coûts tout en augmentant l'efficacité et l'accessibilité, posant ainsi les bases d'une gestion écologique et proactive des déversements en mer.

2.4 Conclusion

Ce chapitre a fourni l'occasion de présenter les concepts et les notions clés nécessaires à notre étude théorique sur les démarches de développement et de conception de notre solution, qui sera présentée au chapitre 3. En explorant les concepts clés de la blockchain et son fonctionnement, ainsi que les aspects fondamentaux des applications décentralisées (Dapp), des contrats intelligents et du système de gestion des données distribué, nous avons acquis une compréhension globale. Cette compréhension générale est cruciale pour la suite de notre travail.

Chapitre 3

Etude Conceptuelle

3.1 Introduction

Après avoir défini les besoins et les objectifs dans le chapitre précédent, nous nous concentrons sur l'aspect architectural de notre projet. Cette phase vise à concevoir les schémas généraux qui permettent la modélisation et la description d'une manière non ambiguë du fonctionnement souhaité de l'application. Ce chapitre commence par le choix de la méthodologie de conception. Ensuite, nous identifions les acteurs et les diagrammes des cas d'utilisation, avant de présenter le diagramme de séquence et le diagramme de classe.

3.2 Démarche de conception et méthodologie de travail

Après l'étude des processus existants, leurs critiques et la comparaison entre les solutions similaires qui existent sur le marché, nous allons présenter dans cette partie la méthodologie utilisée. Nous présenterons la démarche du projet, la description du langage de modélisation, l'outil de conception utilisé et le choix de la méthodologie qui sera adaptée tout au long de ce rapport. Nous pouvons prouver cela à travers les travaux de Smith et al.(2020) et Johnson (2019), qui ont démontré l'efficacité de certaines méthodologies pour les projets de séminaires. De plus, des études telles que celles de Garcia (2021) et Patel et al. (2018) fourniront des justifications supplémentaires pour nos choix. Enfin, nous expliquerons pourquoi la méthodologie choisie est appropriée pour notre projet, en nous appuyant sur des références telles que Lee et Smith (2017) et Brown (2022).

3.2.1 Langage de modélisation UML

La phase de préparation des données est la phase primordiale et la plus importante. Aussi, elle nécessite des ressources spécialisées pour notre projet. À partir des données nous devons créer un modèle qui soit d'une ampleur appropriée.

a - Définition

UML (Unified Modeling Language) est un langage formel et normalisé en termes de modélisation objet. Son indépendance par rapport aux langages de programmation, aux domaines de l'application et aux processus, son caractère polyvalent et sa souplesse ont fait de

3.2. DÉMARCHE DE CONCEPTION ET MÉTHODOLOGIE DE TRAVAIL

lui un langage universel. En plus, UML est essentiellement un support de communication, qui facilite la représentation et la compréhension de solution objet. Sa notation graphique permet d'exprimer visuellement cette solution, ce qui simplifie la comparaison et l'évaluation des solutions. L'aspect de sa notation limite l'ambigüité et les incompréhensions. Ce langage fournit un moyen astucieux permettant de représenter diverses projections d'une même. Une vue est constituée d'un ou de plusieurs diagrammes où nous distinguons 2 types de vue de modélisation UML :

- **Vue statique** : permettant de représenter le système physiquement : Le Diagramme de Cas d'Utilisation décrit les interactions entre les acteurs et le système, mettant en évidence les fonctionnalités offertes par le système du point de vue des utilisateurs, "Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development" par Craig Larman, publié en 2005. Le Diagramme de Classes représente les classes du système, ainsi que leurs relations et leurs attributs, "The Unified Modeling Language Reference Manual" par James Rumbaugh, Ivar Jacobson et Grady Booch, publié en 2004.

Vue dynamique : qui met en lumière le comportement du système : Le Diagramme de Séquence illustre la séquence d'interactions entre les objets du système dans un scénario particulier, montrant l'ordre chronologique des messages échangés, "Enterprise Patterns : Concurrency, Transactions, and Serialization" par Martin Fowler, publié en 2004.

b - Choix du langage

Nous avons choisi le langage **UML** car il est un support de modélisation performant et normalisé qui permet de concevoir et de déployer une architecture logicielle développée dans un langage objet (Java, C++, VB.net). Il permet ainsi : Un gain de précision et de temps. UML cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Il peut être intégré à n'importe quel processus de développement logiciel. Sa polyvalence et sa souplesse font de lui un langage universel.

3.2.2 Les méthodes de conception

Les "méthodologies" disent qu'une méthode, pour être opérationnelle, doit avoir 3 composantes : une démarche (définissant les étapes, les phases et les tâches de mise en œuvre),

3.2. DÉMARCHE DE CONCEPTION ET MÉTHODOLOGIE DE TRAVAIL

des formalismes (incluant les modélisations et les techniques de transformation). À son époque, Merise s'est efforcée de proposer un ensemble "cohérent" de ces trois composantes. Cependant, avec le temps, certaines de ces composantes ont vieilli et ont nécessité une actualisation (notamment la démarche), tandis que d'autres sont toujours pertinentes (telles que les modélisations). Certains éléments ont été maintenus, d'autres ont été retirés ou modifiés en raison de leur obsolescence ou de leur inefficacité dans les pratiques actuelles.

A - Les méthodes agiles :

A.1 -Définition :

Les méthodes agiles sont une manière de réduire le cycle de développement des projets informatiques de répondre plus rapidement aux évolutions des demandes de l'utilisateur final versatile. Les projets informatiques agiles sont gérés de manière adaptative, incrémentale et itérative.

- **La méthode de développement rapide d'applications**, dite méthode (RAD1) est la première méthode de développement de logiciels où le cycle de développement est en rupture fondamentale par rapport à, celui des méthodes antérieures dites "en cascade". Ce nouveau cycle qualifié d'itératif, d'incrémental et d'adaptatif, se retrouvera dans toutes les méthodes dites "agiles".

-Dynamics Systems developpment methods (DSDM) est une méthode de gestion de projet de la catégorie des méthodes agiles, "Abrahamsson et al., 2017, page 123".

B - Les méthodes unifiées

-La méthode PU(Processsus Unifié) : Les méthodes unifiées, telles que le Processus Unifié (PU), sont des approches de développement pour les logiciels orientés objets, "Jacobson et al., 1999, page 45". Le PU se distingue des méthodes séquentielles comme Merise (ou SADT) par sa nature générique, itérative et incrémentale, "Smith et al., 2007, page 78".

-La méthode 2TUP : La méthode 2TUP, proposée par Jones et al. 2005, page211, se distingue par son cycle de développement en Y, qui sépare les aspects techniques des aspects fonctionnels. Ce processus démarre par une étude préliminaire "Thayer et al., 2000, page 95",

3.2. DÉMARCHE DE CONCEPTION ET MÉTHODOLOGIE DE TRAVAIL

visant à identifier les acteurs impliqués, les messages échangés, à produire le cahier des charges et à modéliser le contexte.

3.2.3 La méthode utilisée

Il existe plusieurs méthodes pour partir de l'expression d'un besoin et aboutir à la réalisation d'un système. Notre projet se base sur un processus de développement bien précis qui part de la spécification des besoins fonctionnels du système pour aboutir à la conception et le codage final. Ce processus est basé de nature sur le Processus Unifié, de ce principe, nous avons décidé d'utiliser la méthode 2TUP, puisqu'elle offre une méthodologie qui permet de décortiquer un projet en plusieurs parties fortement réutilisables.

La méthode 2TUP dissocie initialement les contraintes techniques des contraintes fonctionnelles avant de les regrouper dans la phase de réalisation. La figure 3.1 détaille les différentes phases du développement :



FIGURE 3.1 – Processus de développement en Y

* Le processus s'articule ensuite autour de trois phases essentielles :

La branche fonctionnelle : Elle se focalise sur la compréhension du métier de l'entreprise et la capture des besoins fonctionnels. Elle comporte les étapes suivantes :

-Capture des besoins fonctionnels : Elle établit un modèle des besoins centré sur les gestionnaires, prévenant ainsi le risque de développer un système inadapté..

-Analyse : Elle examine minutieusement les spécifications fonctionnelles pour définir clairement

3.2. DÉMARCHE DE CONCEPTION ET MÉTHODOLOGIE DE TRAVAIL

ce que le système réalisera en termes de métier..

• **Branche technique :** Elle capitalise sur le savoir-faire technique et les fonctionnalités techniques nécessaires. Cette branche comporte les étapes suivantes :

- Capture des besoins techniques : Elle identifie toutes les contraintes de conception du système, incluant les spécifications des outils et la structure du matériel, ainsi que les contraintes d'intégration avec les systèmes existants.
- Conception générique : Elle établit les composants nécessaires à l'architecture technique, visant à uniformiser et réutiliser les mécanismes pour l'ensemble du système

• **Phase de réalisation :** Elle combine les branches fonctionnelles et techniques pour une conception applicative adaptée aux besoins. Cette branche comporte les phases suivantes :

- La conception préliminaire : Elle intègre les fonctions métier et applicatives dans l'architecture technique définie précédemment.
- La conception détaillée : Elle détaille la réalisation de chaque composant.
- Le codage et test : Elle produit les composants et effectue des tests unitaires continus.
- Étape de recette : Elle valide les fonctionnalités du système développé

3.2.4 Conception de la technologie de sécurité

La conception de notre projet réalisé à été élaborée en utilisant **le langage UML** en suivant la démarche comme suit :

- L'élaboration de diagramme de cas d'utilisation suit de la spécification fonctionnelle de l'application.
- Dressage des diagrammes de séquences pour mettre en évidence les interactions entre les différents objets du système.
- Elaboration des diagrammes des classes.

3.2. DÉMARCHE DE CONCEPTION ET MÉTHODOLOGIE DE TRAVAIL

TABLE 3.1 – Utilisation d’UML dans la conception du projet

Étape de conception	Utilisation d’UML	Justification
Diagramme de cas d’utilisation	Oui	UML standardise la représentation des interactions entre les acteurs et le système, facilitant ainsi la capture et la compréhension des besoins fonctionnels.
Diagrammes de séquences	Oui	UML fournit des diagrammes de séquences pour visualiser les interactions entre les objets du système, clarifiant le déroulement des actions et des messages échangés.
Diagrammes de classes	Oui	Les diagrammes de classes UML permettent de concevoir l’architecture du système en représentant sa structure statique, favorisant ainsi la planification et l’organisation du code.

À partir de ce tableau, nous pouvons conclure que UML offre une plus grande polyvalence et une notation plus riche pour modéliser les systèmes logiciels par rapport à Merise, qui est davantage axé sur les bases de données et les systèmes d’information. Cependant, Merise peut être plus limité pour représenter les interactions entre objets dans un système orienté objet.

3.2. DÉMARCHE DE CONCEPTION ET MÉTHODOLOGIE DE TRAVAIL

A- Diagramme des cas d'utilisation(DCU) :

Un cas d'utilisation est une représentation discrète d'interaction entre un utilisateur (qu'il soit humain ou machine) et un système logiciel. Il dépeint un scénario spécifique d'utilisation du système. Ainsi, dans ce diagramme les utilisateurs sont appelés acteurs (actors), engagent des actions avec le système afin d'accomplir une tâche particulière ou d'atteindre un objectif déterminé. Ces interactions sont essentielles pour comprendre le comportement fonctionnel du système et sont souvent illustrées dans les diagrammes UML, qui sont utiles pour des présentations auprès de la direction ou des parties prenantes du projet. Toutefois, lors du processus de développement, les cas d'utilisation sont privilégiés pour une représentation plus précise et détaillée des besoins et des fonctionnalités du système.

A.1) Diagramme de cas d'utilisation globale :

L'application est une plateforme de gestion de pollution marin basée sur la Blockchain et IA, offrant un stockage sécurisé et transparent des données es déversements d'hydrocarbures 3.2. Pour les marins et les utilisateurs, elle permet dela Détection précise des déversements d'hydrocarbures (oil spills) grâce à l'IA, évitant les confusions avec d'autres types de pollution.Une accessibilité simplifiée via une interface multiplateforme intuitive et une intégration optimisée de l'IA et de la blockchain pour réduire les coûts tout en améliorant l'efficacité pour les utilisateurs.

