|  |  |
| --- | --- |
| \\polyfile\Images\Logos\bonne resolution\logo group Universiapolis\logo polytechnique universia.png |  |
| **Cycle Ingénieur de L’ecole Polytechnique d’agadir**  **filiere Genie Informatique** | |
| **stage ASSISTANT INGENIEUR** | |
| ***Développement d'une application Web 3.0 avec***  ***Backend hybride Cloud/Blockchain*** | |

|  |
| --- |
| Année universitaire 2023/2024 |

|  |  |  |
| --- | --- | --- |
| **Réalisé par :** |  | **Superviseur :** |
| **Hatim Allouch** |  | **M. Youssef Meghzaz**  **M. Mohammed Toub** |
|  |  | **Tuteur :** |
|  |  | **M. Abdelaaziz El Aissaoui** |

**REMERCIEMENTS**

Le faire dans l’ordre hiérarchique. Je tiens à exprimer ma sincère gratitude envers Monsieur Youssef Meghzaz, ingénieur informatique (Full stack/Web3) et CPO chez FeverTokens, ainsi que le Docteur Mohammed Toub, ingénieur informatique (Web3/Backend) et CTO chez FeverTokens. Leur expertise exceptionnelle et leur dévouement envers l'innovation et la technologie m'ont profondément inspiré tout au long de mon stage au sein de l'entreprise.

Monsieur Youssef Meghzaz, en tant qu'ingénieur (Full stack/Web3) et Chief Product Officer, a été un guide précieux dans ma compréhension des concepts complexes de la blockchain et du développement d'applications Web3. Sa capacité à démystifier les idées techniques tout en les présentant de manière accessible a été une source d'apprentissage continu pour moi. Sa passion pour la création d'interfaces utilisateur intuitives et d'expériences utilisateur exceptionnelles a été une source d'inspiration pour ma propre croissance professionnelle.

Quant au Docteur Mohammed Toub, en tant qu'ingénieur Web3/Backend et Chief Technology Officer, sa profonde connaissance des technologies sous-jacentes de la blockchain et son engagement envers la sécurité et la robustesse des solutions ont façonné ma perception de l'importance de la qualité et de la vérifiabilité dans le développement. Sa capacité à anticiper les tendances émergentes dans le domaine de la technologie m'a montré à quel point l'adaptabilité et l'innovation sont essentielles pour réussir dans un environnement en constante évolution.

Leur leadership, leur mentorat et leur engagement envers le partage des connaissances ont grandement enrichi mon expérience de stage. Je les remercie chaleureusement d'avoir partagé leur expertise, de m'avoir encouragé à explorer de nouvelles perspectives et de m'avoir soutenu dans la réalisation de mes projets au sein de FeverTokens. Mon stage a été enrichissant grâce à leur guidance et leur soutien constants, et je suis reconnaissant d'avoir eu l'opportunité d'apprendre et de grandir sous leur tutelle.

Je tiens également à exprimer ma reconnaissance envers toute l'équipe de FeverTokens pour leur accueil chaleureux, leur collaboration et leur encouragement tout au long de mon parcours au sein de l'entreprise. Leur passion pour l'innovation et leur détermination à façonner l'avenir de la technologie blockchain ont contribué à faire de mon stage une expérience inoubliable et formatrice.

TABLE DES MATIERES

[INTRODUCTION GENERALE 8](#_Toc150119507)

[CHAPITRE 1 : 11](#_Toc150119508)

[Contexte du stage 11](#_Toc150119509)

[**I-** **INTRODUCTION** 12](#_Toc150119510)

[**II-** **PRESENTATION DE L’ENTREPRISE** 13](#_Toc150119511)

[2-1- Activités 13](#_Toc150119512)

[2-2- Organisation 14](#_Toc150119513)

[2-3- Systèmes de production/service 15](#_Toc150119514)

[**III-** **THEMATIQUE** 16](#_Toc150119515)

[**IV-** **CONCLUSION** 17](#_Toc150119516)

[CHAPITRE 2 : 18](#_Toc150119517)

[Etude bibliographique 18](#_Toc150119518)

[**I-** **INTRODUCTION** 19](#_Toc150119519)

[**II-** **OUTILS ET METHODES** 20](#_Toc150119520)

[1. OUTILS 20](#_Toc150119521)

[2. Méthode de travail: 24](#_Toc150119522)

[**III-** **CONCLUSION** 27](#_Toc150119523)

[CHAPITRE 3 : 28](#_Toc150119524)

[Développement & Analyse 28](#_Toc150119525)

[**I-** **INTRODUCTION** 29](#_Toc150119526)

[WAAS - Wallet as a Service (Portefeuille en tant que Service) : 29](#_Toc150119527)

[Web3 Explorer - Accessibilité à la Blockchain sans Connaissances Préalables : 29](#_Toc150119528)

[Documentation des Smart Contracts avec Natspec : 29](#_Toc150119529)

[Tests et Débogage de l'Application : 29](#_Toc150119530)

[**II-** **DESCRIPTION ET ANALYSE DE L’EXISTANT** 30](#_Toc150119531)

[2-1- Description de l’existant 30](#_Toc150119532)

[2-2- Points Forts du Projet : Analyse des Avantages Clés 30](#_Toc150119533)

[**III-** **DEVELOPPEMENT** 31](#_Toc150119534)

[3. WAAS (Wallet As A Service) 31](#_Toc150119535)

[4. Creating Wallet 44](#_Toc150119536)

[5. Web3 explorer 58](#_Toc150119537)

[CONCLUSION GENERALE 69](#_Toc150119538)

[BIBLIOGRAPHIE 71](#_Toc150119539)

[II. Bibliographie 71](#_Toc150119540)

[ANNEXES 72](#_Toc150119541)

TABLE DES ILLUSTRATIONS

[Figure 1 logo entreprise FeverTokens 12](#_Toc150119461)

[Figure 2 hiérarchie de FeverTokens 14](#_Toc150119462)

[Figure 3 logos Git et Github 20](#_Toc150119463)

[Figure 4 logo Editeur de text utiliser 20](#_Toc150119464)

[Figure 5 logo technologie FrontEnd (Nextjs) 21](#_Toc150119465)

[Figure 6 Logo Service Serverless utiliser 21](file:///C:\Users\AORUS\Desktop\3-Modèle%20rapport%20stage%20ingénieur%20détudes.docx#_Toc150119466)

[Figure 7 logo base de donnee DynamoDB 22](#_Toc150119467)

[Figure 8 Logo librairie EtherJs 22](#_Toc150119468)

[Figure 9 Logo Librairie testing HardHat 23](#_Toc150119469)

[Figure 10 Image explicative a la methode de travail 24](#_Toc150119470)

[Figure 11 GitHub Project l'outils utiliser pour monitorer le progresse des taches 25](#_Toc150119471)

[Figure 12 description d'une des taches effectuer 25](#_Toc150119472)

[Figure 13 Pull-request verifier et terminer 26](#_Toc150119473)

[Figure 14 page principale de WAAS 32](#_Toc150119474)

[Figure 15 Page qui démontre la dynamiter 32](#_Toc150119475)

[Figure 16 page Manage Access Keys 33](#_Toc150119476)

[Figure 17 page qui demontre les packets envoyez vers le server 33](#_Toc150119477)

[Figure 18 la possibilité de enregistrer le pair de clés en local 34](#_Toc150119478)

[Figure 19 code responsable à envoyer la demande de génération de clés 35](#_Toc150119479)

[Figure 20 API responsale a recevoir la requête de génération et la traiter 37](#_Toc150119480)

[Figure 21 fonction responsable à envoyer la requête pour stocker la clé public 39](#_Toc150119481)

[Figure 22 traitement de requête de type post pour stocker la clé public 40](#_Toc150119482)

[Figure 23 la requête pour stocker la clé public dans Dynamo DB 42](#_Toc150119483)

[Figure 24 explication de tout le process d'authentification via cryptographie asynchrone 43](#_Toc150119484)

[Figure 25 bouton responsable a lancé tout le process création du Wallet 45](#_Toc150119485)

[Figure 26 Composent personnaliser Nextjs 46](#_Toc150119486)

[Figure 27 Popup Dynamique 48](#_Toc150119487)

[Figure 28 démonstration de dynamiter pour une expérience d'utilisateur optimal 49](#_Toc150119488)

[Figure 29 code du Pop-up personnaliser 50](#_Toc150119489)

[Figure 30 API responsable a création des wallet 52](#_Toc150119490)

[Figure 31 fonction lambda 53](#_Toc150119491)

[Figure 32 statistiques et monitoring des wallets 54](#_Toc150119492)

[Figure 33 Popup pour changer le nom du wallet selectionner 55](#_Toc150119493)

[Figure 34 API responsable du changement du nom des wallets 55](#_Toc150119494)

[Figure 35 fonction qui lance la requête du changement du nom 56](#_Toc150119495)

[Figure 36 PolygonFaucet plateforme qui fournit des fausse cryptocurrencies 56](#_Toc150119496)

[Figure 37 Confirmation de demande 0.2 matic 57](#_Toc150119497)

[Figure 38 la réception de 0.2 matic demandée 57](#_Toc150119498)

[Figure 39 page principale de WEB3 explorer 58](#_Toc150119499)

[Figure 40 formulaire Dynamique pour importer des contrats intelligentes 59](#_Toc150119500)

[Figure 41 fonction validation d'adresse 60](#_Toc150119501)

[Figure 42 fonction d'initialisation du Provider 63](#_Toc150119502)

[Figure 43 fonction pour récupérer l'ABI 64](#_Toc150119503)

[Figure 44 API pour récupérer les contrats interne 66](#_Toc150119504)

[Figure 45 fonction pour trouver l'API convenable au NetworkID 67](#_Toc150119505)

[Figure 46 validation d'Abi récupéré 68](#_Toc150119506)

# INTRODUCTION GENERALE

L'entreprise FeverTokens opère dans le secteur de la technologie blockchain et de l'écosystème web3, en fournissant une plateforme innovante qui permet aux clients de créer des applications web3 sans nécessiter de connaissances en programmation. Au cours de mon stage, d'une durée de [indiquer la durée du stage], j'ai eu l'opportunité stimulante de rejoindre FeverTokens, une entreprise basée en France au cœur de l'innovation technologique.

Mon expérience de stage s'est déroulée à distance, avec un suivi quotidien et une coordination parfaitement synchronisée. Grâce à des outils de communication avancés et à une utilisation efficace de Git pour le contrôle de version, j'ai pu collaborer étroitement avec les membres de l'équipe malgré la distance géographique. Cette approche agile de travail à distance a renforcé ma capacité à travailler de manière autonome tout en restant connecté et aligné sur les objectifs de l'entreprise.

Pendant mon séjour chez FeverTokens, j'ai eu l'occasion d'acquérir des enseignements essentiels sur le secteur dynamique de la blockchain et du web3. J'ai pu explorer de près comment FeverTokens offre un logiciel en tant que service (SAAS) permettant aux utilisateurs de déployer des contrats intelligents et de créer leurs propres marketplaces sans avoir besoin de compétences en codage. Cette approche novatrice démocratise l'accès à la technologie blockchain en ouvrant des opportunités d'innovation à un public plus large.

J'ai appris comment FeverTokens permet aux utilisateurs de créer des collections personnalisées et de les proposer comme offres dans leurs propres marketplaces. Cette expérience m'a non seulement permis de comprendre comment la technologie web3 élimine les obstacles à la création d'applications décentralisées, mais elle a également renforcé ma vision de la manière dont les outils sans code peuvent révolutionner la façon dont nous interagissons avec la blockchain.

Durant cette période, j'ai eu l'opportunité privilégiée de travailler sous la tutelle de Monsieur Youssef Meghzaz, un ingénieur informatique aguerri qui occupe les postes de Chief Product Officer et Senior Web3 Engineer chez FeverTokens. Sa direction experte m'a permis d'acquérir une compréhension approfondie du développement full-stack et de son adaptation aux exigences spécifiques de la technologie blockchain. En tant que maître de stage,

M. Meghzaz m'a guidé à travers l'élaboration de fonctionnalités complexes et l'optimisation des performances des applications web3.

À travers l'encadrement de M. Meghzaz, j'ai pu participer activement à la création d'interfaces utilisateur intuitives et fonctionnelles pour la plateforme FeverTokens. J'ai également été exposé aux intrications des interactions avec les smart contracts sur la blockchain, renforçant ainsi ma compréhension des mécanismes sous-jacents qui sous-tendent l'écosystème web3.

Cette expérience m'a également permis d'affiner mes compétences en développement full-stack, d'explorer des technologies clés telles que Next.js et DynamoDB, et de concevoir des fonctions Lambda pour des fonctionnalités spécifiques

.**Paragraphe 3 : Sujet de stage**

Mon stage au sein de FeverTokens s'est concentré sur le développement d'une application Web 3.0 avec un modèle de Backend hybride Cloud/Blockchain. Au sein d'une équipe collaborative de grande envergure, j'ai participé activement au projet principal de FeverTokens en tant que développeur Fullstack. Ma contribution a englobé une multitude de tâches qui ont touché divers aspects de l'application, incluant la sécurité, le Backend, le Frontend et la dimension Web3.

J'ai joué un rôle essentiel dans le projet emblématique de FeverTokens en concevant, développant et implémentant des fonctionnalités clés. Ma position en tant que développeur Fullstack m'a offert la possibilité de travailler sur plusieurs fronts, de la sécurisation des transactions et des données dans le Backend à l'amélioration de l'expérience utilisateur dans le Frontend. J'ai également intégré les fondements fondamentaux de la technologie Web3 dans le cœur même de l'application.

Pour notre méthodologie de travail, nous avons adopté une approche suivie par de grandes entreprises technologiques. Le processus s'est articulé autour de la collaboration continue au sein de l'équipe. Nous avons utilisé des méthodes modernes de gestion de projet et de développement, en mettant l'accent sur la communication fluide et l'intégration harmonieuse des contributions individuelles. Les pratiques de contrôle de version ont été centrales, en favorisant une meilleure traçabilité et une gestion transparente des changements. Cette approche a permis de garantir une qualité constante et une progression efficace tout au long du projet.

Cette expérience m'a plongé au cœur des enjeux et des perspectives du développement d'applications Web 3.0. J'ai travaillé en étroite collaboration avec des professionnels expérimentés, apportant des solutions techniques novatrices et participant au déploiement réussi d'une application qui fusionne les domaines du Cloud et de la Blockchain.

**Paragraphe 4 : La description des composantes du rapport**

Ce rapport est divisé en plusieurs chapitres, chacun visant à explorer des aspects spécifiques de mon expérience de stage chez FeverTokens. Le premier chapitre, intitulé "Contexte du stage", introduit le cadre global du stage en présentant l'entreprise FeverTokens et en mettant en lumière ses activités, son organisation et ses systèmes de production/services. Le deuxième chapitre, "Etude bibliographique", se penche sur les outils et méthodes utilisés pendant le stage. Enfin, le troisième chapitre, "Développement & Analyse", aborde la phase de développement proprement dite ainsi que l'analyse de l'existant. Le rapport se conclut par une "Conclusion générale" qui synthétise les principaux enseignements et les accomplissements du stage, suivie d'une bibliographie et d'éventuelles annexes.

# 

# CHAPITRE 1 :

# Contexte du stage

1. **INTRODUCTION**



Figure logo entreprise FeverTokens

Le choix de l'entreprise FeverTokens pour mon stage s'est avéré une décision des plus éclairées, soutenue par divers arguments solides. En premier lieu, l'orientation novatrice de FeverTokens dans le domaine de la technologie blockchain et des applications Web3 a instantanément captivé mon intérêt. L'opportunité de plonger dans ce secteur en pleine croissance était un vecteur d'apprentissage exceptionnel, susceptible de m'immerger dans des compétences avant-gardistes et de me familiariser avec des technologies émergentes, notamment l'IPFS et le Millisearch.

La réputation solide de FeverTokens en tant qu'acteur majeur de la blockchain a également été un facteur déterminant. L'entreprise a bâti sa renommée en proposant des solutions novatrices, notamment une plateforme sans code pour le développement d'applications Web3.Ce focus sur la simplification des processus de développement tout en restant à la pointe des technologies promettait de fournir une expérience d'apprentissage exceptionnelle.

L'environnement collaboratif au sein de FeverTokens a également été un élément attractif. La possibilité de travailler avec une équipe multidisciplinaire et de collaborer avec des professionnels chevronnés tels que Monsieur Youssef Meghzaz et le Docteur Mohammed Toub a ouvert des perspectives enrichissantes. La variété des tâches, du développement Fullstack à l'interaction avec la blockchain en passant par des technologies comme l'IPFS et le Millisearch, m'a offert une opportunité unique d'explorer différents aspects du processus de création d'applications Web3.

De plus, le caractère pratique de mon stage chez FeverTokens était un argument incontournable. Participer activement au développement d'une application Web3 concrète, en mettant en œuvre des technologies telles que Next.js, DynamoDB, les smartes contracts, l'IPFS et le Millisearch, a été une expérience d'apprentissage incroyablement enrichissante. Cette immersion dans des projets concrets a largement contribué à solidifier mes compétences et à consolider mes connaissances théoriques.

La possibilité de travailler à distance au sein d'une équipe structurée, utilisant des méthodes modernes de gestion de projet telles que Git, a également renforcé mon autonomie et ma capacité à gérer efficacement mes tâches. De plus, si l'on ajoute le mentorat exceptionnel de professionnels tels que Monsieur Youssef Meghzaz et le Docteur Mohammed Toub, ma croissance professionnelle était garantie.

En somme, l'entreprise FeverTokens incarnait l'occasion parfaite d'associer mon apprentissage académique à une expérience pratique dans un domaine technologique en évolution rapide. L'ensemble de ces facteurs combinés a fait de FeverTokens le choix indiscutable pour mon stage, aboutissant à une expérience d'apprentissage exceptionnelle, productive et formatrice.

1. **PRESENTATION DE L’ENTREPRISE**

### Activités

FeverTokens se distingue par ses activités centrées sur la technologie blockchain et les applications Web3, et a également marqué sa présence dans des compétitions de renom, démontrant ainsi son expertise et son engagement envers l'innovation. En tant que participant clé au CAST CHALLENGE, une compétition de grande envergure initiée par des institutions financières et technologiques de premier plan, dont ABN-AMRO, AXA Investment Managers, Banco Santander, Crédit Agricole CIB, Société Générale-FORGE et VISA, FeverTokens a clairement prouvé sa position en tant qu'acteur influent dans le secteur. Le CAST CHALLENGE vise à développer des solutions open-source compatibles avec le cadre CAST pour les marchés de Security Tokens, en répondant aux défis du secteur financier et en promouvant l'adoption de la technologie blockchain à un niveau de qualité bancaire.

FeverTokens a brillamment remporté la première place dans cette compétition, mettant en avant son expertise en matière de développement innovant dans le domaine des actifs numériques. Cette victoire illustre la capacité de l'entreprise à concevoir et à implémenter des solutions répondant aux besoins complexes du marché des Security Tokens. De plus, cela démontre l'engagement continu de FeverTokens envers la technologie blockchain et sa capacité à relever les défis posés par l'industrie financière moderne.

En remportant le CAST CHALLENGE, FeverTokens a contribué à renforcer les liens entre les marchés financiers traditionnels et l'écosystème des actifs numériques en pleine croissance. Cette participation et cette victoire témoignent de l'approche proactive et collaborative de FeverTokens, qui vise à stimuler l'innovation et à explorer de nouvelles opportunités dans le secteur de la blockchain. La réussite de FeverTokens au CAST CHALLENGE ajoute une couche supplémentaire à sa réputation d'innovateur de premier plan et souligne sa volonté de rester à la pointe de la technologie blockchain, en contribuant activement au développement du secteur financier à l'ère numérique.

Au sein de l'environnement dynamique de FeverTokens, la culture de l'apprentissage et du partage de connaissances est un élément central. Chaque semaine, deux stagiaires, employés ou membres du personnel se rassemblent pour organiser des présentations et des ateliers afin de partager leur expertise et d'approfondir leur compréhension des domaines liés au projet. Cette approche collaborative renforce la diversité des compétences au sein de l'équipe et favorise l'enrichissement mutuel, créant ainsi un écosystème propice à l'innovation.

