



Le génie pour l'industrie

RAPPORT TECHNIQUE  
PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
DANS LE CADRE DU COURS MGL850 - APPLICATIONS ET SYSTÈMES  
DÉCENTRALISÉS

**REVERSE**  
**UN ÉCHANGE IMMOBILIER PAIR À PAIR DÉCENTRALISÉ**

Khalil Anis ZABAT – ZABK86080008

Youssef ROULAMELLAH – ROUY29049301

Kamil TALEB – TALK29089800

Azeddine AIT-OUARAB – AITA22069804

Hanz SAMI – SAMH26079805

DÉPARTEMENT DE GÉNIE LOGICIEL ET DES TI

**Professeur**

**M. Kaiwen ZHANG**

MONTREAL, 17 AOÛT 2023  
ÉTÉ 2023



Khalil Anis Zabat, Youssef Roulamellah, Kamil Taleb, Azeddine Ait-Ouarab, Hanz Sami, 2023



Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

## **REVERSE : RAPPORT TECHNIQUE PLATEFORME D'ÉCHANGE IMMOBILIÈRE**

Khalil Anis ZABAT – ZABK86080008  
Youssef ROULAMELLAH – ROUY29049301  
Kamil TALEB – TALK29089800  
Azeddine AIT-OUARAB – AITA22069804  
Hanz SAMI – SAMH26079805

### **RÉSUMÉ**

Notre projet s'articule principalement autour de l'utilisation de la technologie blockchain pour révolutionner le marché immobilier, offrant transparence, rapidité et accessibilité dans les transactions immobilières. En mettant l'accent sur la tokenisation des propriétés immobilières à travers des jetons non fongibles (NFTs), nous proposons une nouvelle approche des échanges immobiliers.

Ce rapport détaille notre démarche, mettant en lumière la manière dont la blockchain peut résoudre les problématiques actuelles du secteur immobilier. Nous expliquons les décisions de conception qui ont guidé notre travail et les stratégies adoptées pour la mise en œuvre de cet écosystème.

Considérant nos cas d'utilisation et notre expertise, notre choix s'est porté sur la blockchain Ethereum, utilisant le langage Solidity. La logique inhérente à nos contrats intelligents, combinée à l'intégrité immuable de la blockchain, garantit aux propriétaires une totale souveraineté sur leurs biens tout en simplifiant les transactions entre les différentes parties.

Pour ce qui est de nos livrables, ces derniers rassemblent une interface utilisateur basée sur Next.js 13 qui interagit avec nos contrats grâce au framework Thirdweb. Nous utilisons MetaMask comme gestionnaire de portefeuille et l'accès à la blockchain est assuré par Infura.

La plateforme REverse permet plusieurs fonctionnalités. Absolument, il est possible de tokeniser une propriété pour la vendre et d'assurer la mise en place d'un système d'enchère. De plus, un système d'approbation de vente est garanti par des parties tierces autorisées telles qu'un notaire et un inspecteur.

Nous abordons également des concepts innovants tels que la propriété fractionnelle et l'emprunt avec une propriété comme garantie. Finalement, ce rapport présente notre conception du système, les résultats de notre mise en œuvre, ainsi qu'une analyse des coûts et des performances de notre solution.