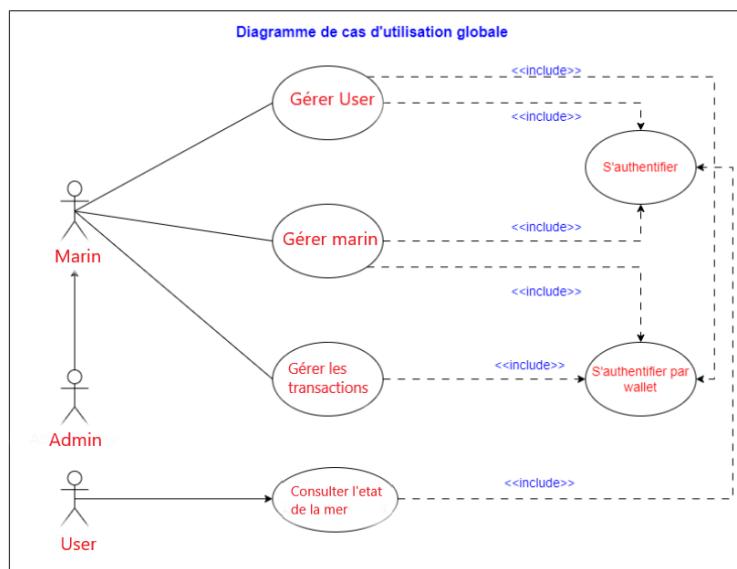


FIGURE 3.2 – Diagramme de cas d'utilisation global

3.3. MÉTHODOLOGIE DE PARTIE IA DE TRAVAIL :

Le transfert d'une donnée anormal entre les marins implique une authentification numérique, une saisie et une vérification des données, l'exécution d'un contrat intelligent, le paiement des frais de transaction, la confirmation par l'utilisateur, la mise à jour des informations du patient et une nouvelle connexion au système.

3.3 Méthodologie de partie IA de travail :

Comme pour tout projet de développement, il est essentiel de commencer par choisir une méthodologie appropriée et de documenter ses spécificités. Dans cette section, nous présenterons la méthodologie sélectionnée ainsi que les raisons de nos choix.

3.3.1 Méthodologie CRISP-DM :

Pour ce projet visant à améliorer l'accessibilité web pour les personnes atteintes de déficience visuelle, nous avons choisi de suivre la **méthodologie CRISP-DM (Cross-Industry Standard Process for Data Mining)**.

CRISP-DM est un cadre méthodologique bien établi et largement utilisé qui propose une approche structurée et étape par étape pour les projets centrés sur les données, ce qui en fait un choix idéal pour le développement de notre solution de synthèse vocale destinée à améliorer l'accès aux contenus web. La méthodologie se compose de six phases distinctes garantissant une progression logique depuis la définition du problème jusqu'à la mise en production :

1. Compréhension des besoins métier : Cette phase consiste à acquérir une compréhension approfondie des objectifs du projet du point de vue des utilisateurs et des besoins métier. Dans notre cas, l'objectif est d'améliorer l'accessibilité des contenus web pour les personnes atteintes de déficience visuelle en convertissant le texte en audio. Nous cherchons à garantir que la solution proposée réponde aux besoins des utilisateurs aveugles ou malvoyants afin de rendre la navigation web plus inclusive.

2. Compréhension des données : Cette phase se concentre sur la collecte et l'analyse des données nécessaires au projet. Nous identifions, collectons et explorons divers types de contenus web (éléments textuels, en-têtes, articles, etc.) tout en étudiant les défis d'accessibilité auxquels font face les utilisateurs malvoyants. Cela nous permet de déterminer les types de données à

3.3. MÉTHODOLOGIE DE PARTIE IA DE TRAVAIL :

traiter et à convertir en audio.

3. Préparation des données : Dans cette étape, les données brutes sont nettoyées, transformées et organisées afin de les rendre adaptées aux analyses et au développement du modèle. Cela inclut la préparation des éléments des pages web, en s'assurant que le texte est correctement extrait et formaté pour une conversion efficace en synthèse vocale. Les données inutiles ou bruitées sont filtrées afin d'optimiser les performances du modèle.

4. Modélisation : Cette phase consiste à créer des modèles capables de transformer les contenus web en audio. Nous utilisons des techniques d'apprentissage automatique et de traitement du langage naturel (NLP) pour développer des algorithmes capables de lire et de synthétiser le texte avec précision. L'accent est mis sur la génération d'une voix claire et naturelle à partir de divers types de textes.

5. Évaluation : Une fois le modèle développé, il est évalué pour mesurer sa précision et son efficacité. Cela implique de tester la solution en fonction de métriques prédéfinies telles que la clarté vocale, la détection de la langue et la satisfaction des utilisateurs. Les retours des utilisateurs malvoyants sont essentiels pour déterminer si la solution répond à leurs besoins spécifiques et attentes.

6. Mise en production : Dans la phase finale, le modèle fonctionnel est intégré dans des applications concrètes. Pour notre projet, cela signifie intégrer la solution de synthèse vocale dans les environnements web, permettant ainsi aux utilisateurs malvoyants d'écouter le contenu des pages web lors de leur navigation. La mise en production garantit que la solution est prête à être utilisée et accessible à la cible visée.

3.3.2 Justification du choix de CRISP-DM :

Nous avons choisi la méthodologie CRISP-DM pour ce projet car elle offre une approche flexible, itérative et centrée sur l'utilisateur. Ses phases structurées nous guident à la fois dans le développement technique et dans la prise en compte des besoins des utilisateurs, ce qui la rend particulièrement adaptée aux défis liés à l'accessibilité. Cette méthodologie garantit également que chaque phase s'appuie sur la précédente, favorisant un cycle d'amélioration continue essentiel pour concevoir des solutions fiables et inclusives dans les projets d'accessibilité.

3.4. CONCLUSION

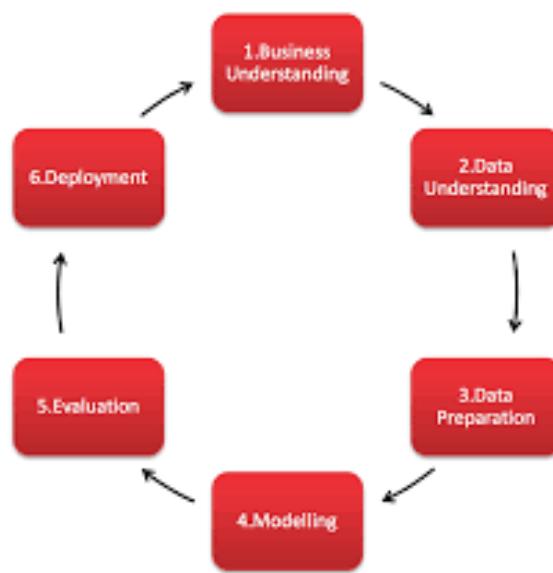


FIGURE 3.3 – CRISP-DM

3.4 Conclusion

A travers ce chapitre, nous avons présenté la conception proposée pour l'application. Nous avons cité, au premier lieu, la conception globale. Par la suite, nous avons détaillé la conception à travers les diagrammes UML qui nous ont permis de dégager la vue statique ainsi que la vue dynamique de l'application. Afin de mieux concrétiser cette conception, nous présenterons dans le chapitre suivant la description et les différentes interfaces réalisées par notre application.

Chapitre 4

Implémentation de la solution proposée

4.1 Introduction

Cette partie contient le dernier volet de ce rapport. Elle a pour objectif d'exposer le travail achevé. Dans un premier temps, nous présentons l'environnement matériel, logiciel et les différents outils de développement utilisés. Dans un second temps, nous illustrons la réalisation de notre travail par des captures d'écran des interfaces les plus importantes de notre application.

4.2 Outils et Environnements de travail

Dans cette section, nous présentons les différents outils matériels misent à la disposition du présent projet ainsi que l'environnement logiciel que nous avons utilisé pour le développement notre application en justifiant les choix technologiques.

4.2.1 Environnement logiciel

Pour élaborer ce travail nous avons fait recours à un certain nombre de logiciels et languages de programmation :

4.2.2 Languages utilisés

A - HTML5 :



Un langage universel qui utilise de nombreuses balises. Aujourd'hui, le langage HTML en est à sa cinquième version, le HTML 5 avec de nouveau éléments , attributs et comportements. Il fait partie des trois principales inventions du World Wide Web avec le protocole HTTP ou les URL. Le langage HTML a apporté une véritable révolution sur Internet qui permet d'accéder aux sites web plus variés et puissants.

4.2. OUTILS ET ENVIRONNEMENTS DE TRAVAIL

B - CSS3 :



Le CSS pour Cascading Style Sheets, est un langage informatique utilisé sur Internet pour la mise en forme de fichiers et de pages HTML. On le traduit en français par feuilles de style en cascade.

Apparu dans les années 1990, le CSS se présente comme une alternative à la mise en forme via des balises, notamment HTML.

C - Javascript :



Est un langage de programmation intégré au navigateur web. il constitue l'un des meilleurs moyens de rendre un site web interactif. En effet, Javascript est le seul langage qui fonctionne directement au sein de tout navigateur. il dispose d'un accès aux images ,formulaires et fenêtres qui constituent une page web .

D - Keras :



Keras est une bibliothèque open-source de haut niveau pour l'apprentissage profond, conçue pour construire et entraîner des modèles de réseaux de neurones de manière simple et rapide. Elle fonctionne comme une interface pour TensorFlow et prend en charge des tâches telles que la classification, la régression et le traitement d'images, de texte et de séries temporelles..

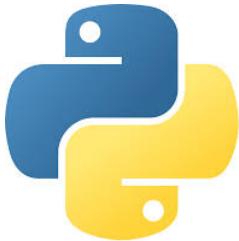
D - Tensorflow :



TensorFlow

TensorFlow est une bibliothèque open-source développée par Google pour le calcul numérique et l'apprentissage automatique. Elle est largement utilisée pour construire, entraîner et déployer des modèles d'intelligence artificielle, notamment des réseaux de neurones profonds. TensorFlow prend en charge une grande variété de plateformes, allant des serveurs aux appareils mobiles, et permet des calculs distribués pour traiter de grandes quantités de données.

D - Python :



Python est un langage de programmation interprété, polyvalent et facile à apprendre, largement utilisé dans des domaines tels que le développement web, l'analyse de données, l'intelligence artificielle, l'automatisation, et plus encore. Il se distingue par sa syntaxe claire et concise, qui favorise la lisibilité du code. Python dispose d'une vaste collection de bibliothèques standard et tierces (comme NumPy, Pandas, TensorFlow, Flask, Django), qui accélèrent le développement de solutions complexes. Grâce à sa communauté dynamique et à son support multi-plateformes, Python est devenu un outil incontournable pour les développeurs, les scientifiques des données et les ingénieurs.

D - Flask API :

⚡ FastAPI

Flask est un micro-framework en Python conçu pour développer des applications web légères et des APIs RESTful. Il est minimaliste, flexible et modulaire, ce qui permet aux développeurs de personnaliser leur application selon leurs besoins sans être contraints par une structure rigide. Flask repose sur le moteur de templates Jinja2 et le serveur WSGI Werkzeug. Il prend en charge des fonctionnalités telles que la gestion des routes, le traitement des requêtes HTTP (GET, POST, PUT, DELETE), et la sérialisation JSON pour échanger des données entre clients et serveurs. Grâce à sa simplicité et à sa compatibilité avec de nombreuses bibliothèques Python, Flask est idéal pour développer des prototypes rapides, des APIs pour des modèles d'apprentissage automatique, ou des projets web complets.

4.2.3 Les applications utilisées

A - Visual Studio



Est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'auto complétion, la coloration syntaxique, le débogage (exécution pas à pas), et les commandes git.

B - MySQL :



MySQL® MySQL est le serveur de base de données le plus utilisé dans le monde. Son architecture logicielle le rend extrêmement rapide et facile à personnaliser. Les principaux avantages de MySQL sont sa rapidité, sa robustesse et sa facilité d'utilisation et d'administration. Un autre avantage majeur de MySQL est sa documentation très complète et bien construite.

4.2. OUTILS ET ENVIRONNEMENTS DE TRAVAIL

D - DRAW.IO :



draw.io

Draw.io est une application en ligne gratuite de création de diagrammes et de schémas. Elle offre une interface intuitive et conviviale, permettant aux utilisateurs de concevoir facilement une grande variété de diagrammes, tels que des organigrammes, des diagrammes de flux, des diagrammes UML, des diagrammes réseau.