J'ai eu la chance de contribuer à cet échange de connaissances en donnant une présentation approfondie sur la sécurité des applications. Dans le cadre de cette présentation, j'ai abordé en détail les concepts essentiels de la sécurité des applications, en mettant l'accent sur les défis spécifiques rencontrés dans notre projet. J'ai fourni des informations détaillées sur les meilleures pratiques en matière de sécurité, en mettant en évidence les mesures de protection, les stratégies d'authentification et d'autorisation, ainsi que les techniques de prévention des attaques courantes.

Grâce à cette initiative de partage de connaissances, chaque membre de l'équipe a eu l'occasion de renforcer ses compétences et d'acquérir des connaissances approfondies dans des domaines clés du développement Web3. Ces sessions hebdomadaires ont non seulement permis de mieux comprendre les complexités techniques, mais ont également favorisé l'échange d'idées et la résolution collaborative de problèmes. Cette culture du partage de connaissances au sein de FeverTokens reflète l'engagement de l'entreprise à soutenir le développement professionnel de ses membres et à favoriser une approche d'apprentissage continu.

### Organisation

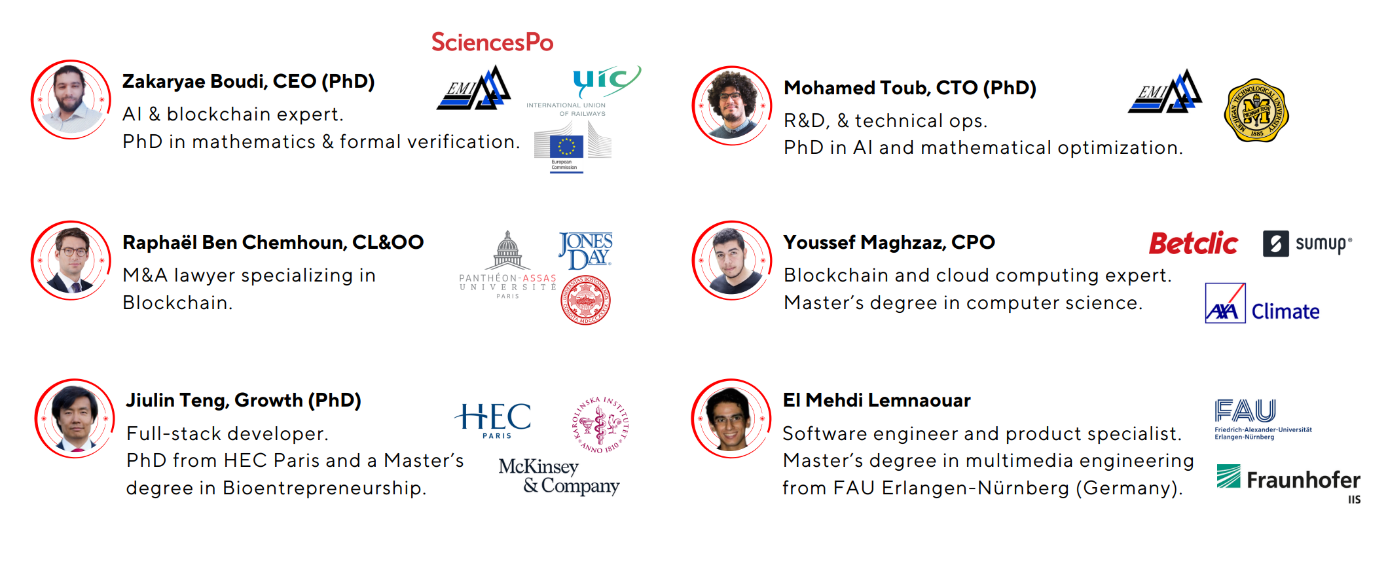


Figure hiérarchie de FeverTokens

### Systèmes de production/service

Au cœur de l'organisation opérationnelle de FeverTokens réside un système méthodique et orienté vers l'atteinte des objectifs, créant ainsi un environnement professionnel propice à la collaboration et à la transparence. Chaque jour, une réunion "standup" rassemble les membres de l'équipe pour discuter des éventuels obstacles rencontrés, partager les accomplissements de la journée précédente et exposer les objectifs clés à réaliser au cours de la journée en cours. Cette pratique assure une communication régulière, permettant la résolution agile des problèmes émergents.

De manière complémentaire, l'équipe se réunit chaque vendredi pour un "end sprint meeting", une session hebdomadaire destinée à évaluer les réalisations de la semaine, encourager les échanges d'idées et formuler des solutions pour les défis identifiés. Cette réunion permet de tirer des enseignements de la semaine écoulée, d'analyser les succès et les échecs, et de planifier de manière stratégique les actions à entreprendre dans la période à venir. Cette approche favorise la clarté des objectifs et la cohésion d'équipe.

Dans le cadre de la gestion de projets, FeverTokens adopte Github Projects, une plateforme essentielle qui offre une vision exhaustive des différentes tâches et initiatives en cours. Chaque membre de l'équipe sélectionne des "tickets" ou tâches en fonction de ses compétences et de son expertise. Ce système permet aux collaborateurs de suivre l'évolution de leurs tâches individuelles, tandis que les superviseurs peuvent assurer un suivi précis et en temps réel de l'état de chaque projet. Les "tickets" sont étiquetés en fonction de leur progression, tels que "en cours", "en test", "en revue" ou "bloqué", ce qui facilite la compréhension immédiate de l'état de chaque projet et la résolution proactive des éventuels obstacles.

Dans l'ensemble, cette méthodologie opérationnelle, soutenue par des réunions structurées et l'utilisation stratégique de Github Projects, illustre le professionnalisme et l'efficacité de l'approche de FeverTokens en matière de gestion de projets. En favorisant la communication, la traçabilité des tâches et la collaboration proactive, cette méthodologie garantit une mise en œuvre cohérente des objectifs et une réponse agile aux défis qui se présentent.

1. **THEMATIQUE**

Au cours de mon stage au sein de FeverTokens, j'ai été immergé dans le développement d'une plateforme innovante mettant en œuvre une architecture full stack Next.js pour le front-end et Node.js pour le back-end. Cette plateforme a été conçue de manière à tirer parti d'un système hybride, combinant l'efficacité du cloud computing avec la sécurité et la décentralisation de la blockchain.

Mon projet principal a consisté à concevoir et développer une plateforme Web robuste et hautement performante. J'ai été impliqué dans toutes les phases du développement, de la conception initiale à la mise en œuvre finale. En utilisant Next.js pour le front-end, j'ai pu créer des interfaces utilisateur interactives et réactives, offrant aux utilisateurs une expérience fluide et intuitive.

L'intégration du système hybride cloud et blockchain a été une composante essentielle de mon projet. J'ai travaillé sur la conception et la mise en œuvre de mécanismes qui permettent une interaction transparente entre les composants hébergés dans le cloud et ceux utilisant la blockchain. Cette approche a permis d'optimiser les performances tout en garantissant la sécurité et la traçabilité des transactions.

Dans un souci d'amélioration continue, j'ai également eu l'opportunité d'explorer des technologies de pointe telles que Web3.js et Ether.js pour interagir avec la blockchain. Ces compétences m'ont permis d'intégrer des fonctionnalités blockchain dans la plateforme, ouvrant la voie à des cas d'utilisation innovants et sécurisés.

Tout au long de ce projet, j'ai travaillé en étroite collaboration avec une équipe multidisciplinaire, échangeant des idées, résolvant des défis techniques et partageant des connaissances pour assurer la réussite du projet. Mon expérience de stage chez FeverTokens m'a non seulement permis d'approfondir mes compétences en développement full stack, mais aussi de comprendre l'importance stratégique de l'hybridation des technologies pour relever les défis de l'avenir.

1. **CONCLUSION**

En guise de conclusion de ce chapitre, il devient évident que l'étude minutieuse de la thématique abordée est d'une importance capitale. Tout au long de ce stage au sein de FeverTokens, j'ai eu l'occasion de plonger profondément dans les aspects pratiques de l'architecture hybride cloud-blockchain. Cette expérience m'a ouvert les yeux sur les vastes opportunités qu'offrent ces technologies émergentes dans le domaine du développement logiciel

# CHAPITRE 2 :

# Etude bibliographique

1. **INTRODUCTION**

Mon stage chez FeverTokens m'a plongé dans un environnement technologique riche, où j'ai travaillé sur le développement complet d'une plateforme en utilisant Next.js et Node.js, tout en fusionnant habilement un système hybride cloud et la puissance de la blockchain. Ce mélange de technologies a été rendu possible grâce à une sélection minutieuse d'outils de pointe, chacun jouant un rôle spécifique dans la réalisation de cette plateforme innovante

Dans cette section introductive à l'étude bibliographique, je vais vous présenter ces outils et technologies clés qui ont été les piliers de notre projet. Chaque choix a été guidé par la volonté de résoudre des problèmes spécifiques et de surmonter des défis techniques, le tout en contribuant à la création d'une plateforme fonctionnelle et performante. Dans le paragraphe suivant, je vais explorer en détail comment ces outils ont collaboré, en révélant comment ils ont été utilisés pour gérer le projet, construire l'interface utilisateur, garantir la sécurité et interagir avec la blockchain.

De plus, nous allons également aborder les méthodes de travail qui ont été essentielles pour la réussite de ce projet. Le processus de développement a été structuré pour garantir une communication fluide et une coordination efficace entre les membres de l'équipe, malgré le caractère virtuel de notre collaboration. Des réunions quotidiennes de stand-up et des réunions hebdomadaires de clôture de sprint ont permis de discuter des problèmes rencontrés, de suivre les avancées et de proposer des solutions. L'utilisation de la plateforme GitHub Projects a facilité la gestion des tâches, tandis que les branches Git et les pull requests ont permis de maintenir un flux de travail ordonné et vérifié.

En somme, cette introduction à l'étude bibliographique et à la méthodologie donne un aperçu des outils technologiques et des méthodes de travail qui ont été au cœur de mon expérience de stage, jetant ainsi les bases de ce que nous explorerons en détail dans les sections suivantes.

1. **OUTILS ET METHODES**

### OUTILS

#### GitHub Projets et Git Avancé : Un Pilotage Précis de Projet



Figure logos Git et Github

Pour assurer une gestion de projet efficace, j'ai tiré parti de Github Projects. Cette plateforme de gestion m'a permis de planifier, suivre et gérer les différentes tâches du projet de manière structurée. L'utilisation des options avancées de Github, en particulier les pull requests, a grandement facilité la collaboration au sein de l'équipe en permettant des revues de code efficaces et une validation rigoureuse des changements. Le modèle de branches de Git a joué un rôle central dans la gestion du développement parallèle, offrant un contrôle précis sur les versions et les modifications.

#### Visual Studio Code : Environnement de Développement Puissant

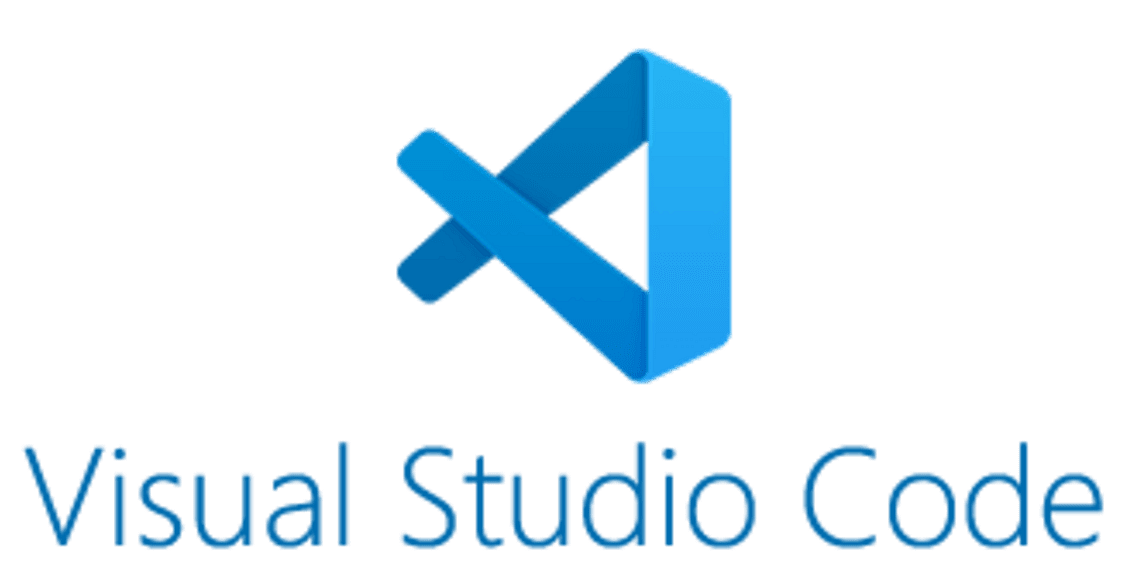


Figure logo Editeur de text utiliser

Pour le développement front-end, j'ai exploité les fonctionnalités avancées de Visual Studio Code. Cet éditeur de code polyvalent a considérablement facilité la rédaction, la gestion et le débogage du code source. Son écosystème d'extensions a été utilisé pour améliorer la productivité et personnaliser l'environnement selon les besoins du projet.

#### GitHub - hoangvvo/nextjs-mongodb-app: A Next.js and MongoDB web application, designed with simplicity for learning and real-world applicability in mind.Next.js : Framework TypeScript pour le Développement

Figure logo technologie FrontEnd (Nextjs)

Next.js a été le cœur du développement, à la fois pour le front-end et le back-end. Ce framework Typescript/JavaScript moderne offre une intégration transparente avec Node.js, permettant de créer des interfaces utilisateur réactives et optimisées. L'approche de rendu côté serveur (SSR) de Next.js a grandement amélioré les performances en assurant un chargement rapide des pages, tout en offrant une expérience utilisateur fluide et réactive.

#### AWS Lambda - Compétences - GusLyon.frServerless avec AWS Lambda : Efficacité et Évolutivité

Figure Logo Service Serverless utiliser

Le modèle serverless a été privilégié pour le développement backend en utilisant les fonctions AWS Lambda. Ce choix a permis de maximiser l'efficacité et l'évolutivité de la plateforme en assurant une réponse rapide aux demandes des utilisateurs. Grâce à AWS Lambda, chaque fonction peut être exécutée indépendamment, permettant une meilleure utilisation des ressources et une réduction des coûts.Exploration de la Blockchain avec Web3.js et Ether.js

#### DynamoDB : Gestion Évolutive de Données



Figure logo base de donnee DynamoDB

Pour la gestion des données, j'ai opté pour DynamoDB, une base de données NoSQL hébergée sur AWS. Cette technologie offre une évolutivité horizontale, permettant de stocker et de récupérer des données rapidement, même à grande échelle. DynamoDB a été utilisé pour stocker diverses informations, des clés publiques des utilisateurs aux détails des transactions, garantissant une performance constante.

#### Exploration de la Blockchain avec Web3.js et Ether.js



Figure Logo librairie EtherJs

Une composante incontournable du projet a été l'intégration de la blockchain. J'ai exploité les bibliothèques Web3.js et Ether.js pour interagir avec la blockchain, notamment pour la génération d'interfaces binaires d'application (ABI). Ces bibliothèques ont été essentielles pour traduire les interactions complexes de la blockchain en fonctionnalités conviviales pour les utilisateurs.

#### HardHat



Figure Logo Librairie testing HardHat

Dans le cadre de mon stage en génie informatique, j'ai eu l'opportunité d'explorer une technologie relativement nouvelle et innovante dans le domaine du développement de smart contracts : Hardhat. Cette expérience m'a permis de plonger dans le monde passionnant du développement blockchain et de comprendre comment Hardhat révolutionne le processus de test des contrats intelligents sur la blockchain Ethereum. Hardhat offre une approche moderne et puissante pour simplifier et améliorer le développement de contrats intelligents, en mettant l'accent sur la robustesse, la sécurité et l'efficacité des tests. En tant que stagiaire, j'ai été chargé d'exploiter les fonctionnalités avancées de Hardhat pour concevoir et exécuter des tests unitaires et d'intégration, tout en automatisant des scénarios de test complexes. Cette expérience m'a permis d'être au cœur de l'innovation technologique, en utilisant Hardhat pour garantir la qualité et la sécurité des contrats intelligents dans un environnement en constante évolution.

#### NatSpec

J’ai eu l'opportunité d'explorer une technologie innovante appelée NatSpec, qui s'est révélée essentielle pour la documentation complète des smart contracts sur la blockchain Ethereum. NatSpec a joué un rôle clé dans la création de documentation détaillée pour une variété de smart contracts, cette technologie a grandement simplifié la rédaction de descriptions précises des fonctionnalités, des paramètres et des comportements de ces contrats, en fournissant un format standardisé pour rendre la documentation plus accessible et compréhensible. Grâce à NatSpec, j'ai pu présenter ces contrats de manière cohérente, ce qui a facilité la compréhension de leur utilisation et de leurs interactions au sein de l'écosystème blockchain. NatSpec a donc été un outil précieux pour garantir que nos smart contracts étaient bien documentés, améliorant ainsi leur transparence et leur accessibilité pour les développeurs et les utilisateurs.

### Méthode de travail:

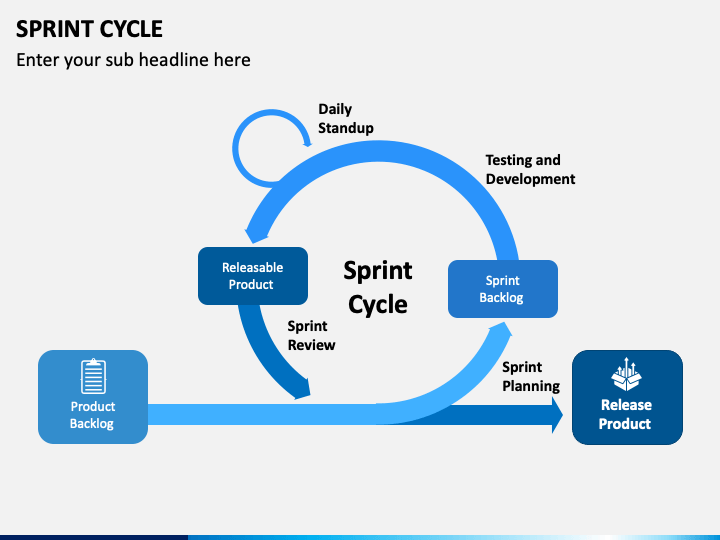


Figure Image explicative a la methode de travail

La méthodologie agile, largement adoptée dans l'industrie informatique moderne, a été le pilier de notre approche de travail chez FeverTokens. Pour assurer une gestion efficace et collaborative de notre projet, nous avons utilisé GitHub comme plateforme centrale pour le suivi et le développement de nos activités.

Chaque étape de notre processus de développement a été encadrée par les principes agiles, avec des itérations courtes appelées "sprints". Ces cycles de travail intenses, d'une à deux semaines, nous ont permis de nous concentrer sur des ensembles spécifiques de tâches, tout en restant flexibles face aux changements et aux défis.