## TABLE DES MATIÈRES

	Page
INTRODUCTION .....	1
CHAPITRE 1 PRÉSENTATION DU RAPPORT .....	2
1.1 Proposition et contexte du projet .....	2
1.2 Spécifications détaillées du projet .....	2
CHAPITRE 2 ANALYSE ET CONCEPTION .....	5
2.1 Analyse des cas d'utilisation .....	5
2.2 Caractéristiques de l'architecture et décisions de conception .....	11
2.2.1 Source de données .....	11
2.2.2 Application Web & Frameworks .....	12
2.2.2.1 Cadriciel Next.js .....	12
2.2.2.2 Librairie Thirdweb .....	12
2.2.2.3 Service de nœud décentralisé.....	13
2.2.2.4 Portefeuille de cryptomonnaie.....	13
2.2.2.5 Blockchain – Testnet Ethereum.....	14
2.2.2.6 Hardhat .....	14
2.2.3 Smart Contracts – Thirdweb et OpenZeppelin .....	14
2.2.3.1 MarketplaceV3 – Sommaire .....	15
2.2.3.2 RealEstateBasic.sol .....	16
2.2.3.3 EnglishAuction .....	17
2.2.3.4 EscrowV4 .....	18
2.3 Décision de conception et d'implémentation .....	18
2.3.1 Choix du standard de “tokens” (Ethereum Request for Comments).....	18
2.3.2 Choix de bibliothèques .....	18
2.4 Diagrammes de classes et de séquences.....	19
2.4.1 Diagramme de séquence .....	19
2.4.2 Diagramme de classe .....	21
CHAPITRE 3 IMPLÉMENTATION .....	22
3.1 Configuration requise.....	22
3.2 Patron de conception.....	23
3.2.1 Patron proxy .....	23
3.2.2 Patron composite .....	24
3.3 Liste des vues de l'application.....	25
CHAPITRE 4 VALIDATION ET VÉRIFICATION .....	33
4.1 Plan de tests, évaluation de performance et résultat .....	33
4.2 Recommandations d'améliorations futures .....	34

CONCLUSION .....	36
LISTE DE RÉFÉRENCES .....	37
ANNEXE I.....	38

## **INTRODUCTION**

L'immobilier est un secteur centralisé et ancré avec de solides règles, normes et procédés. Il aura fallu plusieurs années d'évolution technologique pour commencer à imaginer une solution qui répondrait aux besoins tout en assurant les mêmes actions, mais de manière décentralisée. Cette prouesse est désormais possible grâce à la blockchain. Effectivement, nous sommes sur le point de connaître une transformation radicale, et ce, à l'aide de l'application REverse. Cette plateforme sert donc d'interface innovante qui utilise la technologie des contrats intelligents dans le but d'améliorer le marché immobilier et de le rendre beaucoup plus accessible en assurant un certain niveau de décentralisation. Il sera donc question essentiellement du processus de la tokenisation des propriétés via des jetons non fongibles (NFTs).

Ce rapport technique a pour objectif de détailler le travail réalisé par notre équipe dans le cadre du cours de MGL850. Nous aborderons le contexte et la proposition du projet afin de mettre au clair nos objectifs. Nous poursuivrons avec les spécifications détaillées du produit. Une section sera dédiée à l'analyse et la conception du projet. Bien sûr, nous expliquerons nos choix au niveau de l'implémentation technique des outils. Puis, nous détaillerons le processus de validation et de vérification de notre solution.

# CHAPITRE 1

## PRÉSENTATION DU RAPPORT

### 1.1 Proposition et contexte du projet

Depuis la pandémie Covid-19, l'économie mondiale bat de l'aile. La nouvelle génération se tourne davantage sur les moyens de gagner du revenu passif par la possession de bien immobilier afin de lutter contre l'inflation. Le marché actuel est malheureusement confronté à divers défis en raison de l'opacité, de la lenteur des transactions et de l'accès limité. Ces problèmes découlent en grande partie du caractère centralisé et souvent obsolète des systèmes de gestion immobilière actuels. Afin de remédier à ces enjeux, on propose un projet permettant la décentralisation du système immobilier et une accessibilité à travers le monde.

C'est dans ce contexte que REverse vise à révolutionner le secteur immobilier en exploitant la technologie de la blockchain ainsi que les jetons non fongibles (NFT). Chaque bien immobilier étant intrinsèquement unique, cela rejoint le concept d'un NFT sur la blockchain. Chaque étape du processus de transaction d'une propriété sera automatisée par la plateforme règlementée par des lois. L'achat ou la vente d'une maison n'aura jamais été si simple. Les utilisateurs sauveront énormément de temps. Ils auront la possibilité de gérer leur bien avec transparence et sécurité sur le marché immobilier, peu importe où il se trouve dans le monde seulement par l'usage d'un ordinateur et un accès à Internet. La plateforme REverse est une nouvelle approche technologique prometteuse pour le futur de l'immobilier.