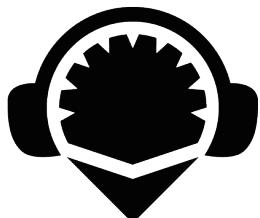
E - PHPMYADMIN :



phpMyAdmin

phpMyAdmin est un outil logiciel gratuit écrit en PHP , destiné à gérer l'administration de MySQL sur le Web. phpMyAdmin prend en charge un large éventail d'opérations sur MySQL et MariaDB. Les opérations fréquemment utilisées (gestion des bases de données, des tables, des colonnes, des relations, des index, des utilisateurs, des autorisations, etc.) peuvent être effectuées via l'interface utilisateur, tout en ayant la possibilité d'exécuter directement n'importe quelle instruction SQL.

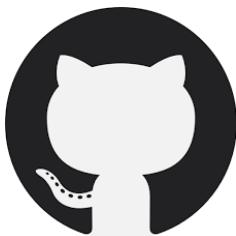
F - REMIXIDE :



Remix Project est un ensemble d'outils riche comprenant Remix IDE, un outil complet de développement de contrats intelligents. Le projet Remix comprend également le moteur de plug-in Remix et les bibliothèques Remix, qui sont des outils de bas niveau destinés à une utilisation plus large.

4.2. OUTILS ET ENVIRONNEMENTS DE TRAVAIL

G - Github :



Est une plateforme open source de gestion de versions et de collaboration destinée aux développeurs de logiciels. Livrée en tant que logiciel à la demande (SaaS, Software as a Service). Elle repose sur Git, un système de gestion de code open source créé par Linus Torvalds dans le but d'accélérer le développement logiciel.

H - MetaMask :



MetaMask est un portefeuille logiciel de crypto-monnaie utilisé pour interagir avec la blockchain Ethereum. Il permet aux utilisateurs d'accéder à leur portefeuille Ethereum via une extension de navigateur ou une application mobile, qui peut ensuite être utilisée pour interagir avec des applications décentralisées. MetaMask est développé par ConsenSys Software Inc, une société de logiciels blockchain développant des outils dans l'univers d'Ethereum.

I - GANACHE :



Nous avons donc utilisé GPB (Ganache Personal Blockchain), qui est une blockchain de développement local qui peut être utilisée pour imiter le comportement d'une blockchain. Il nous permettra de déployer des contrats intelligents, de développer des applications et d'exécuter des tests. Il est disponible sur Windows, Mac et Linux en tant qu'application de bureau et outil de ligne de commande. Dans un premier temps, nous mettons en place une blockchain personnelle et privée pour développer l'application en local.

4.2. OUTILS ET ENVIRONNEMENTS DE TRAVAIL

K - ANDROIDSTUDIO :



Android Studio se présente comme un environnement de développement intégré (IDE) indispensable pour concevoir des applications Android. Il met à disposition des développeurs une interface conviviale et des outils performants pour faciliter le processus de création d'applications mobiles. L'IDE vous accompagne dans chaque étape de votre projet. Il vous permet d'écrire du code en Java, Kotlin ou d'autres langages compatibles avec Android, tout en vous offrant des outils de conception visuelle intuitifs pour façonner des interfaces utilisateur attrayantes.

4.2.4 Les frameworks utilisées pour la programmation

A - Node.js :



Sert à faire du java script cotés serveur (serveur side), et peut être utilisé dans des applications de bases de données, la plus populaire étant MySQL. Ce n'est ni un framework ni un serveur. Node.js est souvent confondu avec ce dernier, car c'est une base : créer des applications en temps réel, ou le serveur a la possibilité de transmettre de l'information au client.

B - Flutter :



Flutter est un framework de développement d'applications multiplateformes révolutionnaire, conçu par Google pour permettre aux développeurs de créer des applications natives à partir d'une seule base de code. Avec son langage de programmation Dart, Flutter offre une expérience de développement fluide et efficace, grâce à ses widgets personnalisables et sa fonctionnalité de Hot Reload, permettant une itération rapide et un débogage en temps réel.

4.3 Présentation de l'enchainement de l'application

Lors de la mise en œuvre de notre application, nous avons développé deux versions, une version web et une version mobile :

4.3.1 Déploiement du smart contract sur Ganache

Dans un premier temps, nous mettons en place une blockchain personnelle pour développer l'application en local, puis nous avons déployé le smart contract sur ce réseau de blockchain locale proposé par Ganache.

A- Connecter avec MetaMask

Nous devons télécharger l'extension Meta Mask et l'ajouter dans le navigateur afin de se connecter au réseau blockchain :

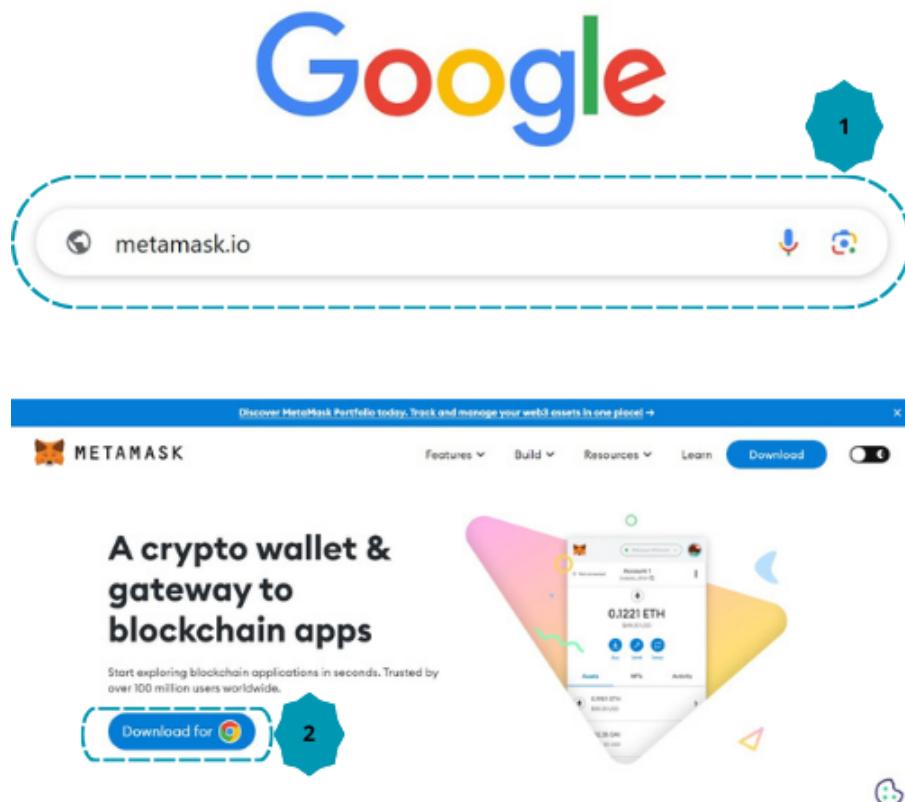


FIGURE 4.1 – télécharger MetaMask

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

- Lorsque la boutique en ligne s'ouvre,nous cliquons sur le bouton ”**Obtenir**” ou ”**Ajouter à Chrome**” du navigateur Chrome.

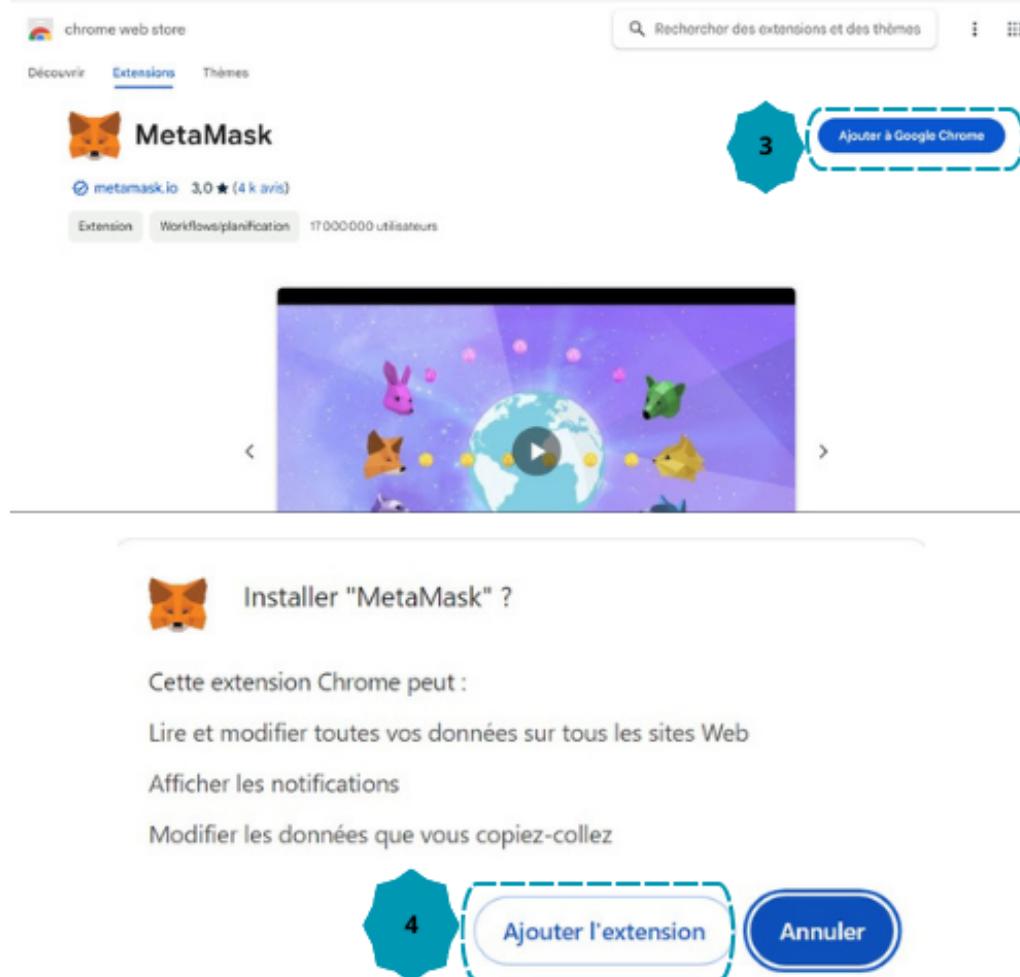


FIGURE 4.2 – Cliquer sur Ajouter Chrome

Lorsque nous cliquons sur le bouton ”**Ajouter une extension**”, l’extension commencera à se télécharger, et après l’avoir téléchargée, l’extension sera installée dans notre navigateur.

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

Créer un portefeuille sur Meta Mask

Passons présentement à l'aspect le plus important de la création de portefeuille Meta Mask.

Création d'un nouveau compte

MetaMask est un portefeuille qui permet d'accéder à notre application Web3 et d'échanger des cryptomonnaies. Nous pouvons créer une nouvelle compte pour se connecter au réseau blockchain. Nous redirigeons vers la page d'accueil de Meta Mask pour créer le compte Portefeuille. Puis , nous cliquons sur le bouton « **créer un nouveau compte** ».