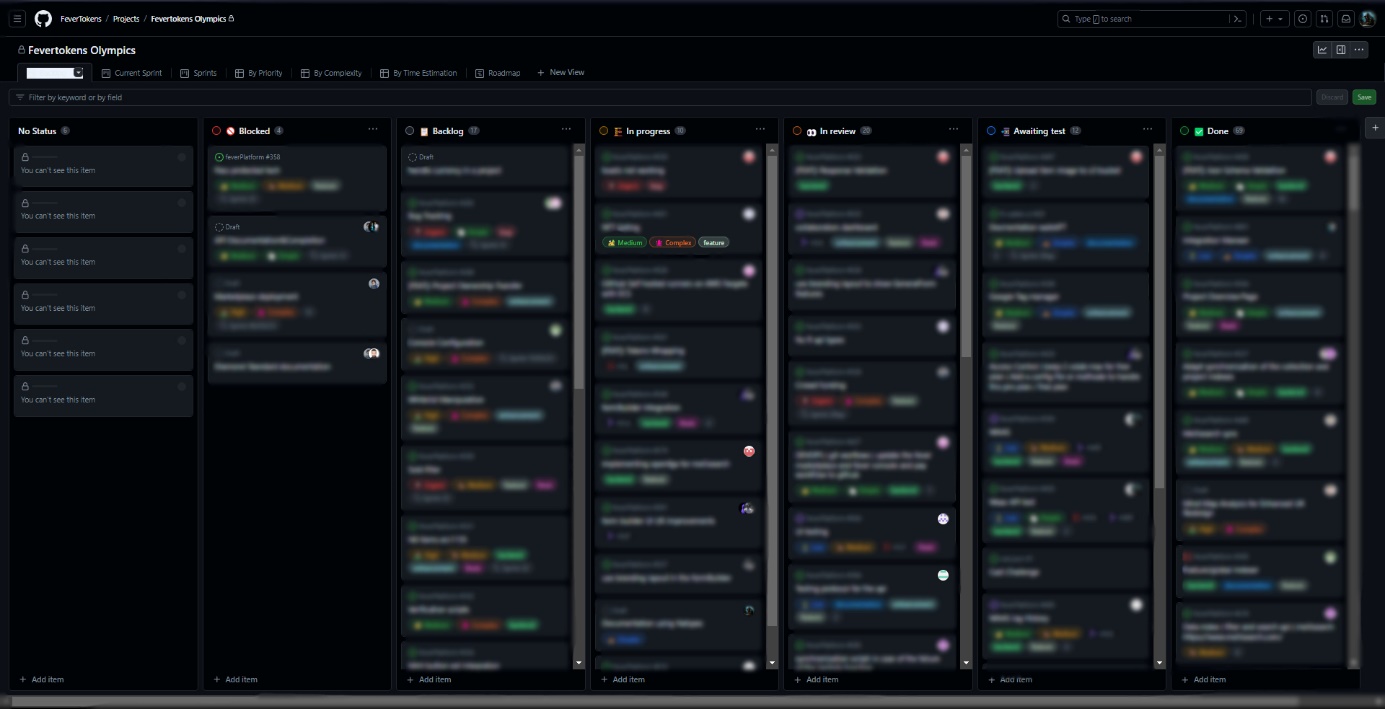


Figure GitHub Project l'outils utiliser pour monitorer le progresse des taches

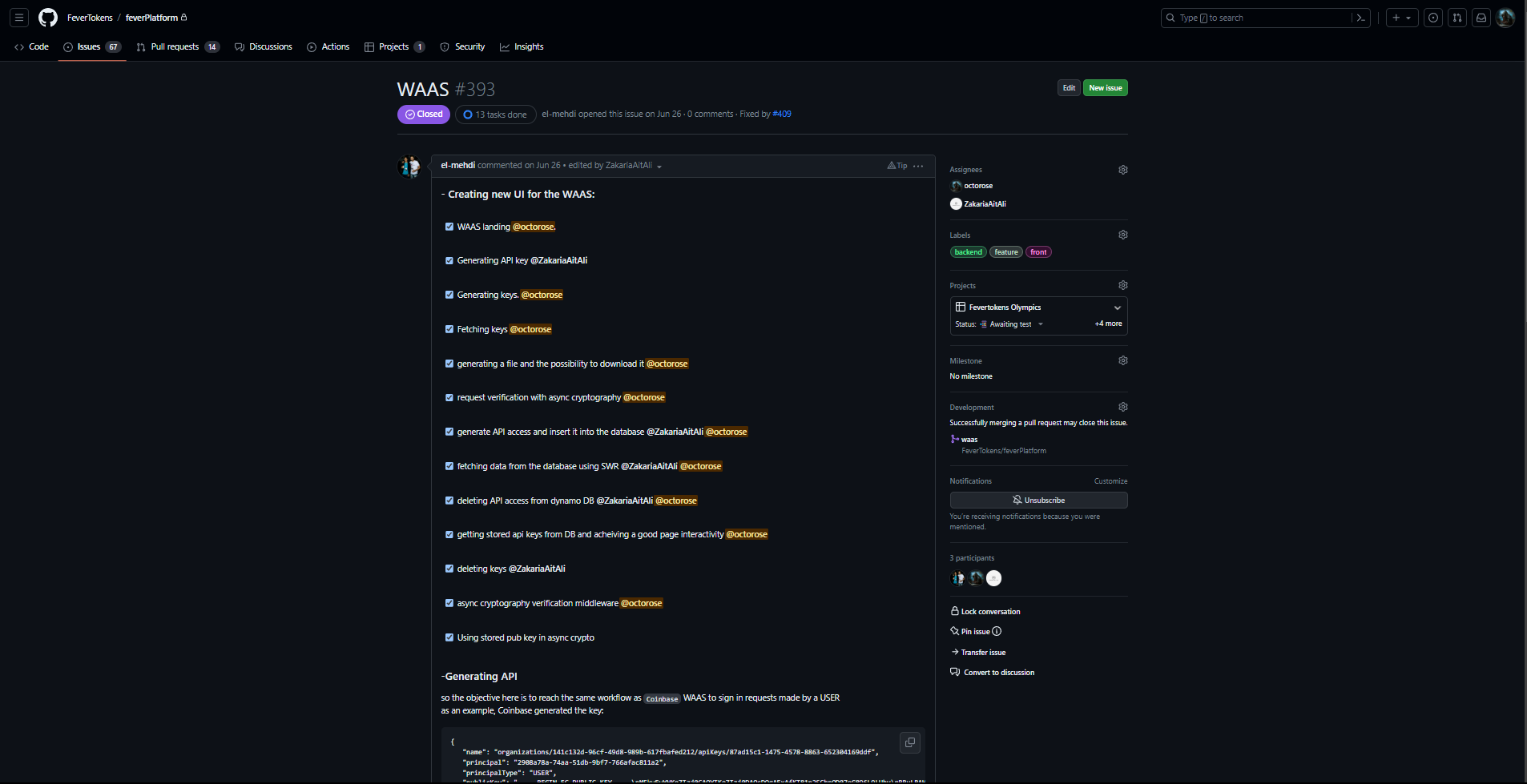


Figure description d'une des taches effectuer

Pour visualiser et gérer nos tâches, nous avons utilisé GitHub Projets, une fonctionnalité qui nous a permis de créer des tableaux Kanban virtuels. Chaque tâche a été représentée par un "ticket", ou une "user story", avec des étiquettes spécifiant leur priorité, leur état et leur complexité, la période estime pour terminer la tâche et la Pull-request dont il ‘est attaché aussi que les personnes qui sont en conception et développement de la tâche. Cela nous a permis d'avoir une vue d'ensemble claire de l'avancement de chaque tâche et de suivre les flux de travail.

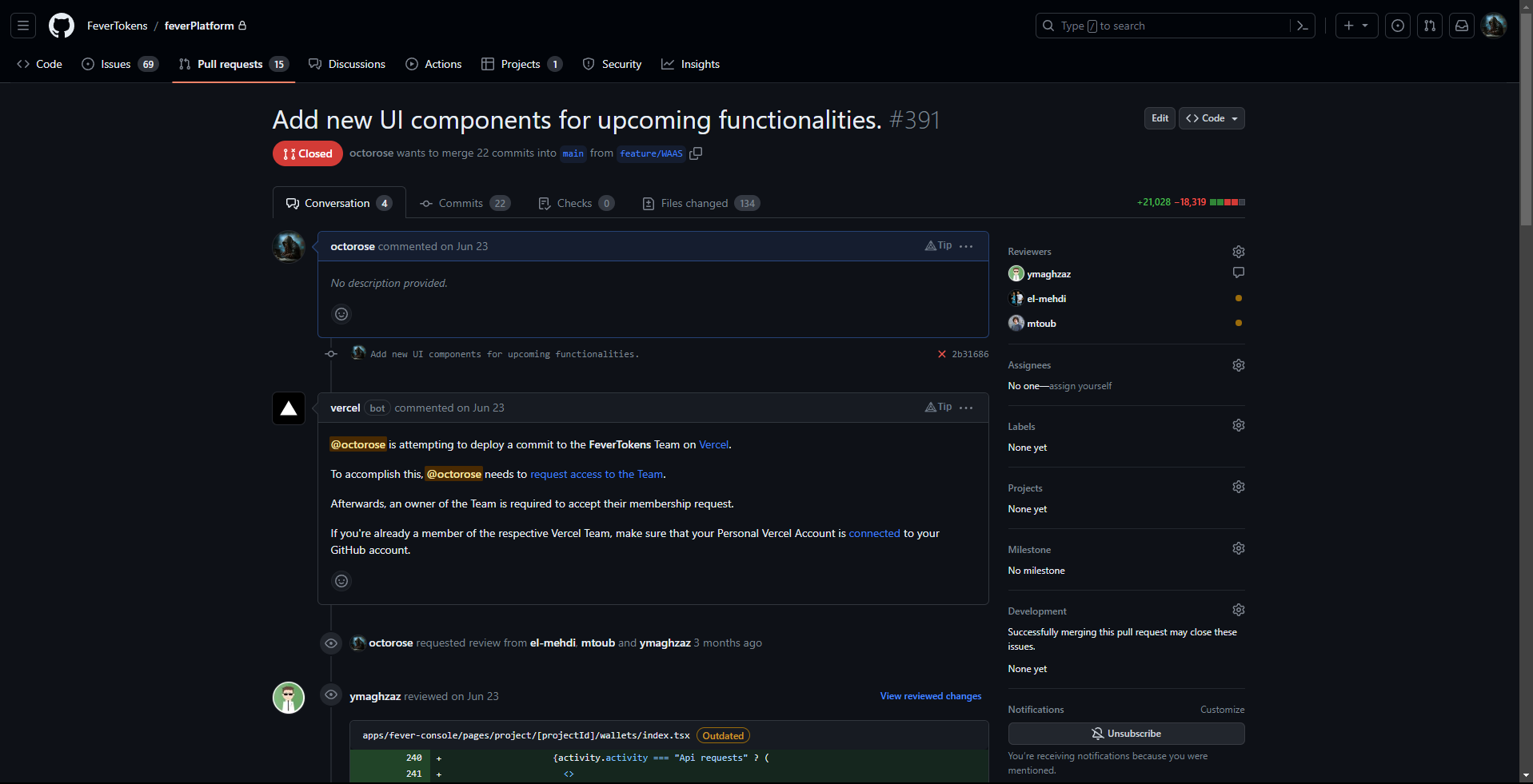


Figure Pull-request verifier et terminer

L'une des pratiques clés qui a grandement favorisé la collaboration au sein de l'équipe lors de mon stage a été l'utilisation des "pull requests". Cette approche s'est avérée essentielle pour maintenir et améliorer la qualité du code tout en bénéficiant de l'expertise des développeurs seniors de l'équipe.

Avant d'intégrer des modifications de code dans la branche principale du projet, chaque membre de l'équipe avait la responsabilité de soumettre une pull request. Ces pull requests étaient ensuite examinées et évaluées par les développeurs seniors, qui apportaient leur expérience et leurs compétences pour garantir que le code respecte les normes de qualité et de sécurité établies.

Cette approche a créé un environnement propice à l'apprentissage et au mentorat, où les développeurs plus expérimentés pouvaient partager leurs connaissances avec les membres plus juniors de l'équipe. Cela a également permis d'identifier et de corriger rapidement les erreurs potentielles tout en favorisant le développement professionnel de l'équipe.

Non seulement cela a renforcé la qualité du code, mais cela a également encouragé la croissance et le développement de compétences au sein de l'équipe, créant ainsi une collaboration dynamique et productive.

Les réunions quotidiennes de stand-up ont été cruciales pour maintenir la communication fluide entre les membres de l'équipe. Chaque jour, nous nous sommes réunis pour partager nos progrès, identifier les obstacles et coordonner nos efforts. Les réunions de clôture de sprint ont permis de réfléchir aux réalisations, de définir les priorités pour la prochaine itération et d'ajuster notre planification en fonction des besoins changeants du projet.

L'utilisation de Git a été fondamentale pour gérer les versions de notre code. Nous avons créé des branches spécifiques pour chaque fonctionnalité ou amélioration, permettant ainsi de travailler simultanément sur différentes parties du projet. Les branches ont ensuite été fusionnées via des pull requests, ce qui a permis un processus de validation et d'approbation collaboratif.

En adoptant cette méthodologie agile et en utilisant GitHub comme plateforme centrale, nous avons pu gérer de manière transparente notre projet, maintenir une communication ouverte et constante entre les membres de l'équipe et produire un code de haute qualité. Cette approche collaborative a non seulement renforcé notre productivité, mais a également contribué à maintenir un environnement de travail adaptatif et innovant au sein de FeverTokens.

1. **CONCLUSION**

Pour conclure ce chapitre dédié à l'analyse de la méthode de travail au sein de l'entreprise et des technologies employées dans nos principales tâches, nous avons dressé un tableau complet des pratiques actuelles. Nous avons mis en lumière l'importance cruciale de l'adoption de méthodologies agiles pour une gestion transparente et efficace des projets. De plus, nous avons exploré en détail les technologies de pointe qui ont facilité notre travail au quotidien.

Néanmoins, il est important de noter que cette analyse reste une vue d'ensemble de nos méthodes de travail et de nos outils technologiques. Dans les chapitres à venir, nous plongerons davantage dans les détails, en examinant comment ces méthodes et ces technologies ont été appliquées dans des situations spécifiques, ainsi que les résultats obtenus. Nous discuterons également des défis rencontrés et des solutions mises en œuvre pour les surmonter.

En somme, ce chapitre a posé les bases de notre compréhension des pratiques de travail au sein de l'entreprise et de l'intégration des technologies. Il nous a fourni un contexte essentiel pour la suite de notre rapport, où nous approfondirons ces sujets et les relierons à notre expérience personnelle au sein de l'entrepris

# 

# CHAPITRE 3 :

# Développement & Analyse

1. **INTRODUCTION**

Dans ce nouveau chapitre, nous allons plonger plus en profondeur dans les différentes tâches que j'ai exécutées au cours de mon stage chez FeverTokens. Avant d'entrer dans les détails de chaque tâche, permettez-moi de récapituler la thématique de mon stage et d'introduire les différentes parties de ce chapitre.

La thématique de mon stage chez FeverTokens est centrée sur le développement d'une plateforme innovante qui intègre des technologies de pointe, notamment la blockchain, pour créer une solution de gestion d'actifs numériques hautement sécurisée et évolutive. Mon rôle au sein de l'équipe était de contribuer à différentes facettes de ce projet, en mettant l'accent sur l'efficacité, la sécurité et l'accessibilité.

Ce chapitre sera divisé en plusieurs sections, chacune correspondant à une tâche spécifique que j'ai entreprise au cours de mon stage. Voici un aperçu de ces sections :

### WAAS - Wallet as a Service (Portefeuille en tant que Service) :

Dans cette section, nous explorerons en détail le concept de WAAS (Wallet as a Service), une infrastructure de portefeuille cryptographique hautement sécurisée et évolutive. Je vais expliquer comment cette solution de portefeuille numérique offre des options flexibles pour la gestion des actifs numériques, tant du point de vue théorique que pratique. Nous examinerons son rôle essentiel dans notre plateforme.

### Web3 Explorer - Accessibilité à la Blockchain sans Connaissances Préalables :

Cette section se concentrera sur l'outil Web3 Explorer, qui a été intégré dans notre plateforme pour permettre aux utilisateurs non-développeurs d'effectuer des transactions sur la blockchain sans avoir de connaissances préalables. Nous explorerons comment cet outil offre une expérience utilisateur conviviale tout en exploitant les capacités de la blockchain.

### Documentation des Smart Contracts avec Natspec :

La documentation des Smart Contracts est cruciale pour leur compréhension et leur utilisation. Ici, je vais expliquer comment j'ai utilisé la technologie Natspec pour documenter nos Smart Contracts de manière théorique et applicative, facilitant ainsi leur intégration et leur utilisation par les développeurs.

### Tests et Débogage de l'Application :

Le processus de test et de débogage est essentiel pour garantir le bon fonctionnement de notre plateforme. Je vais décrire comment j'ai effectué des tests approfondis, identifié les bugs et les problèmes, et ensuite travaillé à trouver des solutions et à déboguer ces problèmes pour assurer la qualité de l'application.

Chaque section de ce chapitre plongera dans les détails théoriques et applicatifs des tâches que j'ai accomplies, fournissant ainsi un aperçu complet de mon expérience de stage chez FeverTokens.

1. **DESCRIPTION ET ANALYSE DE L’EXISTANT**

### Description de l’existant

#### ****Bunzz****

Bunzz vous permet de créer facilement des applications web3. Il s'agit d'un contrat intelligent en tant que plateforme de service permettant aux développeurs de créer des DApps rapidement et facilement sans avoir à écrire une ligne de code.

Les développeurs de Bunzz construisent facilement des contrats intelligents pour le développement de DApp et les déploient sur des réseaux de blockchain compatibles avec EVM en une seule étape, sans écrire une ligne de code.

La plateforme simplifie le développement de contrats intelligents qui nécessitent des connaissances avancées en fournissant un SDK facile à mettre en œuvre, ce qui permet aux développeurs de se concentrer davantage sur le travail à valeur ajoutée.

#### ****Thirdweb****

Thirdweb est une plateforme qui fournit une suite d'outils aux créateurs, artistes et entrepreneurs pour construire, lancer et gérer facilement un projet Web3. Elle permet aux utilisateurs d'ajouter des fonctionnalités telles que les NFT, les places de marché et les jetons sociaux à leurs projets Web3 sans avoir à écrire une ligne de code.

(Onyekwere, 2022)

### Points Forts du Projet : Analyse des Avantages Clés

FeverToken présente plusieurs avantages significatifs, en particulier dans le contexte du développement Web3 et de la gestion d'actifs numériques. L'un de ses points forts réside dans sa capacité à simplifier la création de projets Web3. Que ce soit pour la mise en place d'un NFT store, la tokenisation de bout en bout des actifs, ou la création de marchés alimentés par des tokens, FeverToken offre une infrastructure conviviale et adaptable pour répondre aux besoins variés des développeurs.

Un autre avantage clé de FeverToken est sa gestion autonome de la connectivité à la blockchain. La plateforme dispose de ses propres RPC (Remote Procedure Call) pour garantir une disponibilité continue et une connexion stable à la blockchain. Cela signifie que les développeurs travaillant avec FeverToken ont un accès fiable à la blockchain sans avoir à se soucier des problèmes de connectivité externes.

FeverToken propose également son propre ensemble d'API (Application Programming Interface), facilitant ainsi l'intégration de ses fonctionnalités dans diverses applications et systèmes. Cette API personnalisée permet aux développeurs de tirer parti des capacités de FeverToken pour créer des fonctionnalités spécifiques à leurs projets.

Un autre atout majeur est la disponibilité du service "Wallet as a Service" (WaaS) au sein de FeverToken. Cette solution de portefeuille numérique offre aux utilisateurs une manière hautement sécurisée et flexible de gérer leurs actifs numériques, contribuant ainsi à la commodité et à la sécurité des transactions.

Enfin, FeverToken se démarque par son système de stockage et de traitement de données hybride cloud/blockchain. Cette approche innovante permet de combiner l'efficacité du cloud computing avec la sécurité et la décentralisation de la blockchain. Cela garantit une performance optimale tout en préservant l'intégrité des données, ce qui est essentiel pour les applications Web3 et la gestion d'actifs numériques.

En résumé, les avantages clés de FeverToken dans le domaine Web3 incluent sa facilité d'utilisation pour la création de projets, sa gestion autonome de la connectivité à la blockchain, son API personnalisée, le service Wallet as a Service, et son système de stockage de données hybride, tous ces éléments contribuent à faire de FeverToken une plateforme puissante pour le développement et la gestion d'actifs numériques.