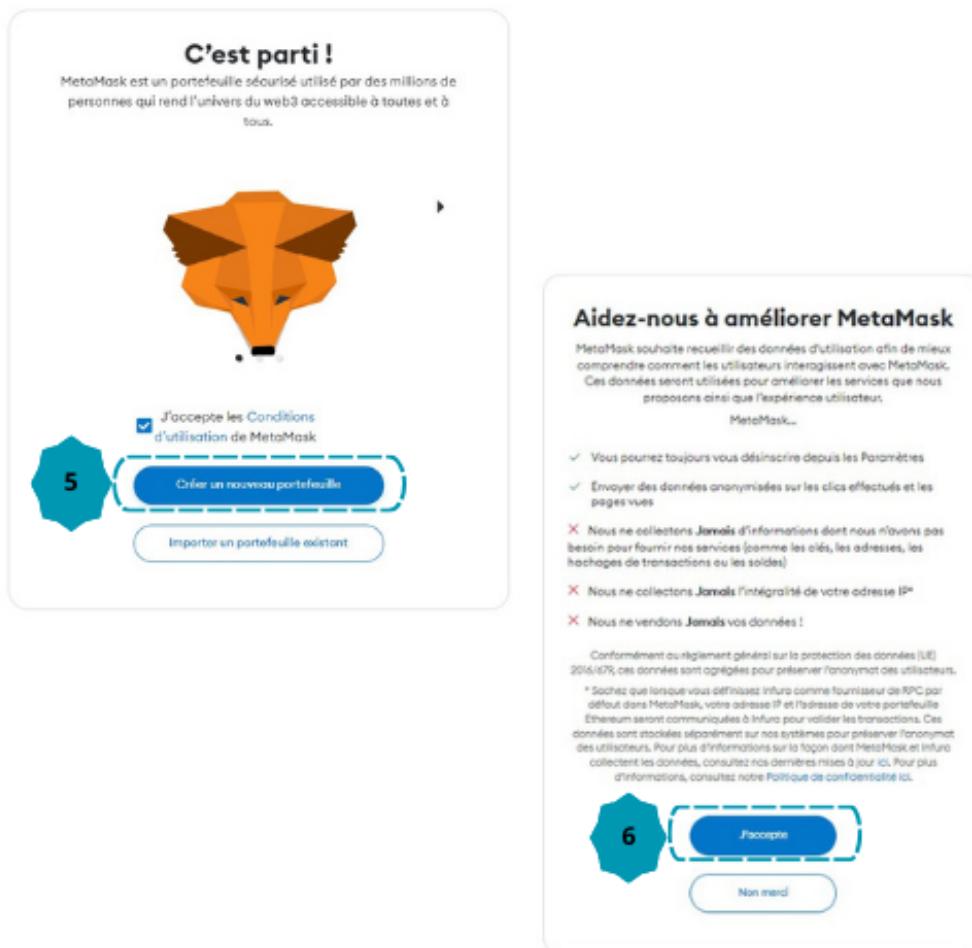


FIGURE 4.3 – Crédit : Création d'un nouveau compte

4.3. PRÉSENTATION DE L'ENCHAÎNEMENT DE L'APPLICATION

Création d'un mot de passe pour se connecter

Ensuite, nous mettons le mot de passe et le confirmer. Nous cochons ensuite la case et nous cliquons sur le bouton "Créer". Un nouveau mot de passe pour le compte a été créé.

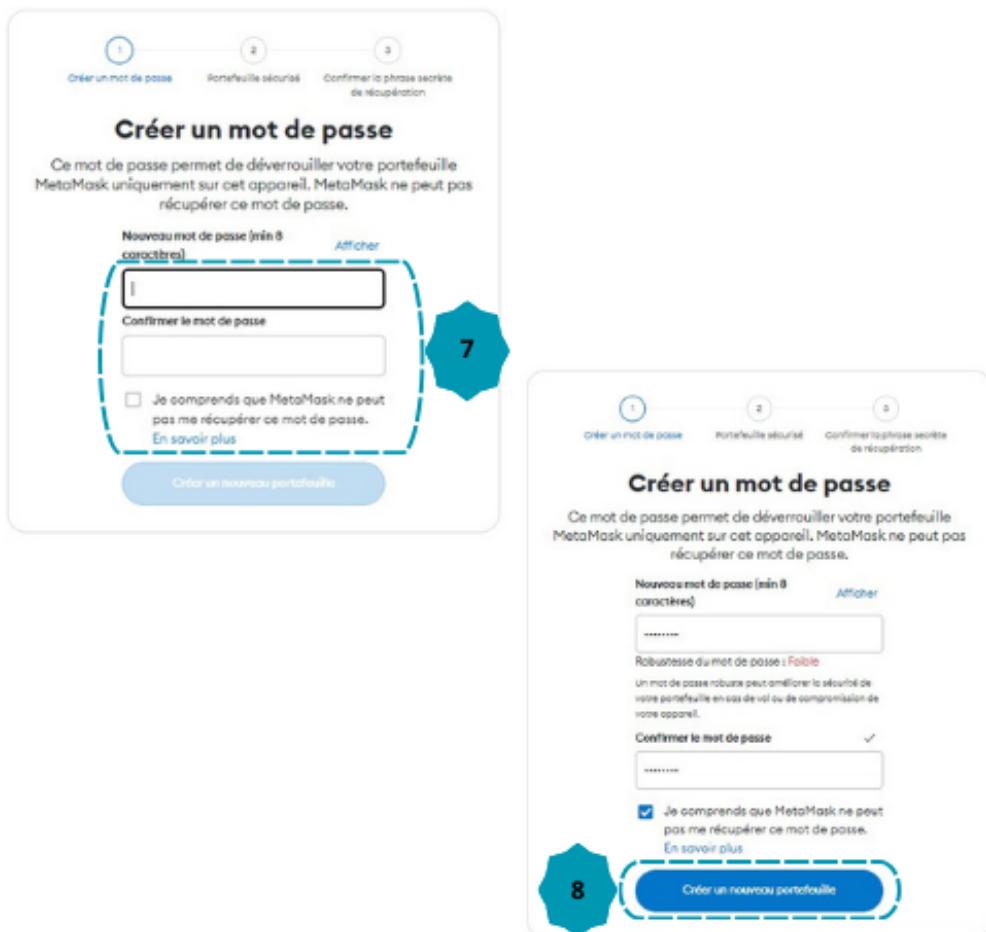


FIGURE 4.4 – Création du mot de passe sur le portefeuille Meta Mask

Réception d'une clé de récupération de 12 mots

À l'étape suivante, notre clé est obtenu, nous pouvons l'utiliser pour récupérer le portefeuille. Pour afficher la clé, nous cliquons sur la zone verrouillée afin que les clés de phase de 12 mots puissent être déverrouillées.

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION



FIGURE 4.5 – Phrase secrète de récupération

Saisie de la clé de récupération dans l'ordre

Lorsque nous passons à la page suivante, nous ferons le verrouillage d'un espace et sélectionnons les 12 phrases qui sont données dans un ordre séquentiel. S'il y a une phrase déplacée, la clé ne fonctionnera pas. Après avoir entré la clé correctement, nous cliquons sur le bouton **"Confirmer"**.

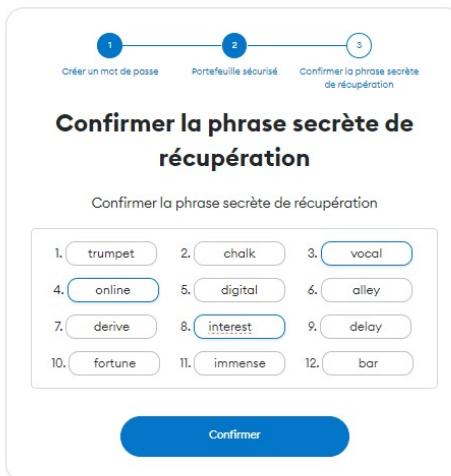


FIGURE 4.6 – Confirmer Phrase secrète de récupération

Une fois la clé saisir correctement, la page suivante apparaît :

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION



FIGURE 4.7 – Compte Meta Mask crée avec succès

B- Installer Ganache

Dans cette étape, nous devons faire l'installation de Ganache afin de le connecter avec MetaMask et l'utiliser comme un crypto monnaie de test.

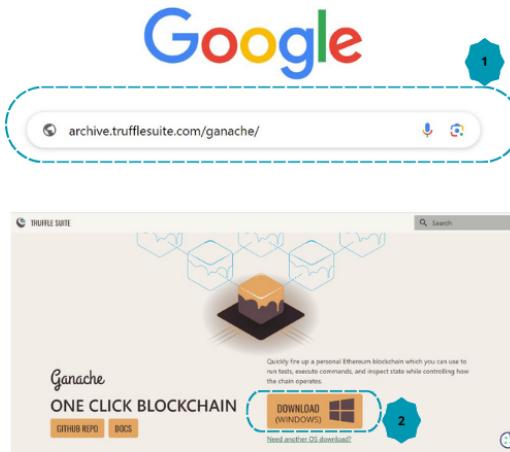


FIGURE 4.8 – Site officiel de Ganache

Nous cliquons sur le bouton installer pour lancer l'installation et sur l'étiquette de lancer une fois prêt.

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION



FIGURE 4.9 – Lancer l'installation

Nous cliquons sur continuer et nous créons un nouveau espace de travail avec un clique sur new Workspace.

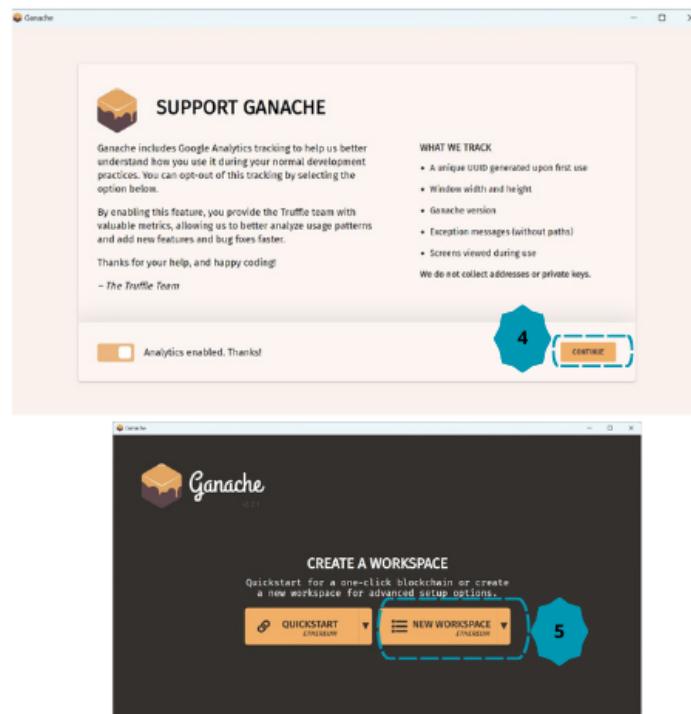


FIGURE 4.10 – Crédit d'un workspace

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

Après la création d'un espace de travail, des informations sur le serveur et une liste de comptes s'afficheront. Chaque compte est automatiquement doté de 100 ethers, ce qui permet de se concentrer sur le développement de votre application sans vous soucier des fonds.

MNEMONIC	HD PATH
federal kiss tired promote educate window crumble icon embrace credit bottom erode	m/44'/69'/0'*account_index
ADDRESS 0xfE2cD07f6CEc3eE758B8d20406FA1145AAC199BD	BALANCE 100.00 ETH TX COUNT INDEX 0 0
ADDRESS 0x635AE414bc80b8D4d8B1eD5E21A65F42e7890d9D	BALANCE 100.00 ETH TX COUNT INDEX 1 0
ADDRESS 0x885cffFc136f5563245FF715805e69862685C0ad6	BALANCE 100.00 ETH TX COUNT INDEX 2 0
ADDRESS 0x337258dB03F3068F3c86FECD240335bF5e106006	BALANCE 100.00 ETH TX COUNT INDEX 3 0
ADDRESS 0xdB860946BBd3bbb83D4BBb2F656381d0Cc6BB771	BALANCE 100.00 ETH TX COUNT INDEX 4 0

FIGURE 4.11 – Espace de travail créé

C- Connecter Ganache avec MetaMask

Maintenant, nous allons convertir notre navigateur Web en navigateur Blockchain, en configurant MetaMask pour communiquer avec Ganache. Nous ouvrons Meta Mask et nous allons dans la liste déroulante des réseaux Ethereum (Ethereum Mainnet) et Cliquons sur “Ajouter un réseau” ou “Add Network”.

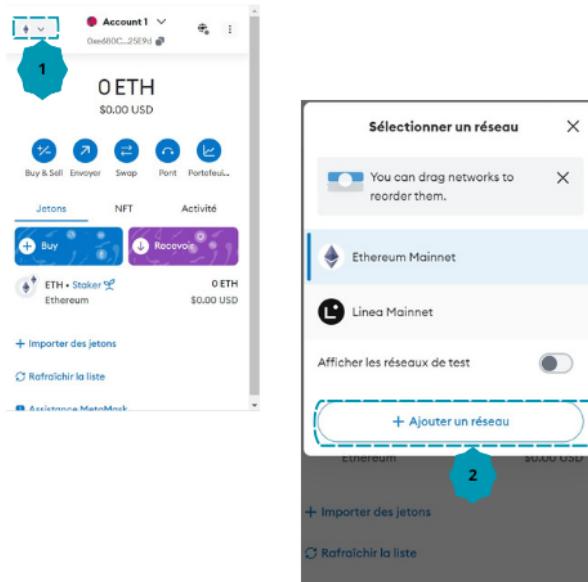


FIGURE 4.12 – Ajouter un réseau etherum

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

Comme indiqué ci-dessous dans la capture d'écran, nous donnons notre nom de réseau, puis pour une nouvelle URL RPC, nous allons à ganache, là nous pouvons trouver l'URL du serveur RPC, copiez-la et collez-la dans le formulaire "Ajouter un réseau" dans Metamask. Ensuite, l'ID de chaîne pour la ganache est 1337. Nous pouvons donner le symbole monétaire comme "**ETH**" ou "**CPAY**" puis cliquons sur enregistrer. Nous sommes connectés avec Ganache maintenant.

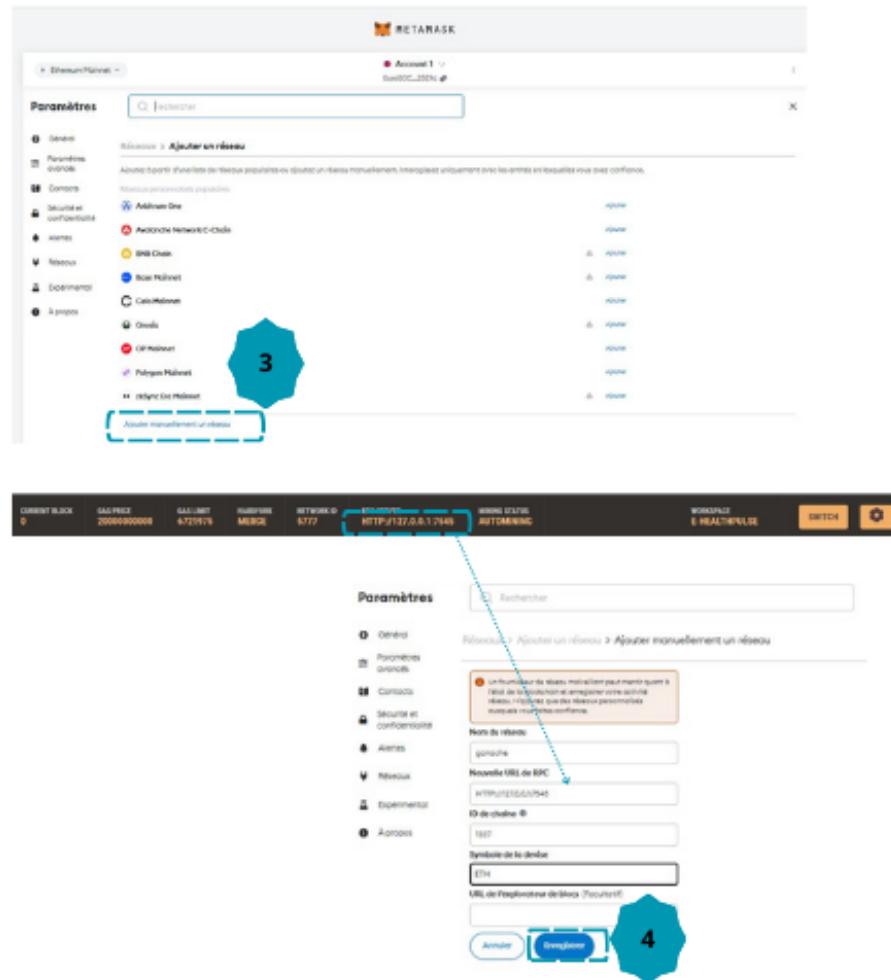


FIGURE 4.13 – enregistrer le réseau ganache

Après cela, cette fenêtre s'affichera avec une notification de réussite de création d'un nouveau compte metamask connecter à ganache.

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

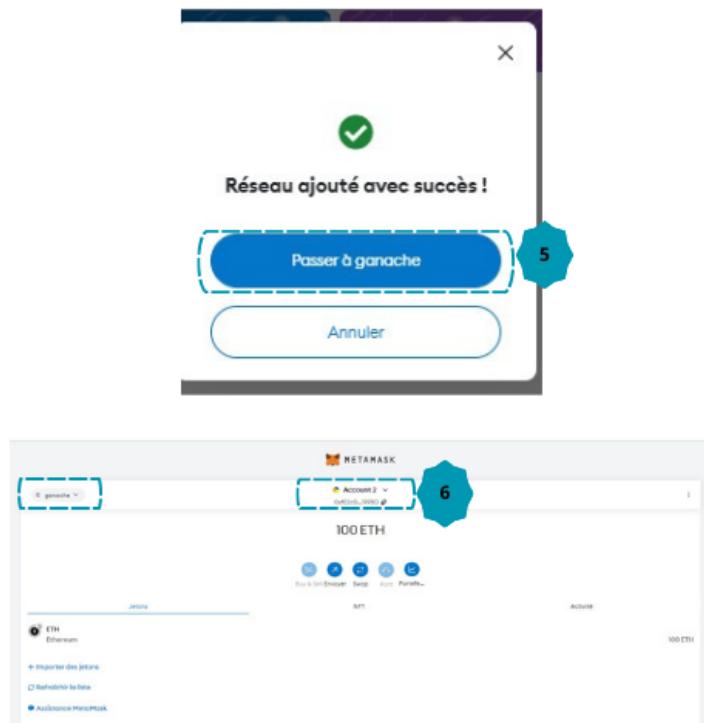


FIGURE 4.14 – réseau ajoutée avec succès

Le compte ganache crée ne possède pas d'argent sur le solde. Il faut impoter un compte ganache avec un solde 100ETH. Nous le détaillons plus tard sur l'alimentation du compte.

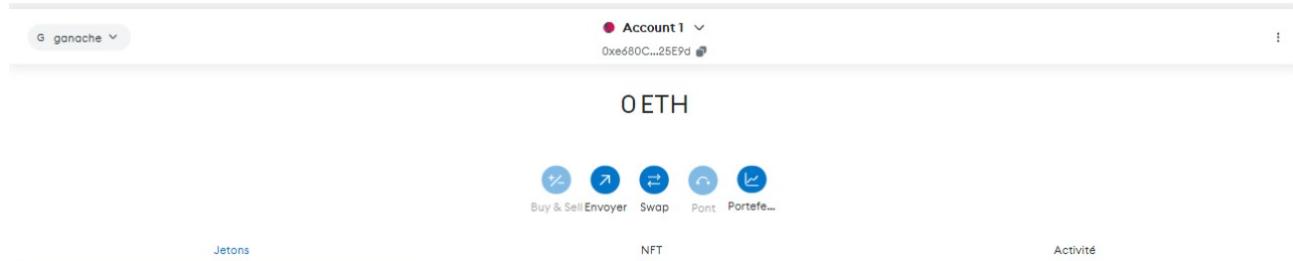


FIGURE 4.15 – Compte ganache vide d'argent

Importer un compte Ganache dans notre portefeuille Metamask

Maintenant, nous allons trouver une clé devant les adresses dans Ganache, il suffit de cliquer dessus et de copier la clé privée de ce compte.

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

MNEMONIC	HD PATH
federal kiss tired promote educate window crumble icon embrace credit bottom erode	m44'/60'/0'account_index
ADDRESS	BALANCE
0xfE2cD07f6CEc3eE758B8d20406FA1145AAC199BD	100.00 ETH
ADDRESS	BALANCE
0x635AE414bc80b8D4d8B1eD5E21A65F42e7090d9D	100.00 ETH
ADDRESS	BALANCE
0x8B5cFFc136f5563245FF715805e69862685C0ad6	100.00 ETH
ADDRESS	BALANCE
0x337258dB03F3068F3c86FEC0D240335bF5e106006	100.00 ETH
ADDRESS	BALANCE
0xdB80946BBd3bbb83D4B8b2F656381d0Cc6BB771	100.00 ETH
ADDRESS	RATE
	TX COUNT
	INDEX
	TX FEE RATE
	INFO

FIGURE 4.16 – récupération de l'adresse pour le compte

Nous avons fait l'importation du compte avec un clique sur Add Account et nous copions l'un des clé privée du compte ganache pour l'utiliser.

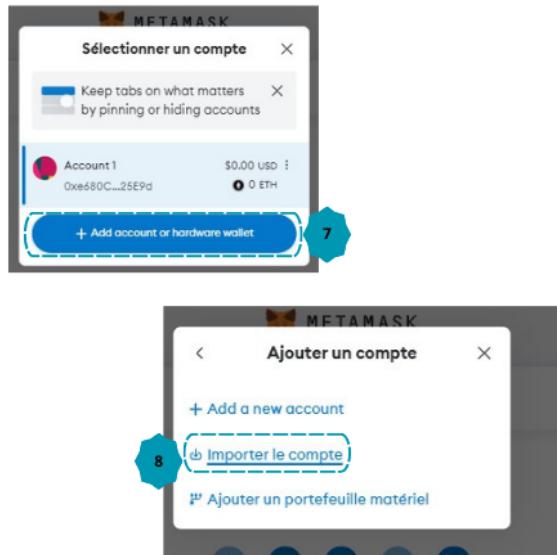


FIGURE 4.17 – Préparer un nouveau compte ganache

Nous avons copier l'une des clé privée du compte ganache et nous le mettons dans le champs importer un compte afin de l'utiliser dans le portefeuille metamask.

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

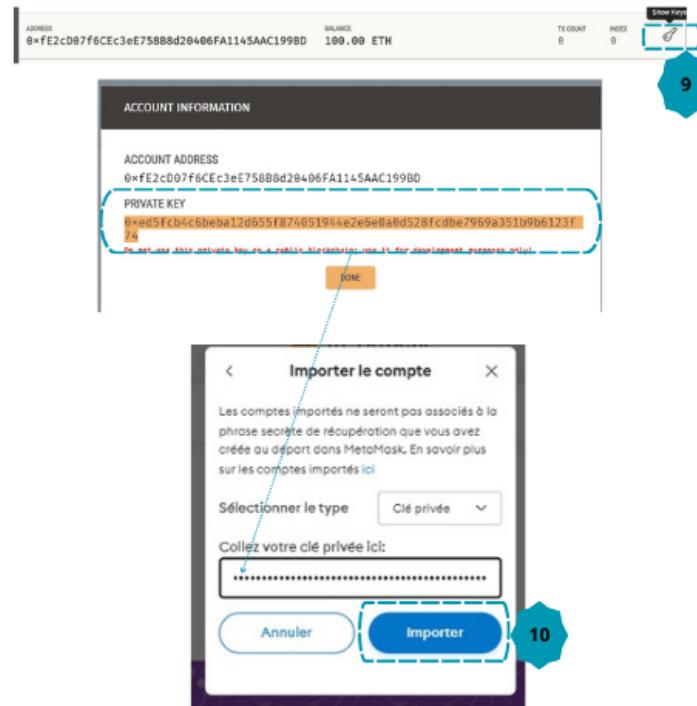


FIGURE 4.18 – Importation du compte depuis Ganache

MetaMask est maintenant connecté avec succès au nœud requis sur la blockchain de ganache. Désormais, nous allons déployer un contrat intelligent écrit en solidity. Cela consommera du gaz de notre compte et vérifiera si les modifications sont reflétées dans notre portefeuille MetaMask ou non.

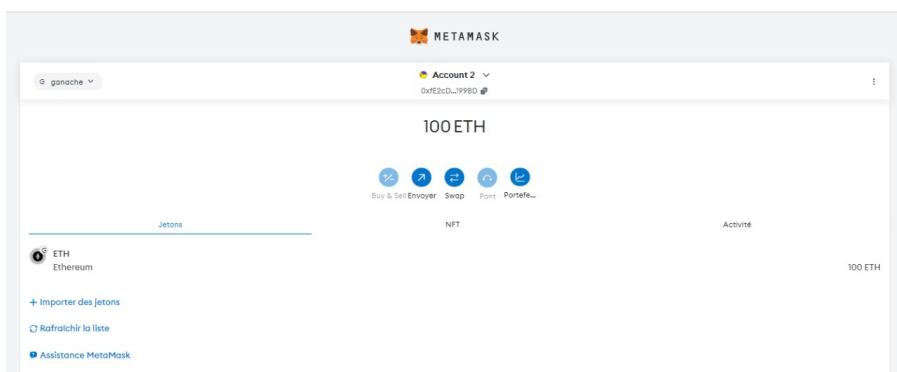


FIGURE 4.19 – Compte approuvé par MetaMask

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

D- Comment déployer notre smart contract sur Ethereum Mainnet

Notez que le déploiement sur Ethereum Mainnet nous obligera à payer des frais de gaz. De plus, chaque fois que nous écrivons sur la blockchain (par exemple, lors de l'exécution de la fonction Add () de notre Smart Contract), cela nous coûterait des frais de gaz, mais la lecture du smart contract serait toujours gratuite. Nous dirigeons vers le site officiel de Remix à partir de ce lien : <https://remix.ethereum.org/>.