1. **DEVELOPPEMENT**

### WAAS (Wallet As A Service)

#### Concept:

WaaS représente une solution financière technologique en mode cloud, offrant aux entreprises la possibilité de proposer à leur clientèle un portefeuille numérique, permettant ainsi le stockage, l'envoi et la réception de fonds.

FeverTokens, propose ce service de manière encore plus étendue à ses clients. En tant que membre de l'équipe qui a contribué à la création de ce service, j'ai eu l'opportunité de travailler sur plusieurs aspects clés, notamment le développement de la partie frontend, la mise en place du backend, et la gestion de la base de données, tout en veillant à ce qu'elle réponde aux besoins spécifiques de nos clients et leur sécurité.

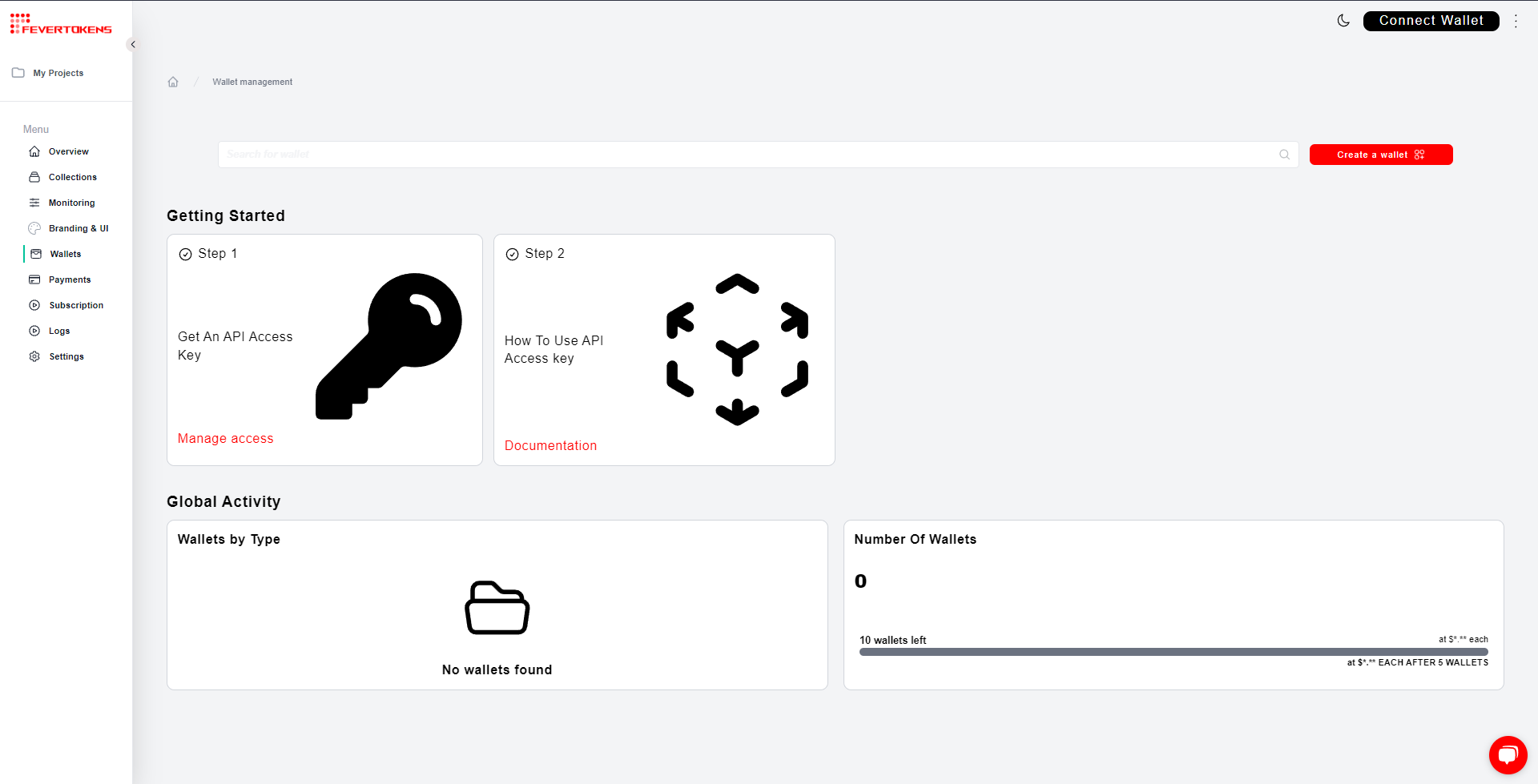


Figure page principale de WAAS

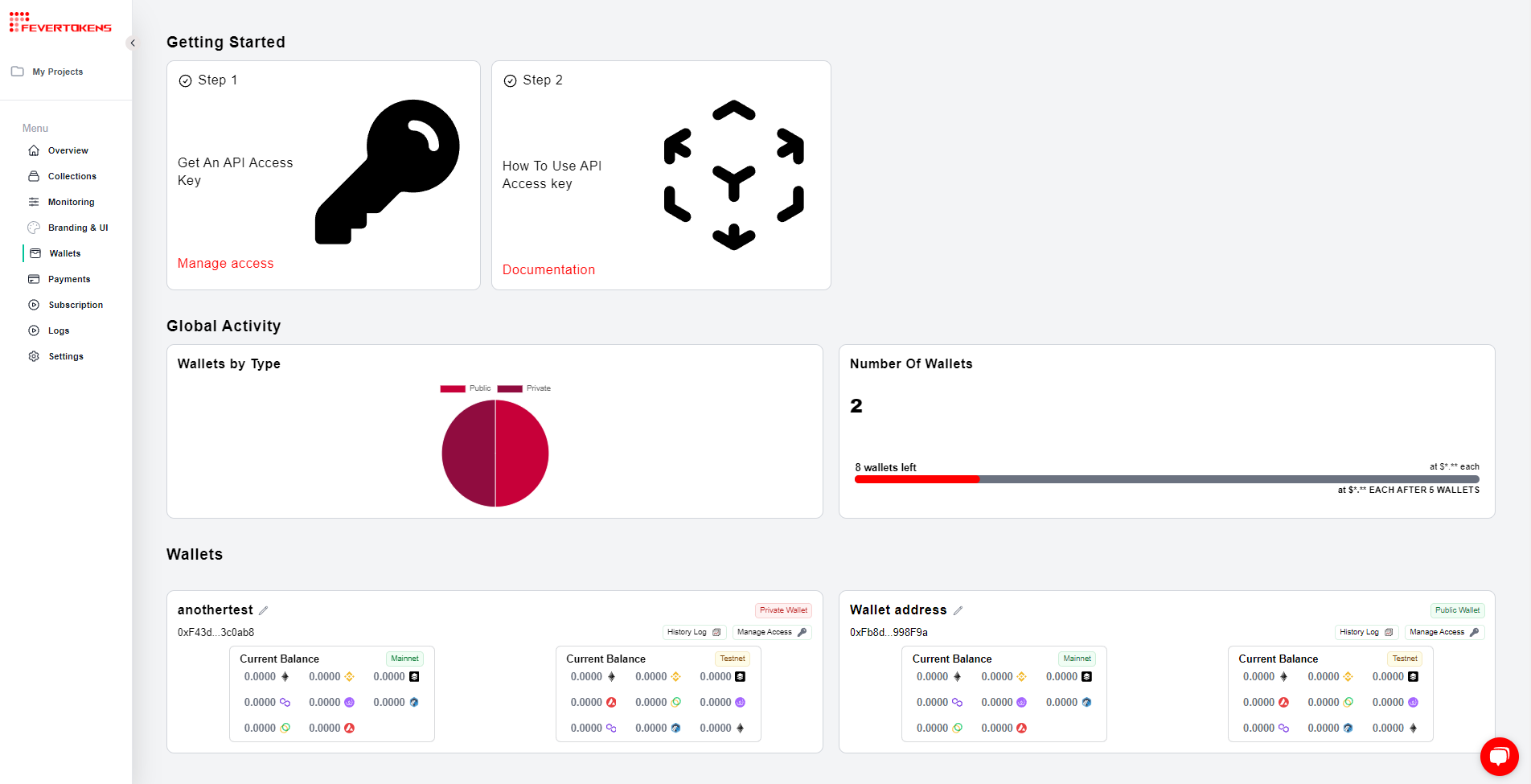


Figure Page qui démontre la dynamiter

Comme vous pouvez le constater, cette interface dynamique est divisée en plusieurs sections, notamment le navbar et la sidebar. Ces deux composants servent d'éléments d'habillage de la page, ce qui signifie qu'ils sont conçus de manière indépendante. Cette approche présente plusieurs avantages, dont la facilité de débogage et la possibilité d'apporter des modifications à ces composants qui se répercutent dans l'ensemble de l'application.

Section “Get an API Access Key”

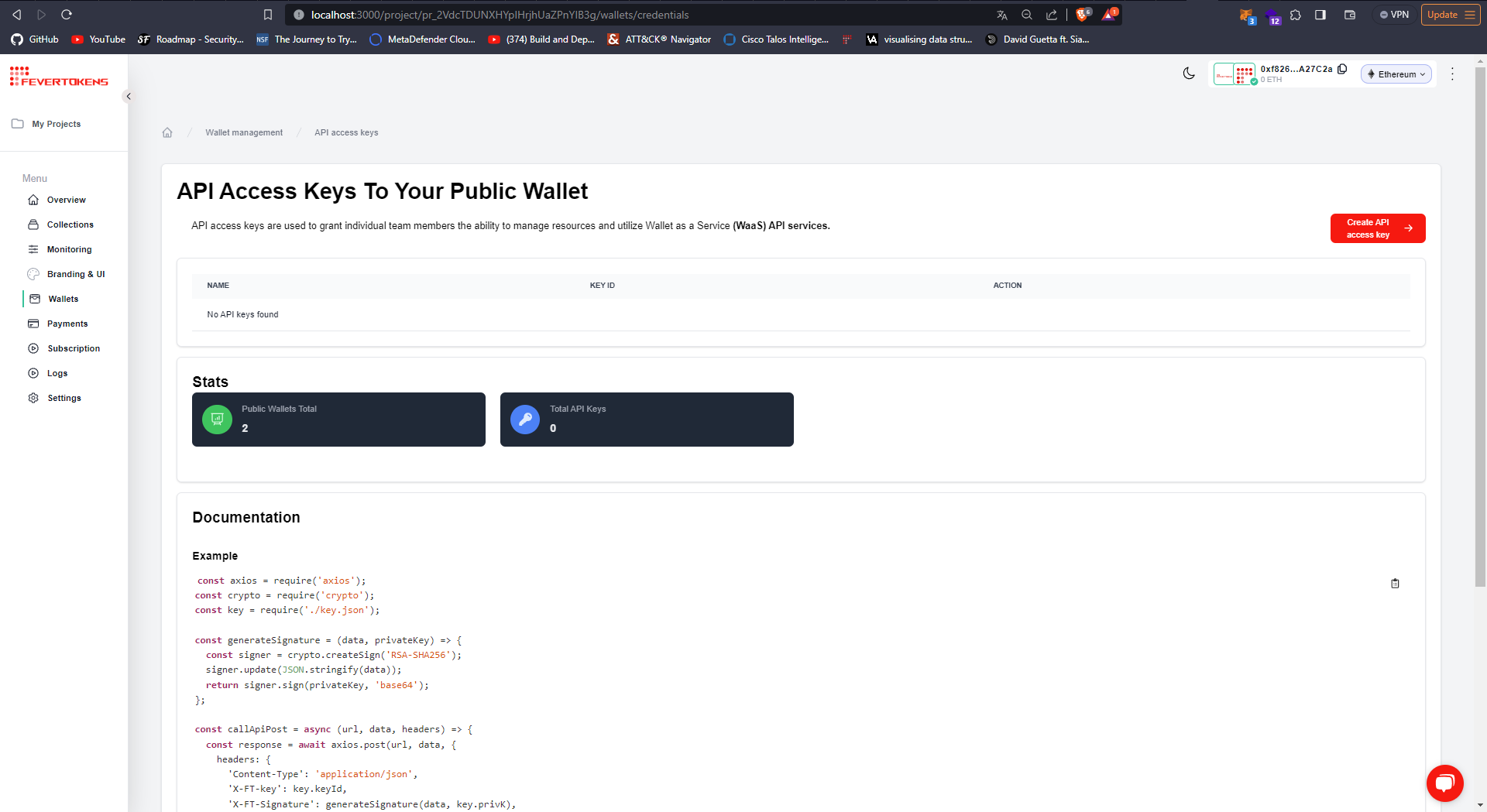


Figure page Manage Access Keys

La génération des clés d'accès pour les portefeuilles, comprenant la clé privée et la clé publique, est réalisée dans un cadre de sécurité optimal. La génération de la clé publique s'effectue côté client, tandis que la création de la clé privée se déroule côté serveur. Cette génération de clés se base sur l'utilisation du module Crypto de Node.js.

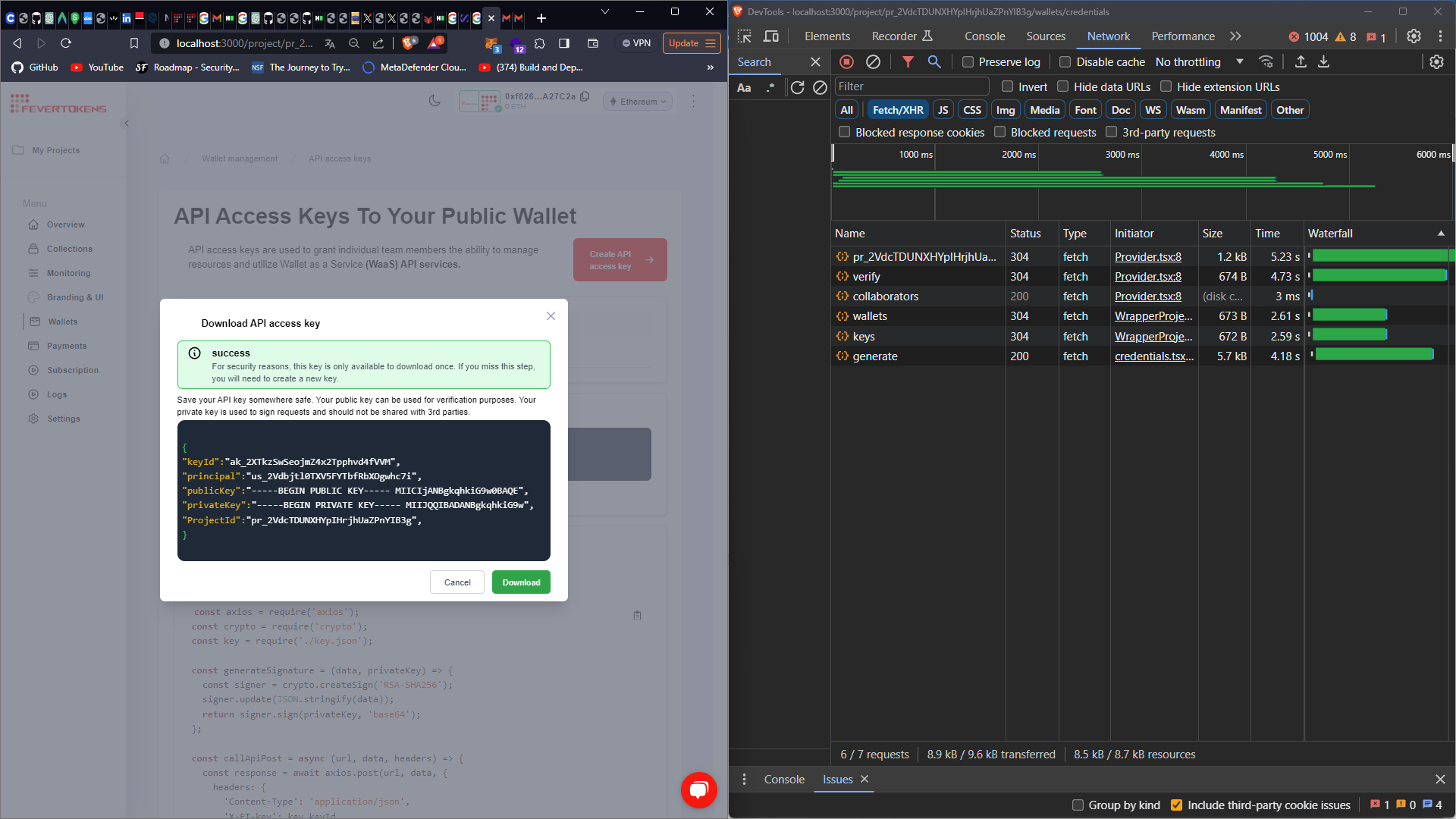


Figure page qui demontre les packets envoyez vers le server

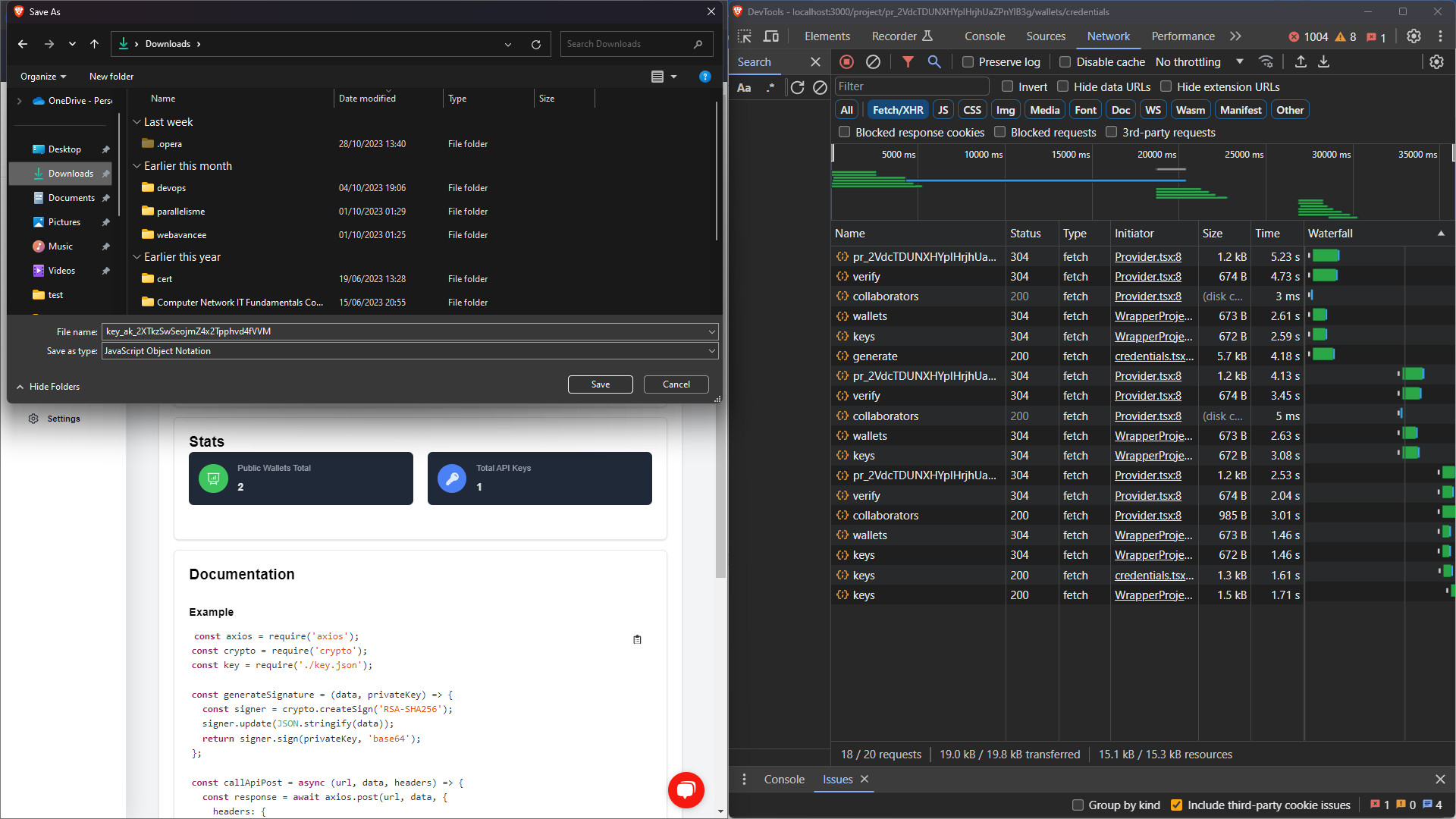


Figure la possibilité de enregistrer le pair de clés en local

De plus, il est essentiel de noter que le transfert des données depuis le client vers le serveur est effectué de manière cryptée, garantissant ainsi la sécurité des informations des clients tout au long du processus.

#### fetchKeys API Call:



Figure code responsable à envoyer la demande de génération de clés

La fonction fetchkeys que vous avez fournie semble être utilisée pour récupérer des clés (keys) depuis une API dans le contexte d'un projet spécifique. Voici une explication plus détaillée des objectifs que vous avez mentionnés :

Sécurité :

* La fonction commence par activer un indicateur de chargement (setLoading(true)) pour indiquer à l'utilisateur que l'opération est en cours.
* Ensuite, elle fait une requête à l'API (fetch) pour récupérer les clés. Si la réponse n'est pas réussie (response.ok est faux), un message d'erreur est généré.
* En cas de succès, les données récupérées sont cryptées (dycrypt(secretPath, data.key)) à l'aide d'une clé spécifique (data.key). Cela ajoute une couche de sécurité pour protéger les données sensibles lors de leur transmission.

Requête API :

* La fonction utilise l'API pour effectuer une demande (fetch) à un chemin spécifique (/api/projects/${projectId}/wallets/keys/generate). Cela suggère qu'il y a un point d'extrémité (endpoint) de l'API qui génère et renvoie les clés associées au projet (projectId).

Contrôle des useState :

* La fonction met à jour l'état (setLoading(true) et setLoading(false)) pour refléter l'état actuel de l'opération. Cela permet à l'interface utilisateur de réagir en conséquence, par exemple, en montrant un indicateur de chargement pendant le processus.

Maintien de l'interactivité avec l'utilisateur :

* L'utilisation de l'indicateur de chargement (setLoading) permet à l'utilisateur de savoir que quelque chose se passe en arrière-plan. Cela aide à maintenir une interaction fluide et à informer l'utilisateur sur le statut de l'opération.

#### API Fetchkeys:

****

Figure API responsale a recevoir la requête de génération et la traiter

Figure :La fonction GET que vous avez fournie semble être une route d'API dans un environnement Next.js (un framework JavaScript/TypeScript côté serveur).

Voici une explication de la fonction en respectant la confidentialité des informations :

Session et Identité de l’Utilisateur :

* La fonction commence par extraire la session de l'utilisateur (const session: any = await getSessionFromContext({ req, res });). La session contient des informations d'identification de l'utilisateur.
* L'ID de l'utilisateur est extrait de la session (const userId = session?.token?.uid;). Cela permet d'associer les clés générées à un utilisateur spécifique.

Génération de Paires de Clés RSA:

* La fonction utilise la bibliothèque crypto pour générer une paire de clés RSA. Il s'agit d'une méthode couramment utilisée pour la cryptographie asymétrique.
* Une clé publique (publicKey) et une clé privée (privateKey) sont générées.

Contrôle de Cache:

* Les en-têtes de réponse sont configurés pour indiquer aux navigateurs de ne pas mettre en cache la réponse. Cela garantit que les clés ne sont pas stockées localement sur le navigateur de l'utilisateur.

Création de la Paire de Clés Cryptées:

* Une paire de clés, composée de la clé publique, de la clé privée, d'un identifiant de clé (keyId), et de l'ID de l'utilisateur (principal), est regroupée dans un objet pairkey.
* Ces informations sont ensuite cryptées (createSecret) avec un chemin secret unique basé sur des données confidentiels

Réponse de l'API:

* La clé cryptée est renvoyée en tant que réponse de l'API (res.status(200).json({ key: cryptedData });). Cela signifie que l'utilisateur recevra uniquement la clé cryptée.

#### Keys API Call:

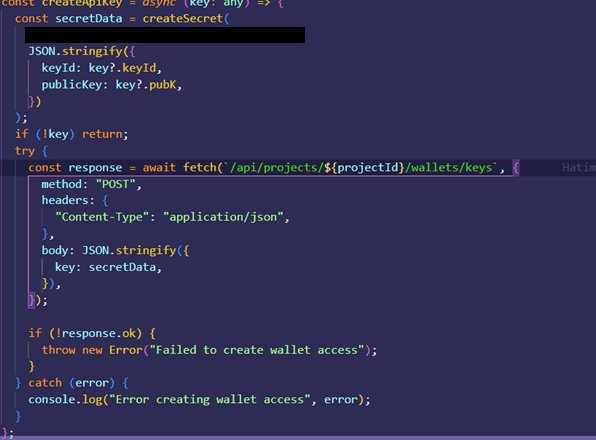


Figure fonction responsable à envoyer la requête pour stocker la clé public

**Explication :**

La fonction que vous avez décrite semble être responsable de l'envoi d'une requête HTTP de type POST à une API. Cette requête contient des données cryptées sous forme de paire clé-valeur, où l'ID et la clé publique sont inclus. Il est important de noter que la clé privée n'est pas incluse dans cette requête, car elle est considérée comme une donnée critique et ne doit pas être transmise à l'API.

**Voici une explication plus détaillée :**

Nature de la requête : Cette fonction est conçue pour envoyer des données à une API via une requête HTTP POST. Les données envoyées comprennent un identifiant (ID) et une clé publique. Ces données sont cryptées pour des raisons de sécurité.

Exclusion de la clé privée : La clé privée n'est pas incluse dans la requête. Cela est fait délibérément car la clé privée est une donnée hautement sensible et doit être gardée confidentielle. Elle ne doit jamais être transmise à l'API ou être exposée à des tiers non autorisés.

Traitement des erreurs de l'API : La fonction prend également en compte la possibilité que l'API puisse renvoyer une erreur en réponse à la requête. Cela suggère que la fonction inclut un mécanisme de gestion des erreurs pour gérer ces situations de manière appropriée.

L'approche de sécurité adoptée, en ne transmettant que la clé publique et en gardant la clé privée confidentielle, est une bonne pratique de sécurité. Cela garantit que seules les informations nécessaires sont partagées avec l'API, tout en préservant la confidentialité des informations sensibles.

#### API Keys :

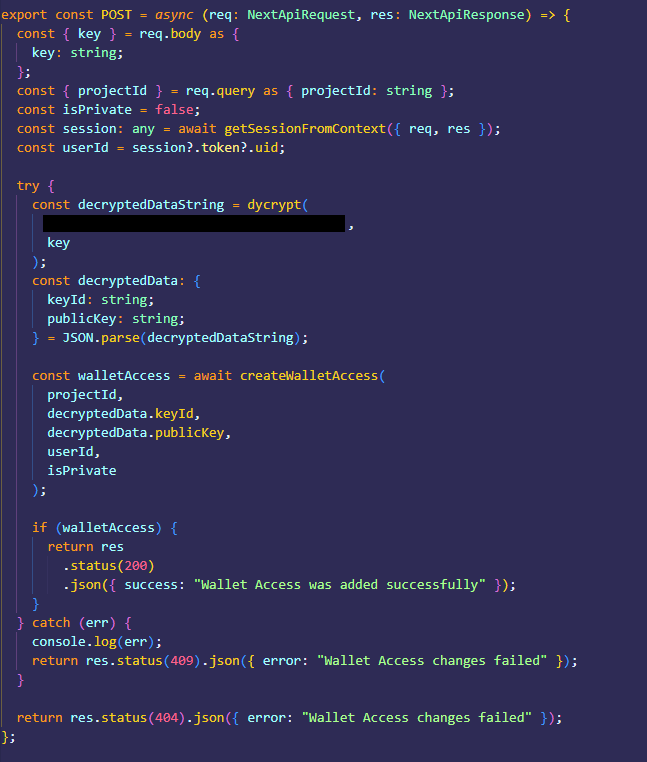
****

Figure traitement de requête de type post pour stocker la clé public

**Le processus détaillé est le suivant :**

Extraction des données de la requête :

* La variable key est extraite du corps de la requête, représentant la clé cryptée fournie par le client.
* Le projectId est extrait de l'URL de la requête pour identifier le projet associé à cette clé.

Préparation des données :

* La variable isPrivate est définie à false, ce qui indique que cette clé appartient à un portefeuille public.

Récupération de la session utilisateur :

* La session de l'utilisateur est récupérée. Cela permet d'associer cette opération à un utilisateur spécifique.

Décryptage de la clé :

* La clé cryptée est déchiffrée en utilisant une chaîne générée à partir de l'ID de l'utilisateur et du projectId.

Parsing des données déchiffrées :

* Les données déchiffrées sont ensuite analysées en tant qu'objet contenant un keyId et une publicKey.

Appel au microservice :

* La fonction createWalletAccess est appelée avec les informations pertinentes, comme le projectId, keyId, publicKey, l'ID de l'utilisateur (userId), et le statut de privacité (isPrivate).

Vérification du succès de l'opération :

* Si l'opération de création d'accès au portefeuille est réussie (c'est-à-dire si walletAccess est évalué comme true), une réponse JSON avec un statut 200 est renvoyée, indiquant que l'accès au portefeuille a été ajouté avec succès.

Gestion des erreurs :

* Si une erreur se produit à tout moment dans le processus, elle est enregistrée dans la console et une réponse JSON avec un statut 409 est renvoyée, signalant un échec dans les modifications de l'accès au portefeuille.

Réponse par défaut :

* Si aucune des conditions ci-dessus n'est remplie, une réponse JSON avec un statut 404 est renvoyée, indiquant que les modifications de l'accès au portefeuille ont échoué.

#### Service CreateWallet :

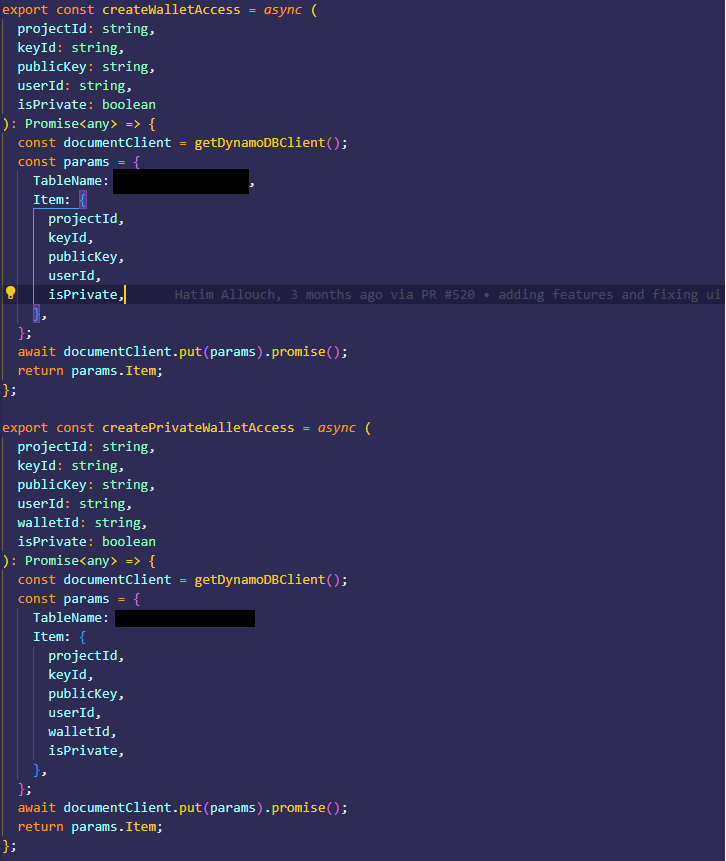
****

Figure la requête pour stocker la clé public dans Dynamo DB

**Explication :**

La fonction createWalletAccess représente un microservice conçu pour simplifier le processus de création d'un accès à un portefeuille électronique au sein de l'application.

Elle prend en charge cinq paramètres essentiels :

* **projectId** : Identifiant du projet lié au portefeuille électronique.
* **keyId** : Identifiant de la clé associée à cet accès.
* **publicKey** : Clé publique utilisée pour l'authentification.
* **userId** : Identifiant de l'utilisateur auquel l'accès est associé.
* **isPrivate** : Un booléen indiquant si le portefeuille électronique est privé ou non.

Ce microservice utilise une instance du client DynamoDB pour interagir avec la base de données NoSQL.

L'opération put est utilisée pour insérer ces informations dans la table DynamoDB.

En cas de succès, le microservice retourne l'objet créé, confirmant ainsi la création de l'accès au portefeuille électronique.

**Explication d’utiliter :**

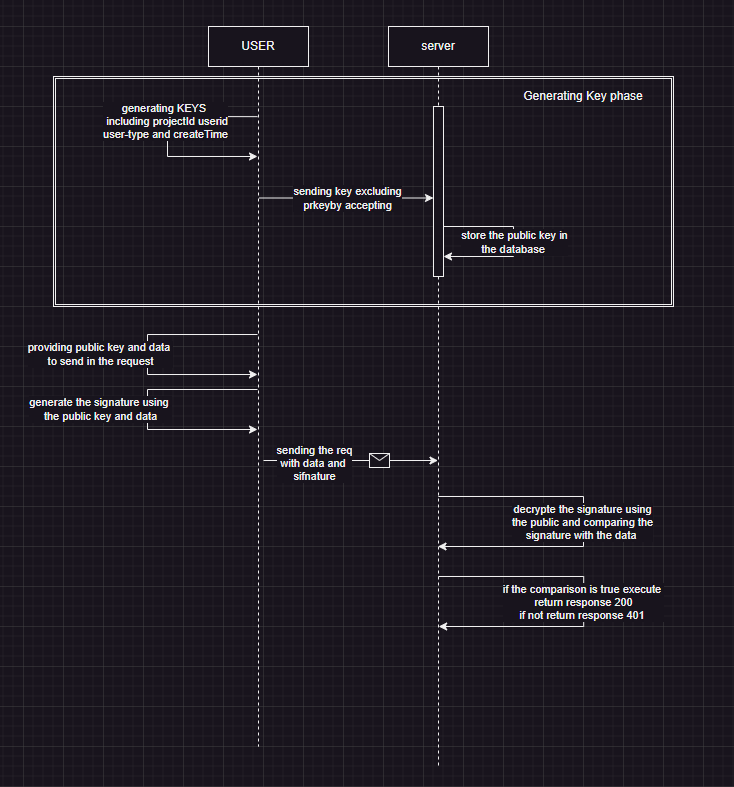
****

Figure explication de tout le process d'authentification via cryptographie asynchrone

L'authentification des requêtes dans le service "**Wallet as a Service**" repose sur l'utilisation de la cryptographie asymétrique, une méthode robuste de sécurisation des transactions. Voici les étapes clés de ce processus :

**Génération des Paires de Clés :** Tout d'abord, chaque utilisateur autorisé génère une paire de clés, composée d'une clé publique et d'une clé privée. La clé privée est gardée secrète, tandis que la clé publique est accessible publiquement.

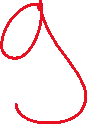
**Association de la Clé Publique :** La clé publique de l'utilisateur est enregistrée dans le système d'authentification du service "Wallet as a Service". Cela permet au service de vérifier l'authenticité des requêtes signées par cette clé publique.

**Création de la Requête Signée :** Lorsqu'un utilisateur souhaite effectuer une opération sur son portefeuille, il signe la requête à l'aide de sa clé privée. Cela génère une signature numérique unique associée à la requête.

**Transmission de la Requête Signée :** La requête, accompagnée de sa signature, est envoyée au service "Wallet as a Service" pour traitement.

**Vérification de la Signature :** À réception de la requête, le service utilise la clé publique de l'utilisateur pour vérifier la validité de la signature. Si la vérification réussit, cela confirme que la requête provient de l'utilisateur authentifié.

**Traitement de la Requête Autorisée :** Si la signature est validée avec succès, le service "Wallet as a Service" procède à l'exécution de l'opération demandée sur le portefeuille de l'utilisateur.



L'utilisation de cette méthode garantit un niveau élevé de sécurité, car seule la personne possédant la clé privée peut générer des signatures valides. De plus, la clé publique, bien que publique, ne révèle aucune information sur la clé privée correspondante, assurant ainsi un processus d'authentification fiable et sécurisé.

### Creating Wallet

#### concept

un "portefeuille" (ou wallet en anglais) fait référence à une application logicielle ou un outil qui permet aux utilisateurs de gérer leurs avoirs en cryptomonnaie et d'interagir avec les réseaux de blockchain. Il sert d'interface numérique permettant aux utilisateurs de stocker de manière sécurisée, d'envoyer et de recevoir des cryptomonnaies telles que le Bitcoin, l'Ethereum et d'autres jetons.

Voici quelques points clés que vous pouvez inclure dans votre CV de stage pour expliquer ce qu'est un portefeuille en développement web3 :

* Définition :

Un portefeuille web3 est une application logicielle qui permet aux utilisateurs de gérer leurs actifs en cryptomonnaie et d'interagir avec les réseaux de blockchain.

* Fonctions :

Stockage Sécurisé :

Il offre un environnement sécurisé pour stocker les clés privées, qui sont essentielles pour accéder et contrôler les avoirs en cryptomonnaie d'un utilisateur.

Gestion des Transactions :

Il permet aux utilisateurs d'envoyer et de recevoir des cryptomonnaies, ainsi que de consulter leur historique de transactions.

Certains portefeuilles permettent aux utilisateurs d'interagir avec des contrats intelligents sur des plateformes de blockchain comme Ethereum.

Gestion des Jetons :

Ils permettent aux utilisateurs de gérer différents types de jetons, y compris les jetons non fongibles (NFT).