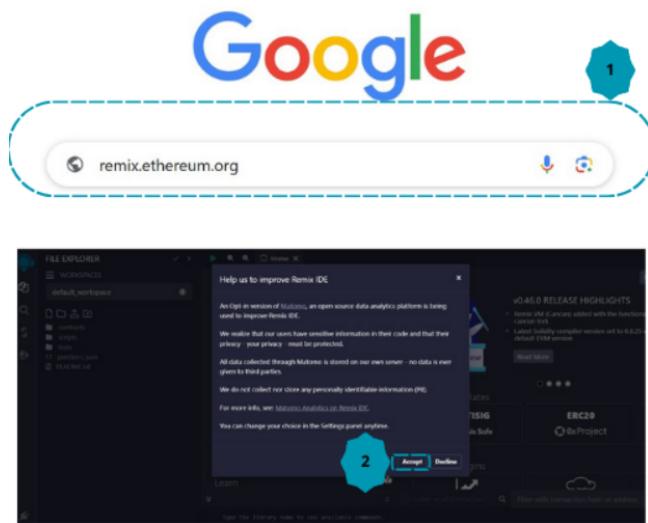


FIGURE 4.20 – Interface d'accueil du RemixIDE

Maintenant, nous créons ou modifions le projet existant, nous créons le fichier **blockchain.sol** et y collons tout notre code de contrat intelligent.

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

Dans la barre latérale, nous pouvons voir une option pour basculer l'icône **SOLIDITY COMPILER** dans la barre latérale. Cliquons dessus et compilons le code du contrat intelligent Solidity. Il devrait créer les fichiers ABI et bytecode. Après avoir compilé, nous ouvrons l'icône **DEPLOY** et **RUN TRANSACTIONS** dans la barre latérale. Il est maintenant temps de déployer sur le réseau principal (Mainnet). Tout d'abord, nous nous assurons d'avoir un solde Ethereum réel dans notre portefeuille Metamask, afin que nous puissions payer les frais Gas. Habituellement, nous pouvons mettre environ une valeur de ETH dans notre portefeuille de n'importe quelle plate-forme de marché d'échange de crypto-monnaie comme Binance, Uphold, Coinbase, etc, puis transférer le **ETH** vers le portefeuille Metamask. Un échange crypto-monnaie est un marché où vous pouvez acheter et vendre des crypto-monnaies, comme Bitcoin, Ether, Monero ou toute autre crypto-monnaie que nous voulons. Dans le panneau **DEPLOY** et **RUN TRANSACTIONS**, nous définissons l' "Environnement" de JavaScript VM dans Injected Web3. Lorsque nous le ferons, nous connecterons notre portefeuille Metamask au réseau principal (Ethereum Mainnet)

Nous ouvrons metamask et nous cliquons sur le bouton continuer pour se connecter avec remixide.

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

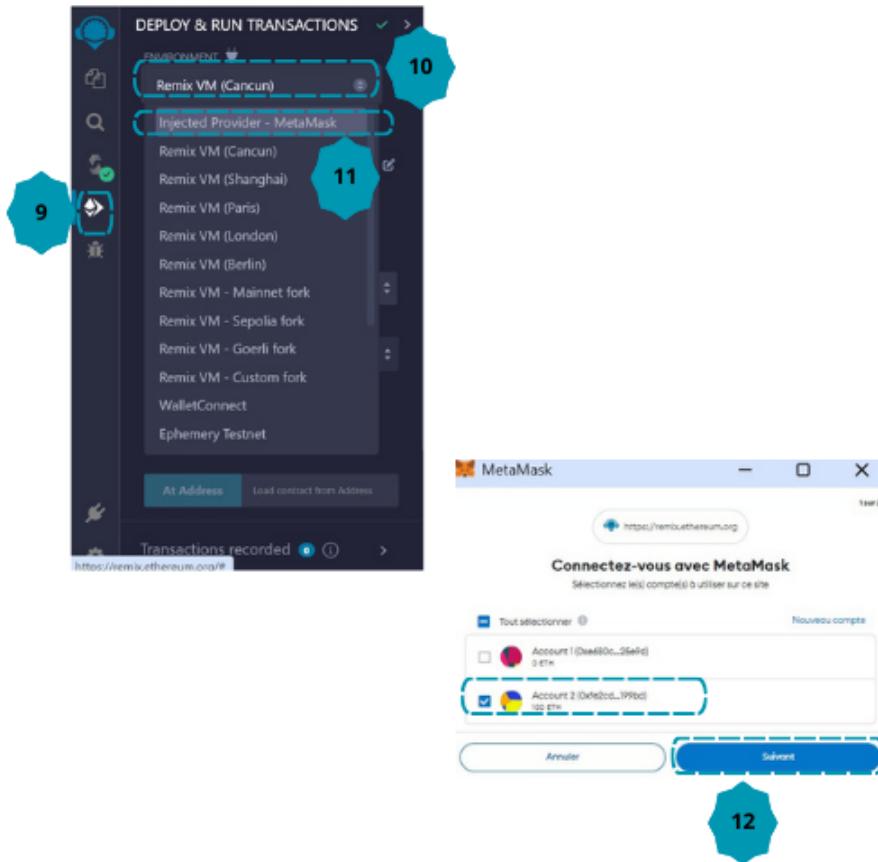


FIGURE 4.21 – Injecter Notre compte MetaMask avec remix-ide

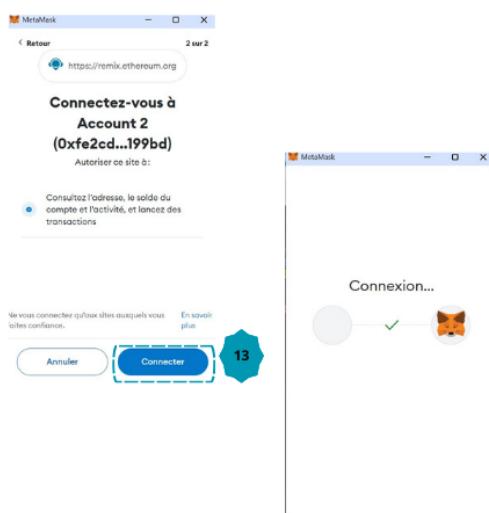


FIGURE 4.22 – RemixIDE connectée avec le compte MetaMask

Pour faire le déploiement,nous copions l'adresse héxadécimale du compte metamask et le

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

mettre dans le champs d'injection metamask puis nous cliquons sur déployer pour faire le déploiement et sur le bouton de confirmation pour confirmer la transaction.

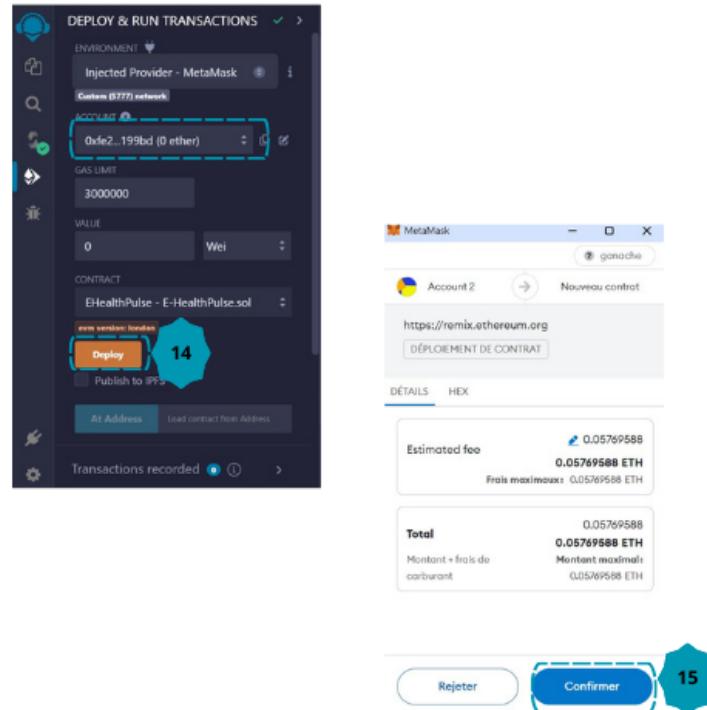


FIGURE 4.23 – Lancer le déploiement

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

Lorsque le déploiement est réussi avec succès ,un message s'affiche dans l'invite de commande du blockchain et nous pouvons copier l'adresse du déploiement et mettre dans notre code pour connecter notre application avec le réseau blockchain.

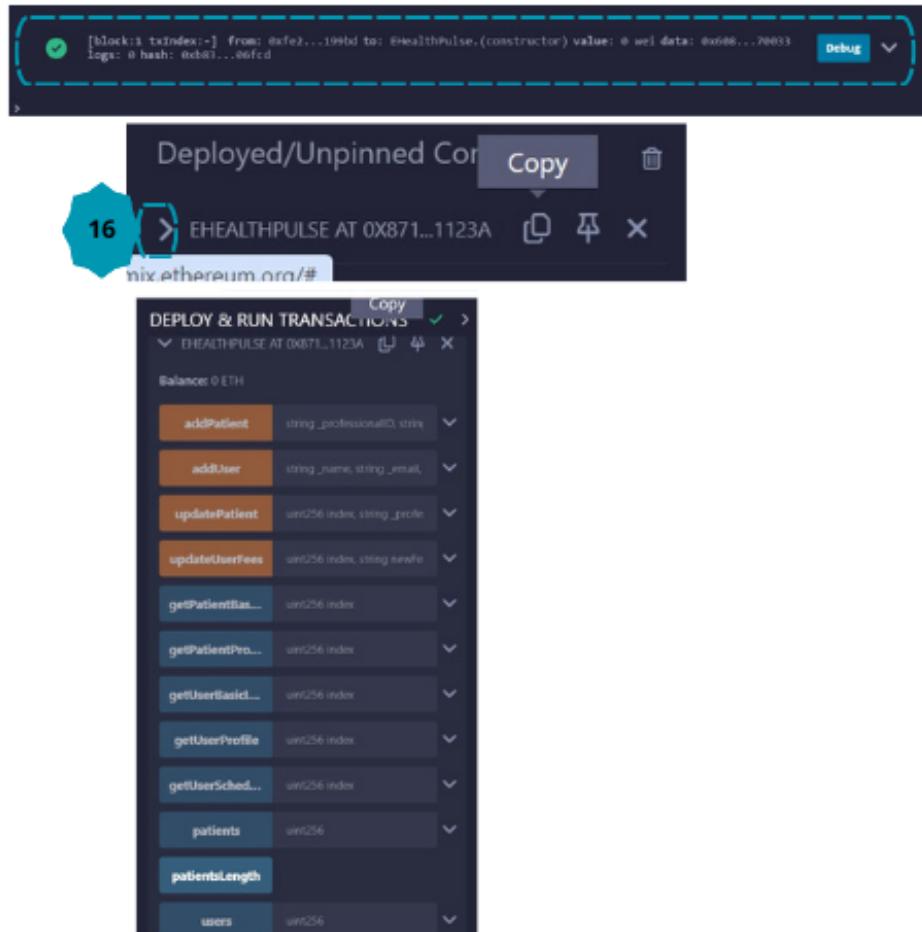


FIGURE 4.24 – déploiement réussi avec succès

E- Comment Analyser notre smart contract sur caliper

Dans cette étape, nous ferons analyser notre code du smart contract en utilisant l'outil de test caliper afin d'avoir des analyses statiques et la présence des vulnérabilités de sécurité ou non. Pour cela nous activons cette méthode et faire l'exécution dans Remix IDE et dans notre machine locale. Dans un premier temps, nous redirigeons vers la barre latéral gauche du remixide puis aller vers plugin manager pour faire activer l'outil caliper. Après, nous importons le projet Dans Remix IDE et se connecter en mode localhost.