#### FrontEnd :

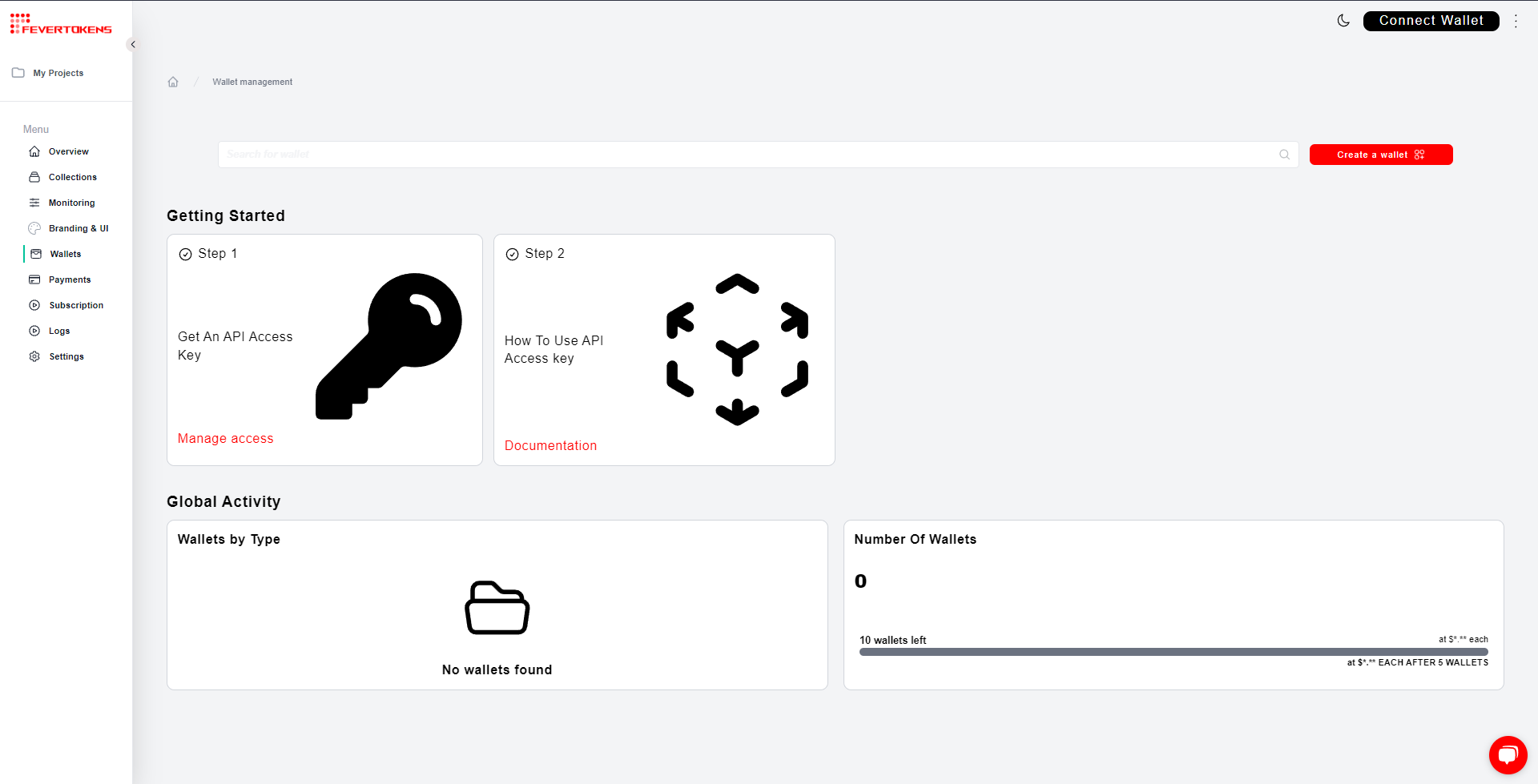




Figure bouton responsable a lancé tout le process création du Wallet



Figure Composent personnaliser Nextjs

1. Composant de Bouton (<GlobalButton>):
   * Ce composant représente un élément interactif que les utilisateurs peuvent cliquer dans une interface utilisateur.
2. Propriétés du Bouton (className, onHoverBgColor, onHoverTextColor):
   * Ces propriétés permettent de personnaliser l'apparence du bouton.
   * className permet d'appliquer des styles TailWind prédéfinis.
   * onHoverBgColor et onHoverTextColor définissent les couleurs du bouton lorsqu'il est survolé avec la souris.
3. Fonction de Gestion du Clic (onClick):

* Cette propriété spécifie une fonction qui sera exécutée lorsque le bouton est cliqué.

1. Mise à Jour de l'État (setAlert((prev) => { ... })):

* Lorsque le bouton est cliqué, cette fonction est appelée pour mettre à jour l'état de l'application.
* Dans ce cas, elle modifie un état appelé alert en utilisant sa valeur précédente.

1. Utilisation de l'État (setIsPrivate(false) et setAlert({ isOpen: true, onCancel: ... })):

* Le code met à jour l'état isPrivate pour le définir sur false. Cela pourrait potentiellement avoir un impact sur d'autres parties de l'application qui dépendent de cet état.
* De plus, il ouvre une alerte ou PopUp en définissant isOpen sur true et en spécifiant une fonction à exécuter lorsqu'elle est annulée (onCancel).

**Avantages :**

* Interactivité :

Le bouton permet aux utilisateurs d'effectuer des actions en cliquant dessus, ce qui améliore l'expérience utilisateur.

* Personnalisation :

Les propriétés comme className, onHoverBgColor, et onHoverTextColor offrent une flexibilité pour adapter le bouton à l'esthétique et au style de l'application.

* Gestion de l'État :

L'utilisation de l'état (alert et potentiellement isPrivate) permet de suivre et de gérer l'état de l'application, ce qui est essentiel pour une expérience utilisateur dynamique.

* Réutilisation :

Le composant de bouton peut être réutilisé à différents endroits de l'application, ce qui favorise la modularité et la maintenabilité du code.

* Interaction avec l'Utilisateur :

En associant des actions à des événements tels que les clics de souris, l'application peut répondre de manière dynamique aux interactions des utilisateurs, ce qui améliore l'engagement.

##### WalletType PopUp :

Le composant <WalletType> représente un élément essentiel de notre application, conçu pour offrir une expérience utilisateur interactive et flexible lors de la création et de la gestion de portefeuilles. Sa conception dynamique et sa capacité à être réutilisé le rendent polyvalent pour l'exécution de diverses fonctions au-delà de la simple création de portefeuilles.

Ce composant s'adapte de manière intelligente à différentes situations grâce à ses nombreuses propriétés configurables. Il est capable de gérer des opérations asynchrones, de valider des limites et de réagir aux interactions de l'utilisateur de manière fluide.

Que ce soit pour la création de portefeuilles, l'édition de paramètres, ou d'autres tâches, <WalletType> offre une interface intuitive et réactive, facilitant ainsi une variété d'opérations liées aux portefeuilles de manière sécurisée et efficace. Son code bien organisé et modulaire le rend également facilement extensible pour prendre en charge de nouvelles fonctionnalités à l'avenir.

La réutilisabilité de ce composant est un atout majeur de notre architecture, permettant une gestion plus efficace des différentes opérations liées aux portefeuilles au sein de notre application. En l'intégrant dans d'autres parties de notre système, nous sommes en mesure de garantir une expérience utilisateur cohérente et fluide tout au long de l'application.

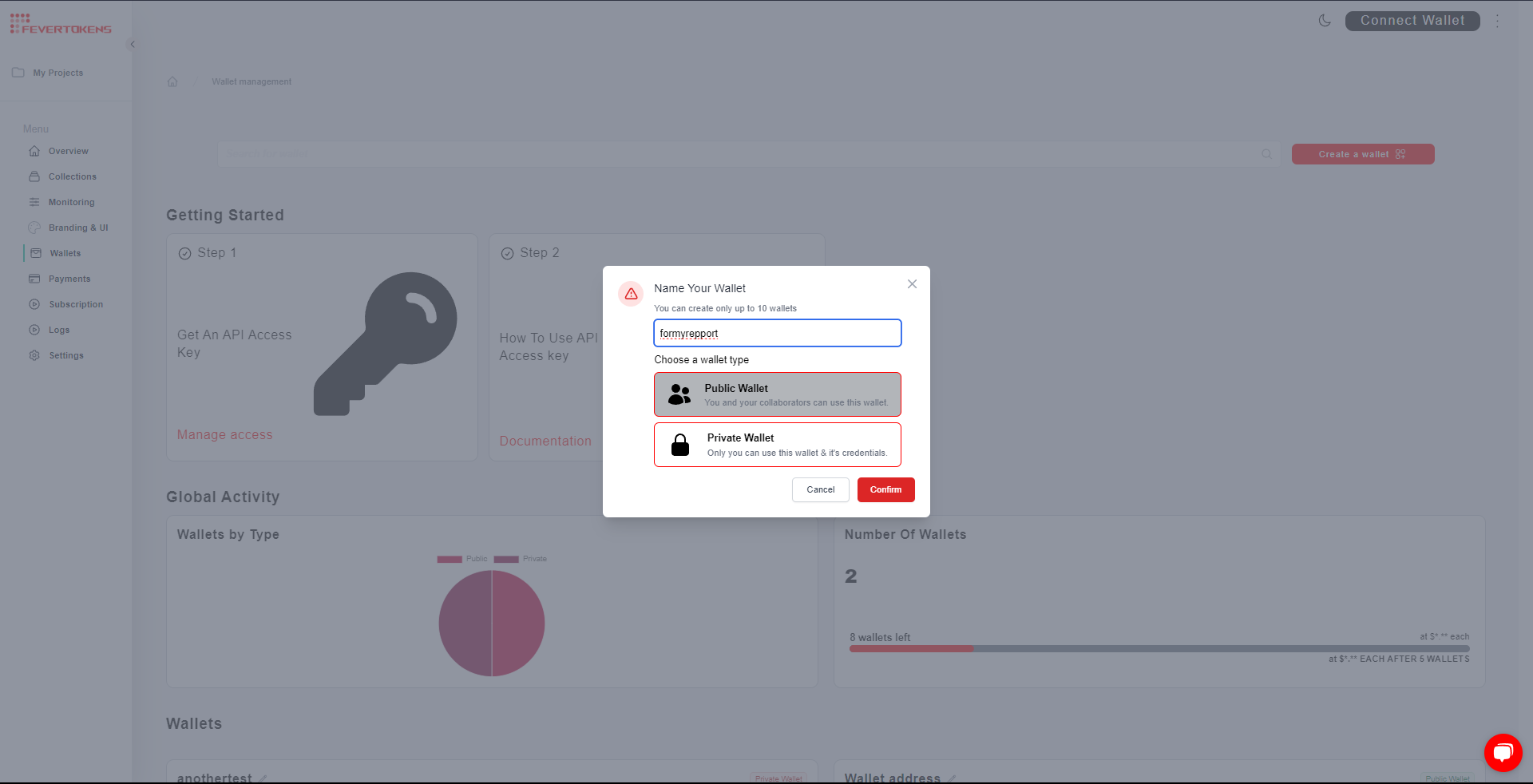


Figure Popup Dynamique

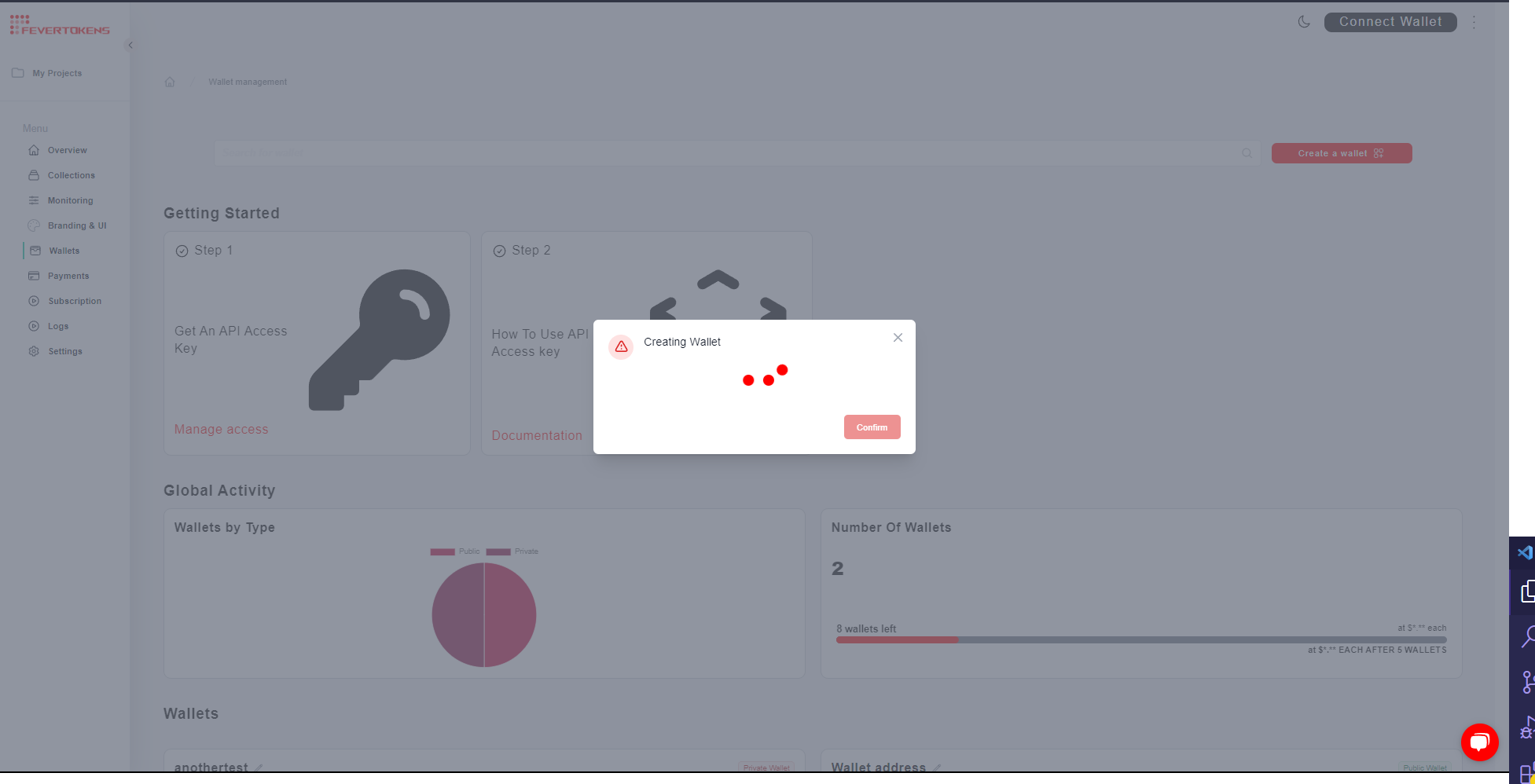


Figure démonstration de dynamiter pour une expérience d'utilisateur optimal

Le composant <WalletType> joue un rôle central dans notre application, offrant une expérience utilisateur interactive et flexible lors de la création et de la gestion de portefeuilles. Sa polyvalence et sa réutilisabilité en font un élément dynamique de notre architecture.

La propriété isOpen contrôle l'état d'ouverture ou de fermeture de la boîte de dialogue associée, permettant une interaction fluide avec le composant. Lorsque isOpen est vrai, la boîte de dialogue est affichée, facilitant ainsi les actions de l'utilisateur.

Lorsque l'utilisateur soumet la boîte de dialogue, la fonction onSubmit est déclenchée. Celle-ci contient une logique conditionnelle qui effectue diverses actions en fonction du nombre actuel de portefeuilles. Cela inclut la création d'un nouveau portefeuille avec des paramètres spécifiques, le défilement vers la liste des portefeuilles, et la mise à jour de divers états pour garantir une expérience utilisateur transparente.De même, la fonction onCancel est appelée lorsqu'un utilisateur choisit d'annuler l'opération. Elle assure la fermeture de la boîte de dialogue et réinitialise les états associés, garantissant une réaction appropriée à l'interaction de l'utilisateur.

En outre, le composant offre une personnalisation avancée grâce à des propriétés telles que alertTitle, qui adapte le titre de l'alerte en fonction de divers scénarios, et alertDescription, qui fournit des informations contextuelles supplémentaires.

L'utilisation de fonctions asynchrones avec async et await permet une gestion propre et lisible des opérations asynchrones, garantissant une expérience utilisateur fluide.

Enfin, la réutilisabilité du composant <WalletType> est un élément crucial de notre architecture, lui permettant d'être déployé dans divers contextes de l'application, ce qui en fait un outil polyvalent pour les interactions liées aux portefeuilles. Ce composant joue ainsi un rôle essentiel dans la création et la gestion de portefeuilles, offrant une expérience utilisateur riche et sécurisée.



Figure code du Pop-up personnaliser

La fonction createNewWallet représente une étape cruciale de notre application, facilitant la création de nouveaux portefeuilles de manière sécurisée et réactive, grâce à l'environnement Next.js. Tout d'abord, elle vérifie la validité des données fournies, s'assurant que le champ du nom du portefeuille n'est pas laissé vide et que le niveau de confidentialité est correctement spécifié. Si ces conditions sont satisfaites, un indicateur de chargement est activé pour informer les utilisateurs que le traitement est en cours, profitant ainsi de la vitesse et de la réactivité caractéristiques de Next.js.

La fonction orchestre ensuite une requête API de type POST vers notre serveur, où les paramètres nécessaires pour la création du portefeuille (tel que le niveau de confidentialité et le nom) sont encapsulés. Une fois cette requête exécutée avec succès, la boîte de dialogue d'alerte est fermée, ce qui assure une expérience utilisateur fluide et réactive, conforme à l'architecture réactive de Next.js.

En cas de détection d'erreurs, telles que des champs vides, des messages d'erreur sont générés pour guider les utilisateurs. De plus, pour garantir que les données de portefeuille sont toujours à jour, nous utilisons la bibliothèque useSWR de Next.js. Cette bibliothèque, non seulement améliore les performances en évitant les appels réseau inutiles, mais permet également à la fonction mutate d'être appelée dans createNewWallet. Ainsi, après la création réussie du portefeuille, les données sont immédiatement mises à jour sans nécessiter de rechargement de page.

En conclusion, la fonction createNewWallet s'intègre harmonieusement dans notre application Next.js, exploitant ses capacités de réactivité et de gestion d'état. Son code rigoureusement structuré et ses fonctionnalités optimisées en font un élément fondamental de notre architecture système, contribuant ainsi à une expérience utilisateur fluide et sécurisée.

#### API wallets :

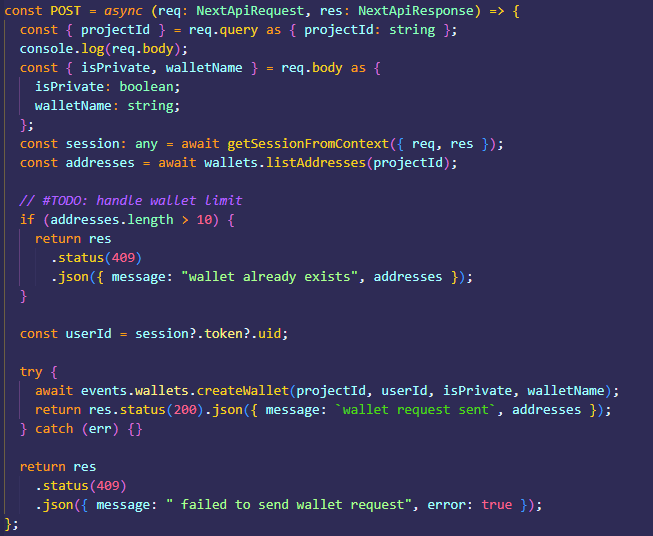
****

Figure API responsable a création des wallet

La fonction POST gère les requêtes de type POST à l'end-point "wallets". Elle récupère des données essentielles du corps de la requête pour la création d'un portefeuille électronique. Elle extrait également d'autres informations de la session utilisateur et de l'URL de la requête.

Cette fonction exécute plusieurs étapes importantes. Tout d'abord, elle vérifie si la limite de portefeuilles a été atteinte en utilisant la fonction wallets.listAddresses. Cette dernière réalise une requête à la base de données pour récupérer tous les portefeuilles associés au même utilisateur.

Ensuite, la fonction appelle un événement. Cela permet d'exécuter une autre fonction, qui prendra en charge le traitement ultérieur de la création du portefeuille électronique.

En résumé, la fonction POST est responsable de la gestion des requêtes POST vers l'end-point "wallets". Elle collecte les données nécessaires, effectue des vérifications importantes (comme la limite de portefeuilles), et déclenche un événement pour poursuivre le processus de création du portefeuille.

#### Lambda createwallet :

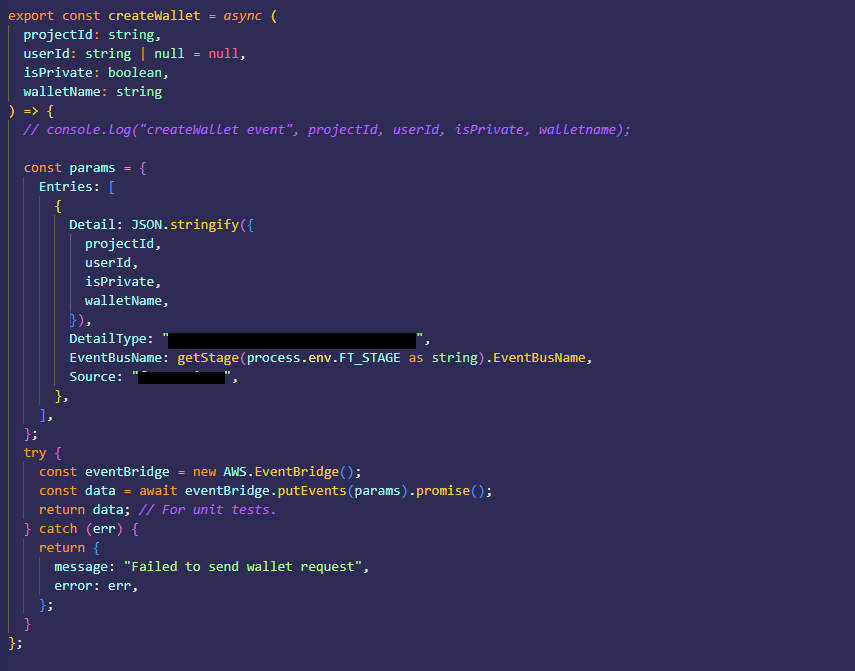


Figure fonction lambda

La signature de la fonction createWallet récupère les informations nécessaires pour créer un portefeuille à partir des paramètres de la requête POST. Ces informations comprennent le projectId, l'userId (qui peut être optionnel), un indicateur de confidentialité isPrivate, et le nom du portefeuille walletName.

L'événement à envoyer à Amazon EventBridge est préparé en empaquetant ces informations sous forme d'une structure de données. Notamment, la valeur du champ EventBusName est déterminée en appelant la fonction getStage qui, à son tour, utilise une variable d'environnement pour sélectionner le bus d'événements approprié.

Ensuite, une instance de l'Amazon EventBridge est créée. Cette instance servira à envoyer l'événement à AWS Lambda, permettant ainsi de déclencher l'exécution de la fonction sans serveur (serverless).

En somme, cette fonction joue un rôle crucial dans le processus de création de portefeuille. Elle rassemble les détails nécessaires, les encapsule dans un événement, puis utilise Amazon EventBridge pour transmettre cet événement à AWS Lambda, déclenchant ainsi l'exécution du code associé à cet événement. Cela permet de gérer la création de portefeuille de manière asynchrone et évite de bloquer le processus principal.

#### Apres creation du wallet :

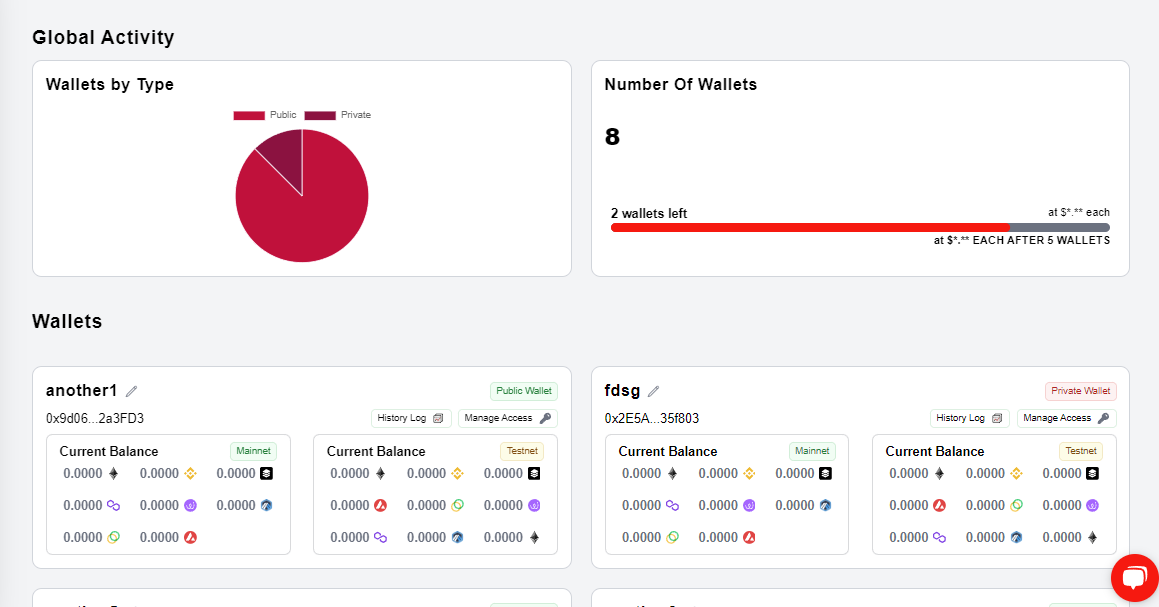


Figure statistiques et monitoring des wallets

Apres la création du wallets l’actualisation des wallets se fait automatiquement aussi que les graphs pour visualiser le status de ses wallets

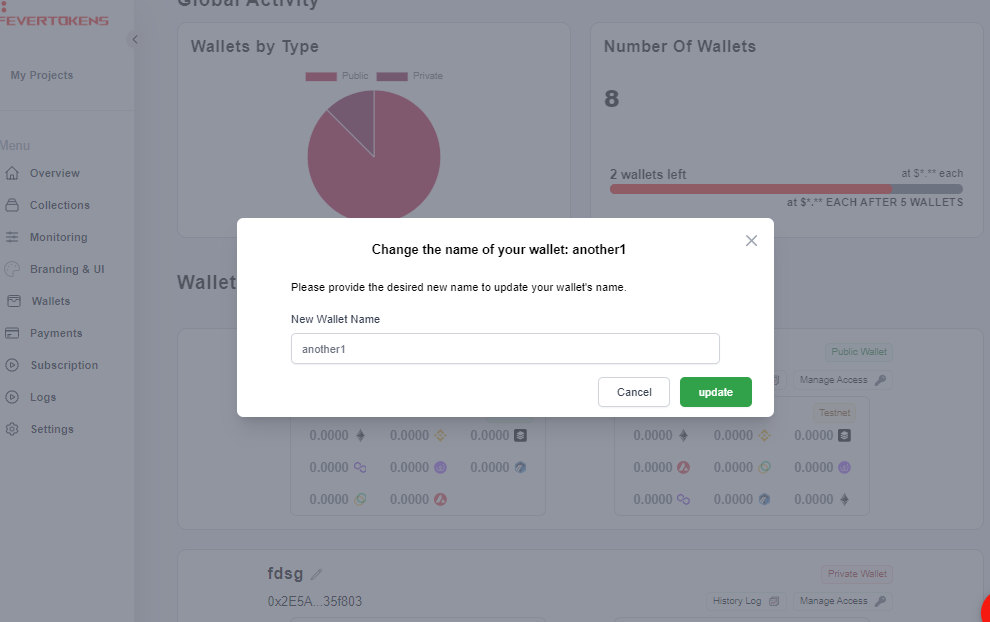


Figure Popup pour changer le nom du wallet selectionner

La possibilité de changer le nom d'un portefeuille instantanément est réalisable grâce à l'utilisation d'une requête de type PUT, de la même manière dont nous avons créé le portefeuille initial. Cette opération permet de mettre à jour le nom du portefeuille sans avoir à recréer un nouveau. Cela offre une grande flexibilité et permet aux utilisateurs de personnaliser et de gérer leurs portefeuilles selon leurs besoins et préférences. Cette fonctionnalité est particulièrement utile dans des situations où l'utilisateur souhaite effectuer des ajustements à la configuration de son portefeuille sans avoir à recourir à un processus de création complet.



Figure API responsable du changement du nom des wallets



Figure fonction qui lance la requête du changement du nom

#### Test du wallet crée :

Pour tester les portefeuilles nouvellement créés, on peut utiliser des faucets de Polygon, qui sont des outils en ligne permettant de distribuer de la fausse cryptomonnaie. Ces faucets sont conçus pour simuler des transactions réelles sur le réseau Polygon, offrant ainsi une façon pratique de vérifier le bon fonctionnement des portefeuilles électroniques. En procédant ainsi, on peut créditer ces portefeuilles avec une petite quantité de cryptomonnaie factice pour effectuer des transactions et observer leur comportement dans un environnement de test. Cela permet de s'assurer que les portefeuilles sont correctement configurés et opérationnels avant de les mettre en production

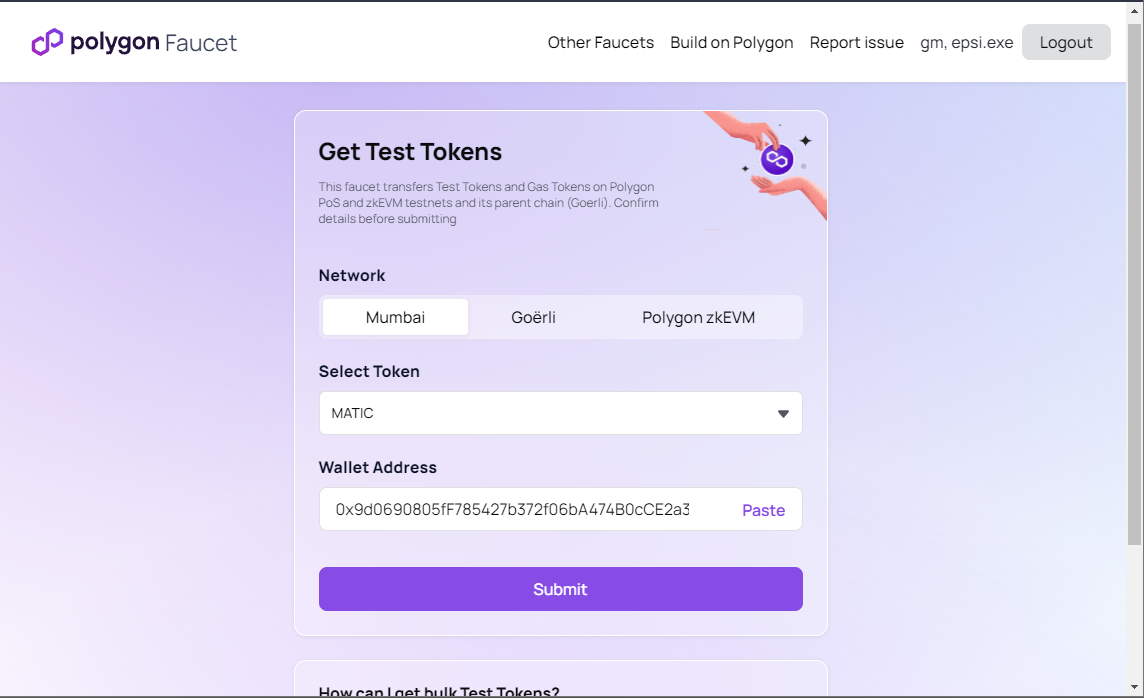


Figure PolygonFaucet plateforme qui fournit des fausse cryptocurrencies

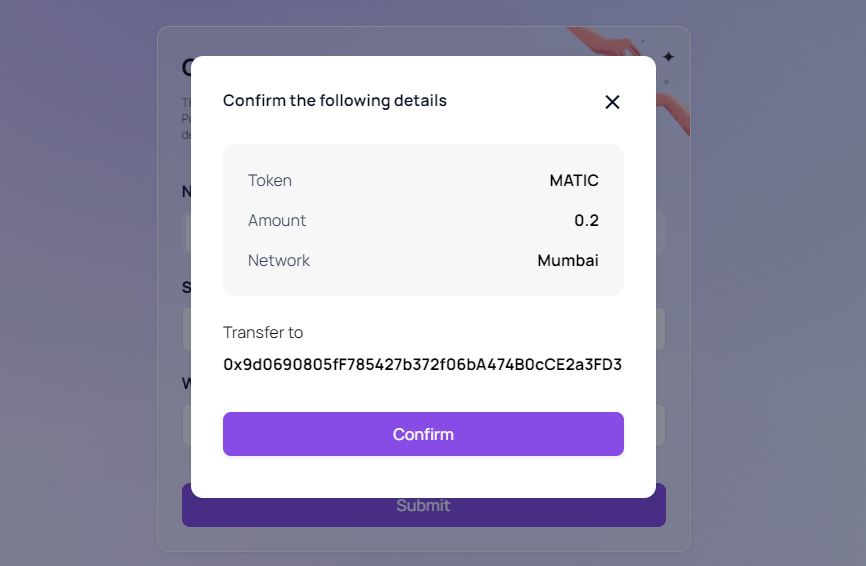


Figure Confirmation de demande 0.2 matic

Pour démontrer le processus, j'ai capturé des images à chaque étape. J'ai utilisé le site web https://faucet.polygon.technology/ comme fournisseur de faucets pour recevoir une allocation de 0.2 faux Polygon (MATIC). En suivant les instructions du site, j'ai fourni l'adresse du portefeuille que j'avais créé. En quelques instants, le faucet a traité ma demande et a transféré 0.2 MATIC sur le portefeuille spécifié. Ces captures d'écran documentent de manière visuelle le succès de cette opération, mettant en lumière l'efficacité du processus de réception de la cryptomonnaie simulée à des fins de test sur le réseau Polygon.

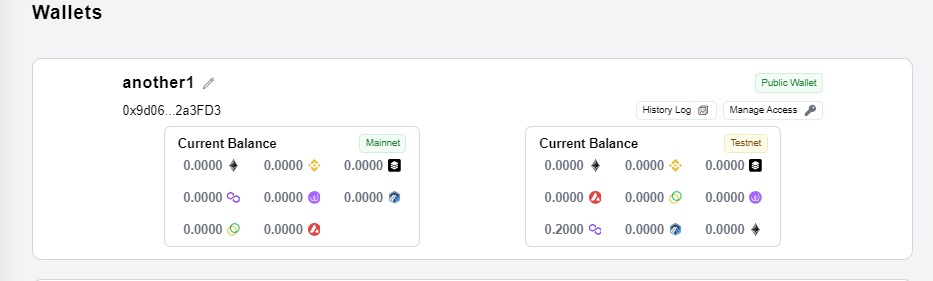


Figure la réception de 0.2 matic demandée

### Web3 explorer

J'ai accompli une tâche particulièrement significative dans mon projet, en développant un explorateur Web3 qui offre une fonctionnalité d'importation de contrats à partir de différents réseaux blockchain. Cette innovation ouvre de nouvelles perspectives pour les développeurs et les utilisateurs qui cherchent à interagir avec des contrats intelligents sur diverses chaînes de blocs. Grâce à cette fonctionnalité, il devient possible d'intégrer des contrats existants dans nos projets sans avoir à les créer manuellement, simplifiant ainsi considérablement le processus de développement d'applications décentralisées (dApps) et d'interaction avec les réseaux blockchain. Dans les paragraphes suivants, je vais expliquer en détail le fonctionnement et les avantages de cet explorateur Web3 ainsi que les étapes que j'ai suivies pour sa mise en place.

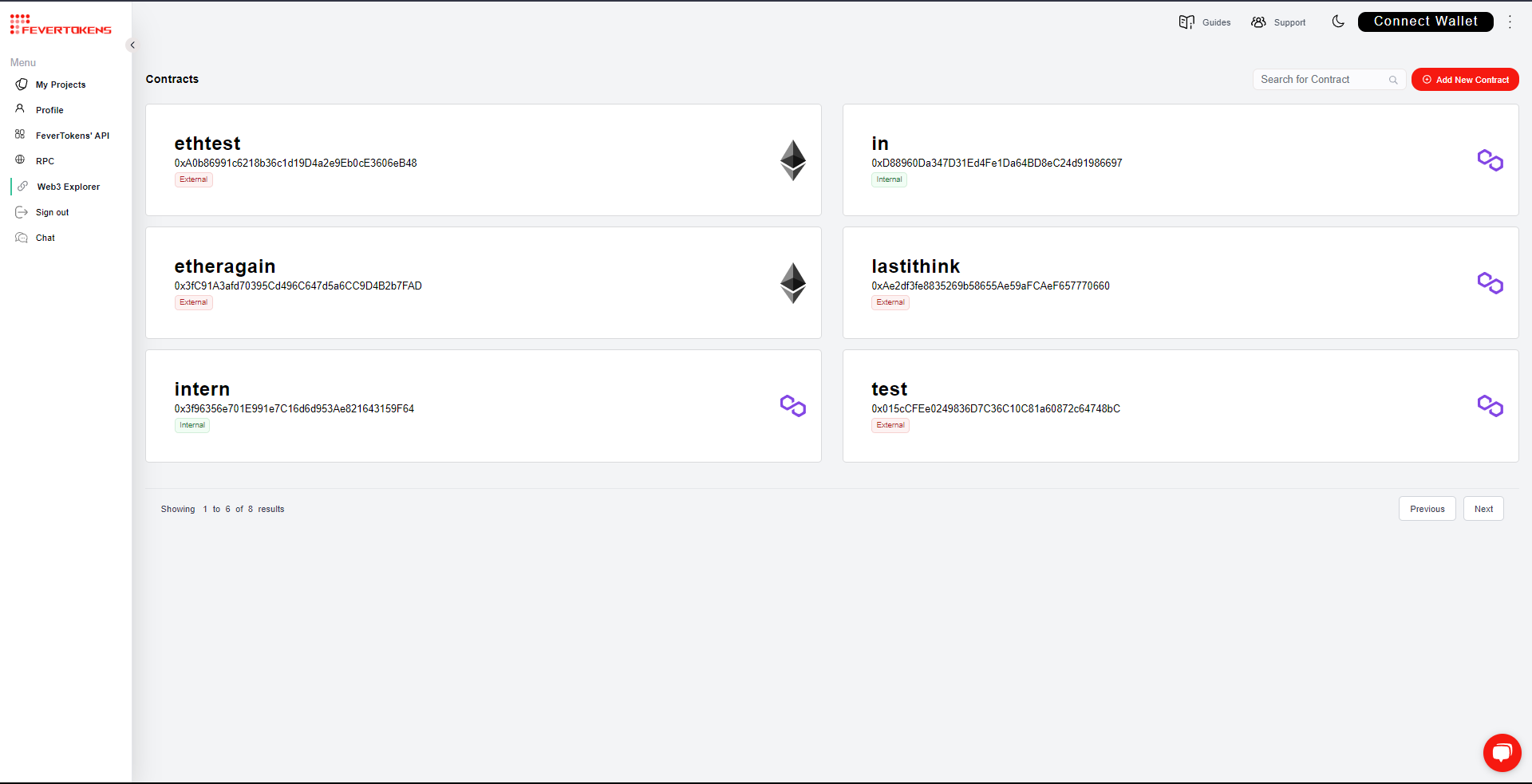


Figure page principale de WEB3 explorer

Sur la première page, vous serez accueilli par une liste de contrats importés. Ces contrats peuvent provenir de deux sources : les contrats externes, importés à l'aide d'API telles qu'Etherscan ou Polygonscan pour le réseau Mumbai, ou bien des contrats internes provenant de FeverTokens.

Une barre de recherche de contrats est également présente pour vous permettre de facilement trouver un contrat spécifique par son nom ou son adresse. Cela vous offre une fonctionnalité de recherche efficace pour naviguer parmi les contrats importés.

De plus, un bouton d'ajout de contrat est disponible, ce qui vous offre la possibilité d'importer de nouveaux contrats dans l'explorateur. Cette fonctionnalité est particulièrement utile pour étendre la liste de contrats à explorer et à exploiter dans votre projet. En un clic, vous pouvez intégrer de nouveaux contrats à votre explorateur Web3, élargissant ainsi la portée de vos fonctionnalités et de vos interactions avec les contrats intelligents.