4.3.2 Description des interfaces d'application

Nous décrivons dans cette partie les étapes à suivre par un utilisateur qui souhaite se connecter pour la première fois à notre application. Ce scénario de test se termine par l'ajout et la modification des données du marin ou utilisateur. Nous vous montrerons toutes les étapes avec des captures d'écran.

1) Lancement et Test de l'application

Interface de connexion : Pour lancer notre application nous commençant tout d'abord à demarrer notre serveur pour que le modèle s'exécute et fonctionner avec le flask en accédant à l'adresse locale , il y a la connexion par l'envoie de requête http et la réponse préférée en exécutant la commande ‘py server.py’

```
PS C:\Users\Copytop\Desktop\BLOCKCHAIN2811liaison\danger_classification_app - Copie\assets> py server.py
2024-12-26 09:13:07.720754: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-12-26 09:13:12.691263: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
* Serving Flask app 'server'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.1.22:5000
Press CTRL+C to quit
```

FIGURE 4.25 – Lancement de serveur

Par suite en accédant vers notre interface avec flask ou tousles tests et affichage fait d'après notre model bien testé :

Notre dataset : Le dataset utilisé est constitué d'un ensemble de valeurs supposées provenant des capteurs installés sur les bateaux des marins, qui détectent la pollution (déversement d'hydrocarbures). Ce dataset est utilisé pour l'entraînement et la préparation du modèle :

La préparation de notre modèle se fait en intégrant notre dataset après un prétraitement, où ce modèle permet de détecter si la valeur mesurée dans la mer est normale (pas de degré de pollution dangereux) ou anormale (très dangereuse). Les données normales sont stockées dans une base de données centralisée, tandis que les données anormales sont enregistrées sur la blockchain pour alerter l'utilisateur ou le marin.

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

```

○ PS C:\Users\Copytop\Desktop\Blockchain2811liaison\danger_classification_app - Copie> flutter run
Connected devices:
Windows (desktop) • windows • windows-x64 • Microsoft Windows [version 10.0.22631.4602]
Chrome (web)       • chrome  • web-javascript • Google Chrome 131.0.6778.205
Edge (web)         • edge    • web-javascript • Microsoft Edge 131.0.2903.112
[1]: Windows (windows)
[2]: Chrome (chrome)
[3]: Edge (edge)
Please choose one (or "q" to quit): 2
Launching lib/main.dart on Chrome in debug mode...
Waiting for connection from debug service on Chrome...          26.5s
This app is linked to the debug service: ws://127.0.0.1:64517/DpbS05xfa6s=/ws
Debug service listening on ws://127.0.0.1:64517/DpbS05xfa6s=/ws

To hot restart changes while running, press "r" or "R".
For a more detailed help message, press "h". To quit, press "q".

A Dart VM Service on Chrome is available at: http://127.0.0.1:64517/DpbS05xfa6s=
The Flutter DevTools debugger and profiler on Chrome is available at:
http://127.0.0.1:9101?uri=http://127.0.0.1:64517/DpbS05xfa6s=
```

FIGURE 4.26 – Présentation de l’application

	A	B	C	D	E	F	G	H
A15		X ✓ fx	14,KÃ©rosÃ¨ne,4773,4.54,45,2500,Oui					
7	6,Gazole,3056,3.61,53,2000,Oui							
8	7,PÃ©trole,2346,3.63,61,3000,Non							
9	8,Mazout,4941,4.5,14,3500,Oui							
10	9,KÃ©rosÃ¨ne,3393,1.45,12,2500,Oui							
11	10,Mazout,3940,4.03,56,3500,Oui							
12	11,KÃ©rosÃ¨ne,2650,2.94,16,2500,Oui							
13	12,Mazout,4418,4.18,72,3500,Oui							
14	13,KÃ©rosÃ¨ne,4193,0.88,62,2500,Oui							
15	osÃ¨ne,4773,							
16	15,Mazout,893,2.2,18,3500,Non							
17	16,Mazout,909,4.97,68,3500,Non							
18	17,Mazout,1935,3.24,71,3500,Non							
19	18,PÃ©trole,2715,4.33,62,3000,Non							
20	19,Mazout,3319,2.9,68,3500,Non							
21	20,PÃ©trole,2273,1.74,35,3000,Non							
22	21,Gazole,1480,1.29,63,2000,Non							
23	22,Mazout,672,4.91,72,3500,Non							
24	23,Gazole,3332,1.74,64,2000,Oui							

FIGURE 4.27 – Présentation de Dataset

Inscription : Si nous nous connectons à l’application pour la première fois, nous remplissons le formulaire d’inscription afin de créer un nouveau compte.

Confirmation de la transaction : Après avoir installé MetaMask et créé notre portefeuille, lorsque nous cliquons sur le bouton d’inscription, une boîte MetaMask apparaît pour effectuer la transaction et confirmer les frais de gaz afin que le compte du marin ou utilisateur soit chargé sur la blockchain.

Le système vérifie les données, calcule le coût de la transaction puis demande à l’utilisateur de confirmer : - Si le solde est insuffisant par rapport à celui demandé, la transaction ne pourra

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

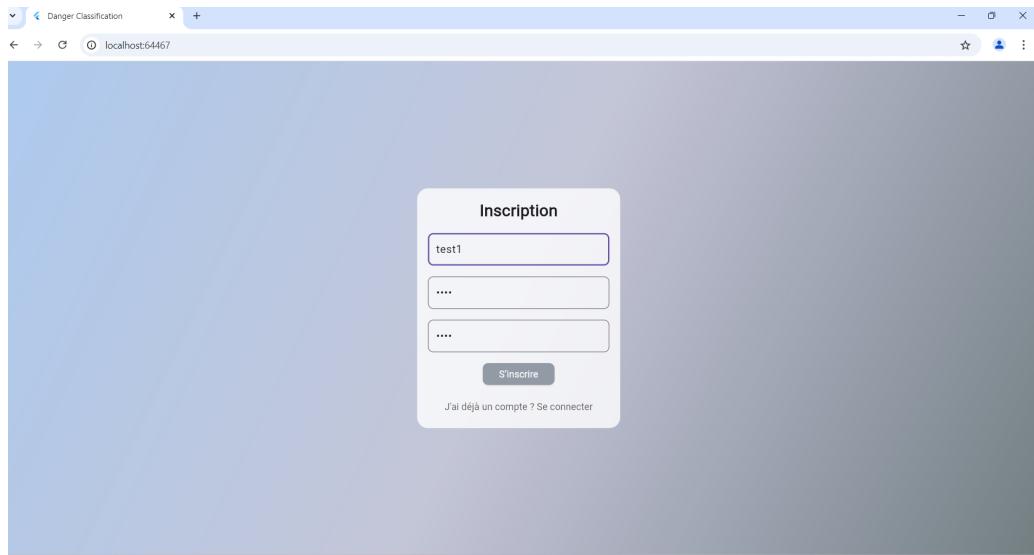


FIGURE 4.28 – Interface d'inscription

pas être validée. - Le coût de gaz est exprimé en **ETH** (crypto-monnaie Ethereum).

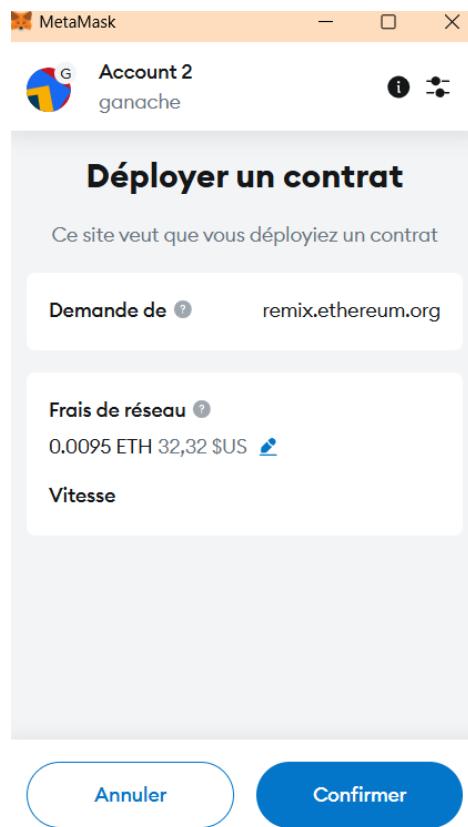


FIGURE 4.29 – Confirmation de la transaction

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

Interface d'authentification du marin ou utilisateur : L'interface d'authentification représente le départ de notre application. Cette interface permet d'accéder aux autres interfaces. Pour cela, nous allons vers l'interface d'authentification pour nous authentifier et accéder à la page d'accueil.

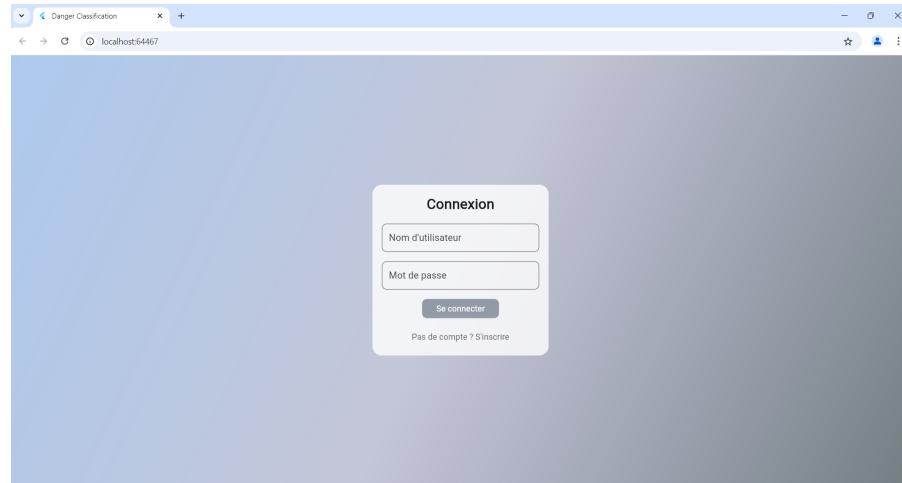


FIGURE 4.30 – Interface d'authentification du marin ou utilisateur

L'utilisateur est bien authentifié donc toute la liste est bien stockée dans la base des données :

	<input type="checkbox"/> Éditer	<input type="checkbox"/> Copier	<input type="checkbox"/> Supprimer	id	username	password
<input type="checkbox"/>	<input type="checkbox"/> Éditer	<input type="checkbox"/> Copier	<input type="checkbox"/> Supprimer	1	test	\$2b\$12\$PAJBQTXk9lw/KIMnRMGgMuIMCKgt1hUUmvX21VQxu6z...
<input type="checkbox"/>	<input type="checkbox"/> Éditer	<input type="checkbox"/> Copier	<input type="checkbox"/> Supprimer	2	test1	\$2b\$12\$3GCWBTFurqfHoC.A0xJ78emIswuDq QK5XZVGcw79DL...

FIGURE 4.31 – Utilisateur connecté-base de données

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

Interface Profil Marin ou utilisateur : Nous accédons au profil marin ou utilisateur en cliquant sur le bouton "Profil". **Interface d'ajout des paramètres dans la mer :** Nous allons ajouter des données : quantité, zone affectée et durée comme des paramètres "features" à tester pour détecter la pollution. Les étapes sont les suivantes :

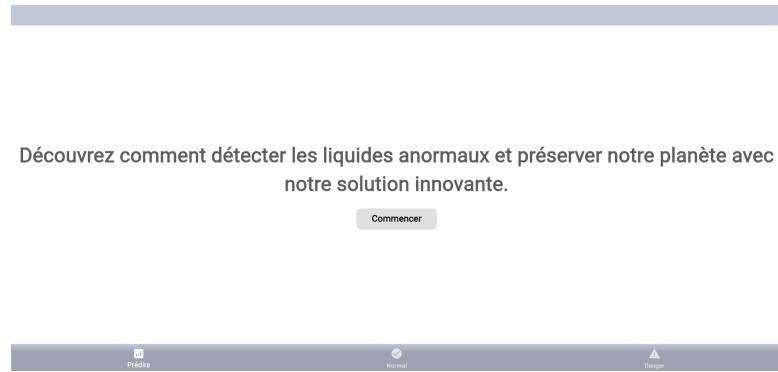


FIGURE 4.32 – Accès vers l'interface d'ajout

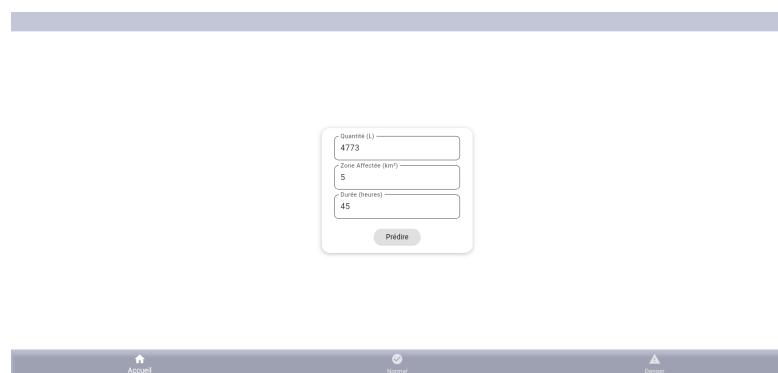


FIGURE 4.33 – Ajout des features à tester

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

Confirmation de la transaction d'ajout : Une boîte MetaMask apparaîtra pour effectuer la transaction et confirmer les frais de gaz afin que le compte du marin ou utilisateur soit chargé sur la blockchain. Nous cliquons sur le bouton de confirmation pour valider la transaction.