#### Importation intern ou extern d’une contrat

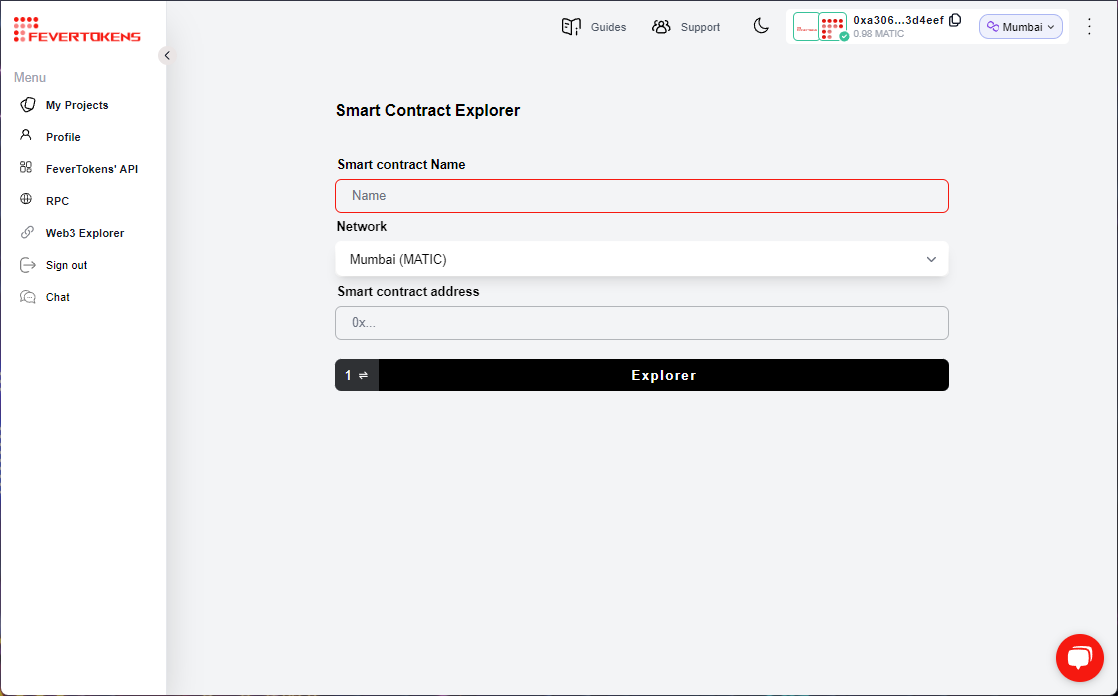


Figure formulaire Dynamique pour importer des contrats intelligentes

Pendant mon stage, j'ai développé un formulaire dynamique qui permet d'importer des contrats intelligents, spécifiquement leurs ABI. Cette fonctionnalité vise à permettre aux utilisateurs, même non-développeurs, d'exécuter ces contrats sans avoir à écrire de code. Le formulaire requiert trois informations essentielles : le nom du contrat intelligent, le réseau de la blockchain et son adresse. Ces détails sont ensuite utilisés par le système pour importer l'ABI associée, simplifiant ainsi l'exécution des propositions ou des fonctions par les utilisateurs.  
validation des address de smarts contracts et leur network :



Figure fonction validation d'adresse

La fonction isSmartContract est une fonction asynchrone qui prend une adresse de contrat intelligent (dans la forme d'une chaîne de caractères) comme argument et retourne une promesse résolue avec un objet contenant des informations sur la validité de cette adresse.

**Voici un aperçu de ce que fait cette fonction :**

Elle vérifie d'abord si provider est défini. Si ce n'est pas le cas, cela signifie que l'utilisateur ne s'est pas connecté à un portefeuille (wallet), et elle renvoie un message d'erreur indiquant qu'il doit se connecter à un portefeuille.

Ensuite, elle vérifie si le réseau actuel de la blockchain correspond au réseau attendu pour le contrat intelligent. Si ce n'est pas le cas, elle renvoie un message demandant à l'utilisateur de changer de réseau

Si tout est en ordre jusqu'à présent, la fonction utilise web3 (une bibliothèque pour interagir avec Ethereum) pour obtenir le code du contrat intelligent en utilisant la fonction getCode. Le résultat est stocké dans isContract.

Elle vérifie ensuite si isContract est une chaîne vide ou nulle. Si c'est le cas, cela signifie que l'adresse du contrat intelligent est invalide dans le réseau actuel, et elle renvoie un message d'erreur en conséquence.

Si l'adresse du contrat est valide, elle indique que c'est le cas et renvoie un message de succès.

Si une erreur se produit pendant ce processus, elle la gère dans la clause catch et renvoie un message d'erreur indiquant que l'adresse du contrat intelligent est invalide.

En résumé, cette fonction vérifie la validité d'une adresse de contrat intelligent dans le réseau actuel. Elle effectue des vérifications pour s'assurer que l'utilisateur est connecté à un portefeuille, que le réseau est correct et qu'il n'y a pas d'erreurs lors de la vérification de l'adresse du contrat intelligent.

##### Migration vers la librairie etherjs :

Modernité et Meilleure Prise en Charge d'Ethereum : Ether.js est une bibliothèque plus récente qui est spécifiquement conçue pour travailler avec Ethereum. Elle prend en charge les dernières fonctionnalités et mises à jour du réseau Ethereum, ce qui peut être essentiel pour rester à jour avec les évolutions de la blockchain.

* Typescript et Meilleure Gestion des Types :

Ether.js est écrit en TypeScript, ce qui signifie qu'il offre une meilleure prise en charge des types par rapport à Web3.js. Cela peut rendre le développement plus sûr et moins sujet aux erreurs de typage.

* Architecture Modulaire :

Ether.js est construit de manière modulaire, ce qui permet aux développeurs de n'inclure que les parties de la bibliothèque dont ils ont besoin, réduisant ainsi la taille de l'application et améliorant les performances.

* Meilleure Documentation et Communauté Active :

Ether.js a été développé en gardant à l'esprit une documentation claire et concise, ainsi qu'une communauté active qui peut fournir un soutien en cas de besoin. Cela facilite l'apprentissage et le développement avec la bibliothèque.

* Interopérabilité avec d'Autres Blockchains :

Ether.js a été conçu pour prendre en charge plusieurs blockchains Ethereum, ce qui le rend potentiellement plus adaptable si vous travaillez avec d'autres réseaux Ethereum ou des blockchains compatibles Ethereum.

* Réactivité et Gestion d'Événements Améliorées :

Ether.js offre une gestion des événements améliorée par rapport à Web3.js, ce qui peut être important pour les applications décentralisées (dApps) qui nécessitent une interaction en temps réel avec la blockchain.

* Meilleures Performances :

Ether.js a été conçu avec un souci de performance, ce qui peut se traduire par des temps de réponse plus rapides et une utilisation plus efficace des ressources.

Cependant, il est important de noter que le choix entre Web3.js et Ether.js dépendra toujours des besoins spécifiques de votre projet, de la familiarité de l'équipe avec chaque bibliothèque, et de la nature de l'application que vous développez. Il peut également être utile de consulter la documentation officielle de chaque bibliothèque pour comparer leurs fonctionnalités et leurs avantages respectifs.

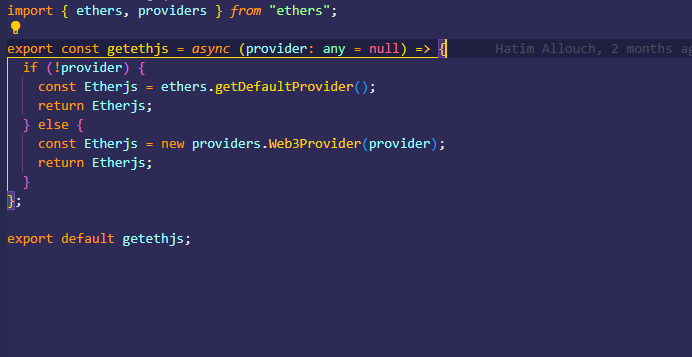


Figure fonction d'initialisation du Provider

La fonction getWeb3 est une fonction JavaScript asynchrone qui permet d'obtenir une instance de la bibliothèque Web3. Cette bibliothèque est utilisée pour interagir avec les blockchains décentralisées, en particulier avec Ethereum.

**Commençons par expliquer les termes clés :**

* Provider : En blockchain, un "provider" est un composant logiciel qui permet à une application de se connecter à un réseau blockchain. Il fournit les moyens de communication entre l'application et le nœud blockchain. Un provider peut être fourni par un service tiers ou peut être un nœud Ethereum local.
* Signer : Un "signer" (ou "signataire" en français) est une entité qui possède une clé privée permettant de signer des transactions sur la blockchain. Il s'agit essentiellement d'une façon de vérifier l'authenticité d'une action effectuée sur la blockchain.

Maintenant, passons à la fonction getWeb3 :

**Explication ligne par ligne :**

export const getWeb3 = async (provider: any = null): Cette ligne définit une fonction nommée getWeb3 qui est asynchrone (elle utilise async). Elle prend un argument provider en entrée, qui est initialisé à null par défaut.

if (!provider) : Cette condition vérifie si provider n'est pas fourni en tant qu'argument. Si tel est le cas, cela signifie que l'application ne dispose pas d'un provider et doit en créer un.

const Web3 = new (await import("web3")).default(); : Cette ligne importe la bibliothèque Web3 (qui doit être installée dans le projet) et crée une nouvelle instance de celle-ci en l'assignant à la variable Web3.

return Web3; : La fonction retourne ensuite cette instance de Web3.

Si provider est fourni en tant qu'argument, cela signifie que l'application dispose déjà d'un provider. Dans ce cas, la fonction crée une nouvelle instance de Web3 en utilisant ce provider et la retourne de la même manière qu'expliqué ci-dessus.

En fin de compte, la fonction retourne une instance de Web3, prête à être utilisée dans d'autres parties de l'application.

En résumé, cette fonction est un utilitaire qui permet d'obtenir une instance de Web3. Si un provider est déjà disponible, il l'utilise, sinon il en crée un nouveau. Cela facilite l'initialisation de Web3 dans différentes parties de votre application.

#### Récupération générative des ABI :

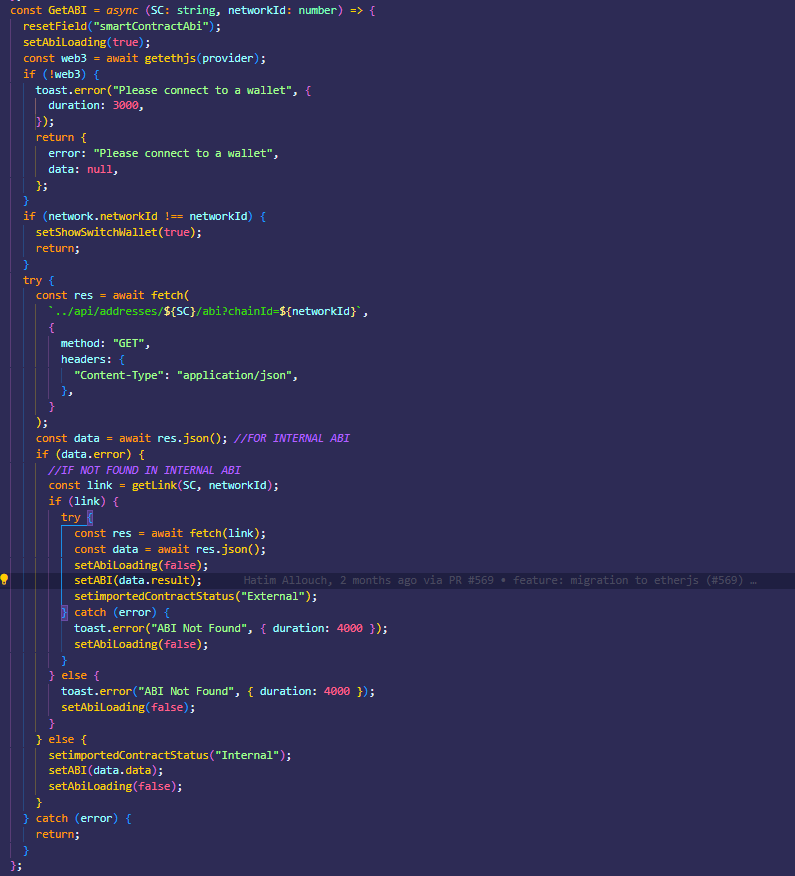


Figure fonction pour récupérer l'ABI

La fonction GetABI a pour but de récupérer l'ABI (Interface Binaire d'Application) d'un contrat intelligent (smart contract). Voici comment elle fonctionne :

Tout d'abord, elle effectue une réinitialisation d'un champ appelé smartContractAbi, probablement pour préparer l'application à recevoir une nouvelle ABI.

Ensuite, elle indique que le chargement de l'ABI est en cours en mettant la variable abiLoading à true, ce qui permet d'informer l'interface utilisateur qu'une opération est en cours.

Ensuite, elle utilise une fonction appelée getethjs pour obtenir une instance de web3, qui est une bibliothèque utilisée pour interagir avec la blockchain Ethereum.

Elle effectue ensuite des vérifications pour s'assurer que l'utilisateur est connecté à un portefeuille (wallet) et que le réseau sélectionné est correct. Si ces conditions ne sont pas remplies, un message d'erreur est affiché et la fonction renvoie un objet avec un message d'erreur.

Ensuite, la fonction envoie une requête HTTP GET pour récupérer l'ABI du contrat intelligent. Le résultat de cette requête est traité et transformé en format JSON.

La suite du code gère différentes situations en fonction de la réponse de la requête :

Si la réponse indique une erreur (indiquée par data.error), cela signifie que l'ABI n'a pas été trouvée localement. Dans ce cas, la fonction tente de la récupérer à partir d'une source externe. Si cela réussit, l'ABI est récupérée et mise à jour dans l'application.

ABI End-poit :



Figure API pour récupérer les contrats interne

La fonction GET est une partie essentielle de cette API. Elle est responsable de gérer une requête GET entrante.

Tout d'abord, elle extrait les paramètres walletAddress et chainId de la requête. Ces paramètres sont essentiels pour la suite de l'opération.

Ensuite, elle effectue des vérifications pour s'assurer que walletAddress et chainId sont bien définis. Si l'un d'entre eux est manquant, elle renvoie une réponse d'erreur avec un code HTTP 400 et un message approprié.En cas de succès, la fonction appelle une autre fonction nommée GetPackageNameWithContractType avec les paramètres walletAddress et chainId. Cette fonction semble renvoyer un objet contenant contractType et error.

Si error est présent dans la réponse de GetPackageNameWithContractType, cela signifie qu'il y a eu un problème et une erreur est générée.

Sinon, l'ABI (Interface Binaire d'Application) associée à contractType est récupérée en appelant la fonction getABI.

La réponse finale est renvoyée sous forme de JSON, contenant soit l'ABI ou un message d'erreur si quelque chose ne se passe pas comme prévu.

En cas d'erreur pendant le processus, une autre tentative est faite pour obtenir l'ABI depuis une source externe, comme indiqué dans le commentaire "getabi from external api".

Si tout se passe bien, la réponse contenant l'ABI est renvoyée avec un code HTTP 200, sinon une réponse d'erreur avec un code HTTP 500 est renvoyée.

Recuperation des API externe :

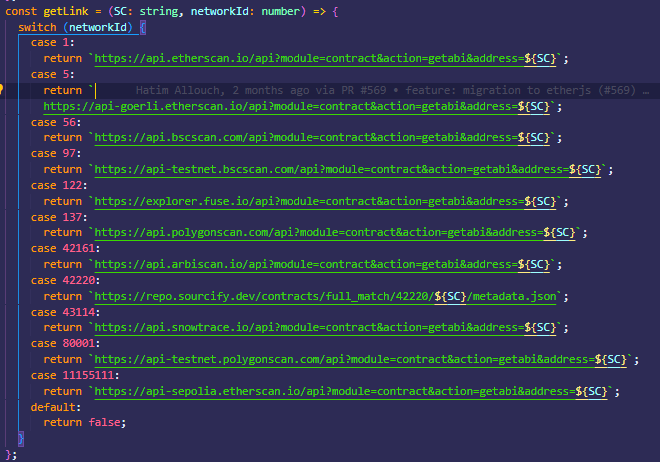


Figure fonction pour trouver l'API convenable au NetworkID

Si la réponse ne contient pas d'erreur, cela signifie que l'ABI a été trouvée localement, et elle est mise à jour dans l'application.

En cas d'erreur pendant le processus, la fonction retourne simplement sans effectuer d'action supplémentaire. En résumé, cette fonction gère le processus de récupération de l'ABI d'un contrat intelligent, en prenant en compte différentes sources possibles et en mettant à jour l'application en conséquence. Elle fournit également des retours d'informations à l'interface utilisateur pour indiquer l'état de l'opération.

**La validation des ABI :**

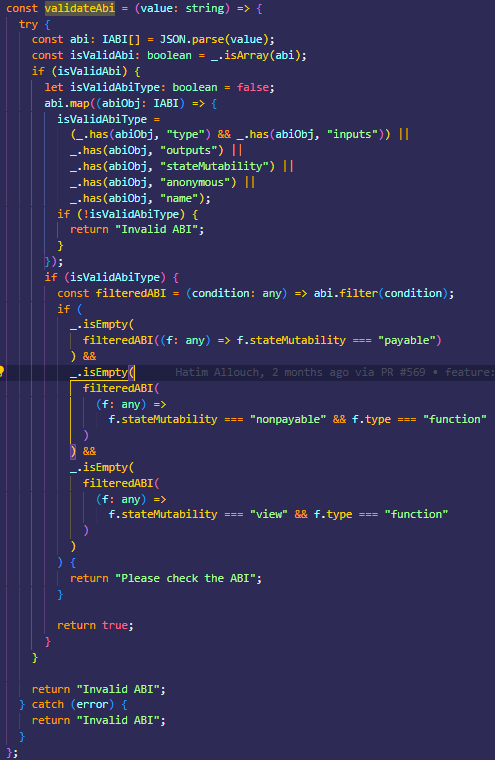


Figure validation d'Abi récupéré

La fonction validateAbi a pour rôle de vérifier la validité de l'ABI (Interface Binaire d'Application) fournie sous forme de chaîne de caractères. Tout d'abord, elle essaie de parser cette chaîne en un tableau d'objets représentant l'ABI. Ensuite, elle vérifie si le résultat est effectivement un tableau. En parallèle, elle effectue une série de vérifications pour s'assurer que chaque objet de l'ABI contient les propriétés essentielles attendues. Si une de ces propriétés est manquante, cela indique une ABI invalide.

En cas de succès jusqu'à ce point, la fonction procède à des vérifications plus spécifiques. Elle filtre les fonctions de l'ABI en fonction de leur état de mutabilité et de leur type, et s'assure qu'au moins une fonction de chaque type est présente.

Si toutes les vérifications réussissent, la fonction renvoie true, signifiant que l'ABI est valide. En cas d'erreur à n'importe quel moment du processus, la fonction retourne "Invalid ABI". En résumé, cette fonction fournit un moyen de valider une ABI et de signaler si elle correspond aux critères requis.

# CONCLUSION GENERALE

En conclusion, mon stage chez FeverTokens a été une immersion captivante dans un environnement technologique dynamique. Travailler sur le développement complet d'une plateforme en utilisant Next.js et Node.js, tout en intégrant un système hybride cloud et la puissance de la blockchain, a été une expérience exceptionnelle. Chaque choix technologique a été minutieusement sélectionné pour résoudre des problèmes spécifiques et surmonter des défis techniques, contribuant ainsi à la création d'une plateforme novatrice et performante.

Les outils et technologies clés, notamment la sécurisation des API, ont collaboré de manière synergique, démontrant leur efficacité dans la gestion de projet, la construction de l'interface utilisateur, la garantie de la sécurité et l'interaction avec la blockchain à travers Web3. Cette intégration harmonieuse a permis d'atteindre les objectifs fixés, démontrant la pertinence des choix technologiques opérés.

Sur le plan méthodologique, la structure de développement mise en place a favorisé une communication fluide et une coordination efficace malgré la nature virtuelle de la collaboration. Les réunions régulières et l'utilisation de plates-formes telles que GitHub Projects ont facilité la gestion des tâches et le suivi des progrès, tout en maintenant un flux de travail organisé et vérifié.

En termes d'apports et de compétences acquises, ce stage m'a permis d'approfondir mes connaissances en développement web, en architecture cloud et en blockchain. J'ai également renforcé mes compétences en communication, en gestion de projet et en résolution de problèmes. Ces acquis seront sans aucun doute précieux pour mes futurs projets professionnels.

Je recommande vivement FeverTokens comme un environnement stimulant pour les passionnés de technologie et d'innovation. Leur approche intégrée de la technologie blockchain et du développement web offre une expérience d'apprentissage inestimable.

En perspective, je suis convaincu que les connaissances et les compétences acquises au cours de ce stage seront un atout majeur dans ma carrière future. Je suis impatient de mettre en pratique ces acquis et de continuer à contribuer à des projets innovants et axés sur la technologie.

# BIBLIOGRAPHIE

# Bibliographie

Onyekwere, T. (2022, janvier 8). *Top 3 No-code/low-code Platforms for Smart Contract Development*. Récupéré sur Medium: https://betterprogramming.pub/3-no-code-platforms-to-build-and-deploy-smart-contracts-11a51196796a