Un email est envoyé au marin ou utilisateur pour l'informer de la création de son compte avec succès. Dans cet email, il peut voir son adresse email et son mot de passe.

Nous pouvons consulter les données ajoutées par le marin ou utilisateur dans phpMyAdmin. Ici, le test est effectué avec des valeurs normales qui sont stockées dans la base de données :

The screenshot shows the phpMyAdmin interface for a database named 'pollution'. The current table is 'sans_danger'. The SQL query displayed is 'SELECT * FROM `sans_danger`'. The results show four rows of data:

	id	Duree	Quantite	ZoneAffectee
<input type="checkbox"/>	1	11	125	1.5
<input type="checkbox"/>	2	6	100	5
<input type="checkbox"/>	3	8	1000	7
<input type="checkbox"/>	4	61	2346	4

FIGURE 4.34 – Liste des données normales dans phpMyAdmin

Avec l'affichage de ce résultat sans alerte dangereuse :



FIGURE 4.35 – Résultat de test sans danger

Nous refaisons le test maintenant avec des valeurs anormales, qui seront ensuite stockées dans la blockchain : Dans les figures suivantes au dessous nous avons présentés nos etapes faites

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

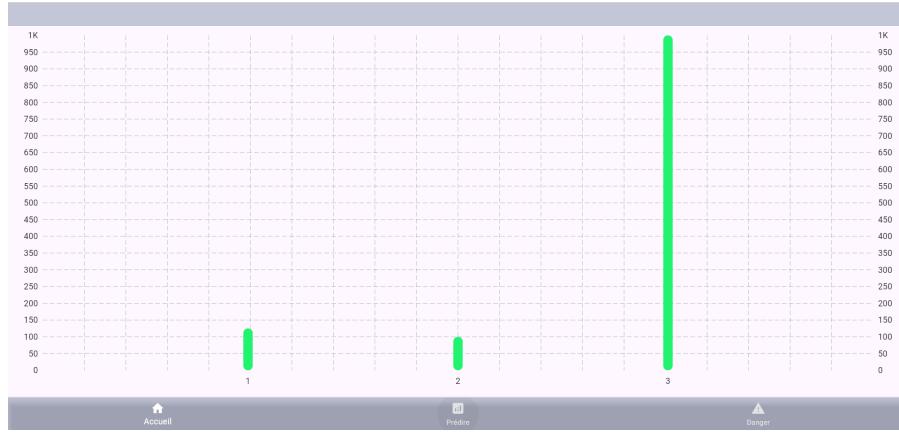


FIGURE 4.36 – Test de modèle sur features normales

pour l'affichages de resultat de teste de notre modelle avec notre interface Flutter :



FIGURE 4.37 – Transaction des de données anormales detectées

-Transaction est faite alors la resultat est bien affichée :

4.3. PRÉSENTATION DE L'ENCHAINEMENT DE L'APPLICATION

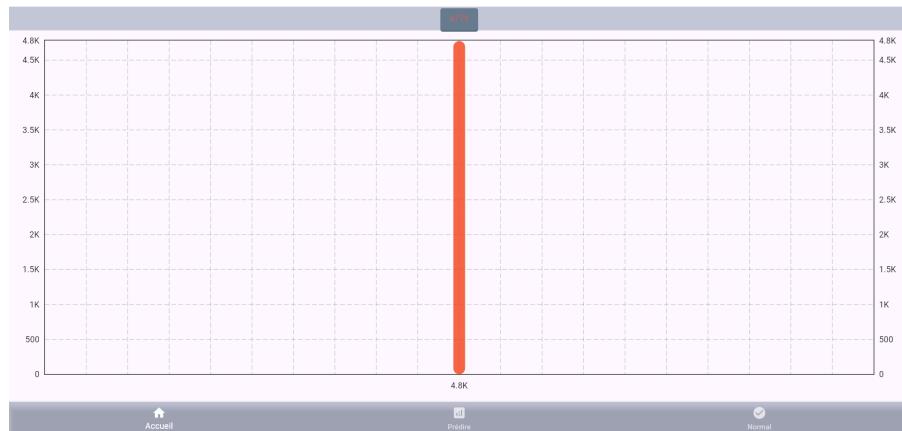


FIGURE 4.38 – Affichage de test de danger

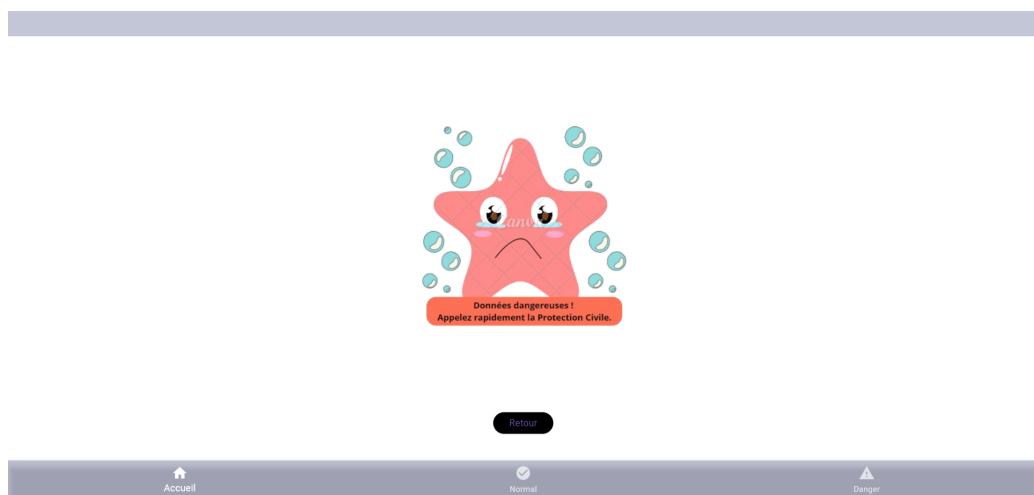


FIGURE 4.39 – Test de modèle avec valeurs anormales

Conclusion Générale

En résumé, notre projet s'est concentré sur le développement d'une solution innovante pour la détection et la gestion de la pollution marine par hydrocarbures. En combinant l'intelligence artificielle et la technologie blockchain, nous avons conçu une plateforme robuste et efficace permettant de surveiller en temps réel la qualité des eaux, d'identifier rapidement les sources de pollution et de garantir la sécurité ainsi que la traçabilité des données collectées.

Cette solution offre plusieurs avantages clés : une détection précise des anomalies, des notifications instantanées aux parties concernées via une application multiplateforme intuitive, et une sécurisation renforcée des informations critiques grâce à la blockchain. Ces fonctionnalités visent non seulement à améliorer la réactivité face aux incidents mais aussi à garantir une gestion transparente et fiable des données environnementales, contribuant ainsi à la préservation de l'écosystème marin.

Malgré les défis techniques rencontrés, notamment en ce qui concerne les conditions météorologiques extrêmes et la complexité de l'intégration des technologies, nous avons atteint nos objectifs principaux. Ce projet a non seulement permis de mettre en pratique des connaissances théoriques, mais également d'acquérir de nouvelles compétences dans des domaines tels que l'IA, le développement d'applications web et mobiles, ainsi que la mise en œuvre de contrats intelligents sur une blockchain.

Pour l'avenir, plusieurs perspectives s'ouvrent :

Optimisation des modèles IA pour une meilleure précision dans des conditions difficiles. Extension de la couverture géographique pour une surveillance plus large. Amélioration des interfaces utilisateur afin de les rendre encore plus accessibles. Renforcement de la sécurité des données pour prévenir toute vulnérabilité potentielle. En conclusion, notre travail constitue une avancée significative dans la lutte contre la pollution marine, démontrant comment l'intégration de technologies avancées peut répondre efficacement à des enjeux environnementaux critiques.

Bibliographie

- [1] <https://www.blockchain.com/> [Consulté : 10 Novembre 2023]
- [2] <https://ibaxnetwork.medium.com/blockchain-evolution-from-1-0-to-4-0-18aa9ca2dbbb>
[Consulté : 11 Novembre 2023]
- [3] <https://www.cmcmarkets.com/fr-fr/apprendre-a-trader-les-crypto-monnaies/qu-est-ce-que-l-ethereum> [Consulté : 15 Novembre 2023]
- [4] https://www.emerald.com/insight/content/doi/10.1108/MF-06-2019-0314/full_html [Consulté : 17 Novembre 2023]
- [5] <https://blog.logrocket.com/building-dapp-flutter-solidity/> [Consulté : 20 Novembre 2023]
- [6] <https://meritis.fr/blog/blockchain-principes-et-fonctionnement/> [Consulté : 24 Novembre 2023]
- [7] <https://www.sciencedirect.com/science/article/abs/pii/S0168169919324329>
[Consulté : 24 Novembre 2023]
- [8] <https://ethereum-blockchain-developer.com/2022-03-deposit-withdrawals/07-injected-web3-provider/> [Consulté : 5 Décembre 2023]
- [9] <https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.22+commit.4fc1097e.js> [Consulté : 6 Décembre 2023]
- [10] <https://metamask.io/> [Consulté : 20 Décembre 2023]
- [11] <https://archive.trufflesuite.com/ganache/> [Consulté : 22 Décembre 2023]
- [12] <https://faucet.goerli.starknet.io/> [Consulté : 25 Décembre 2023]

BIBLIOGRAPHIE

- [13] <https://www.sciencedirect.com/science/article/abs/pii/S0168169919324329>
[Consulté : 24 Novembre 2023]
- [14] [Jameela Al-Jaroodi and Nader Mohamed. Blockchain in industries: A survey.](#) IEEE Access
[Consulté : 30 Décembre 2023]
- [15] [Ahmed Afif Monrat, Olov Scheln, and Karl Andersson. A survey of blockchain from the perspectives of challenges, and opportunities.](#) IEEE Access [Consulté : 12 Janvier 2024]
- [16] [Michael Nofer, Peter Gomber, Oliver Hinz, and Dirk Schiereck. Blockchain. Business & Information Systems Engineering](#) [Consulté : 13 Janvier 2024]
- [17] [Kushal Patil, Nirman Sonawane, Ekta Patil, Kshitija Kulkarni, and Puja Padiya. Blockchain-based security for super-peer wireless sensor networks.](#) In IC-BCT2019: Proceedings of the International Conference on Blockchain Technology, pages 241–256 [Consulté : 16 Janvier 2024]
- [18] <mailto:https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203257-html5-hypertext-markup-langage5-definition-traduction/> [Consulté : 18 Janvier 2024]
- [19] <https://developer.mozilla.org/fr/docs/Web/CSS> [Consulté : 18 Janvier 2024]
- [20] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203585-javascript/> [Consulté : 20 Janvier 2024]
- [21] <https://www.pappleweb.com/index/definition-de-nodejs/> [Consulté : 22 Janvier 2024]
- [22] <https://bility.fr/definition-visual-studio-code/> [Consulté : 10 Février 2024]
- [23] <https://queveutdire.com/technologie/phpmyadmin-definition/> [Consulté : 18 Septembre 2024]
- [24] <https://www.futura-sciences.com/tech/definitions/internet-mysql-4640/>
[Consulté : 27 Mars 2024]