### 1.2 Spécifications détaillées du projet

L'objectif de ce projet est de repenser la façon dont les transactions immobilières sont exécutées, en apportant davantage de transparence, de sécurité et d'accessibilité au processus. L'une des caractéristiques centrales de REverse consiste à tokeniser les propriétés

immobilières. Chaque bien immobilier est symbolisé par un jeton non fongible (NFT), qui est unique et intrinsèquement lié à la propriété qu'il représente. Ils devront respecter les normes de l'industrie (ERC-721) pour assurer l'interopérabilité avec d'autres applications blockchain. Ces NFT peuvent être achetés, vendus et échangés au sein de la plateforme REverse, ce qui facilite les transactions immobilières sans nécessiter d'intermédiaires. Toutes ces actions utilisateurs sont affichées dans un historique facilement accessible.

En plus de cela, REverse introduit la notion de propriété fractionnée. Grâce à la tokenisation, une propriété peut être divisée en plusieurs parts, chacune étant représentée par un NFT. Cette approche permet aux utilisateurs d'acheter des fractions de propriété, démocratisant ainsi l'investissement dans l'immobilier. Les utilisateurs ont la possibilité d'investir dans des biens immobiliers de grande valeur sans avoir à acquérir la propriété dans son intégralité.

La plateforme va être conforme aux règlements et lois d'où provient l'utilisateur. Pour ce faire, l'usager passe à travers le processus de vérification d'identité (KYC). Cela consiste à envoyer des documents légaux pour valider l'identité et ainsi appliquer les réglementations au compte de celui-ci. Le fonctionnement de l'application change selon la localisation de son déploiement et la localisation de l'utilisateur.

En exploitant des contrats intelligents, REverse peut automatiser les processus de transaction, garantissant que toutes les conditions de chaque transaction sont satisfaites avant sa finalisation. Le processus de vérification est tout aussi automatisé pour la protection contre les Rugpull, autrement dit un type de fraude couramment utilisé par les créateurs d'NFT. Cela renforce la sécurité de la plateforme et accroît la confiance des utilisateurs.

Puisque la plateforme est disponible sur les appareils mobiles, un mécanisme d'authentification sécurisé sera intégré, tel que l'identification à deux facteurs ou l'identification faciale. La sécurité de REverse s'étend de la blockchain jusqu'à l'interface

utilisateur de la plateforme. La disponibilité est aussi prise en compte sachant que très grand nombre de requêtes transactionnelles devra être gérées, on ne veut surtout pas que la plateforme soit assujettie à des attaques DDOS.

Dans le cadre de ce projet, une application frontale sera développée en utilisant Next.js. Cette interface offrira diverses fonctionnalités aux utilisateurs, leur permettant de se connecter à l'application via MetaMask. Une fois connectés, ils auront la possibilité de créer leur propre NFT en utilisant un contrat intelligent développé en Solidity. Les utilisateurs pourront accéder à une page affichant tous les NFT qu'ils possèdent sur la plateforme REverse. De plus, un jeton REverse sera émis sur un réseau de test pour faciliter son achat/vente.

## **CHAPITRE 2**

### **ANALYSE ET CONCEPTION**

#### **2.1 Analyse des cas d'utilisation**

Dans le cadre de notre projet, la grande majorité des cas d'utilisation initialement envisagée ont été couverts. La Figure 1 présente l'ensemble des cas d'utilisation inclus dans notre proposition de projet, ainsi que les nouvelles additions effectuées au cours de la réalisation de la preuve de concept. Sur les quinze cas d'utilisation répertoriés, onze ont été couverts avec succès. Tous les cas partiellement ou non couverts ont été consignés dans le Tableau 1. Celui-ci comprend le nom du cas d'utilisation, le pourcentage de réalisation, une description détaillée du cas d'utilisation, ainsi que des commentaires relatifs à l'état actuel de l'implémentation et, le cas échéant, les défis rencontrés.

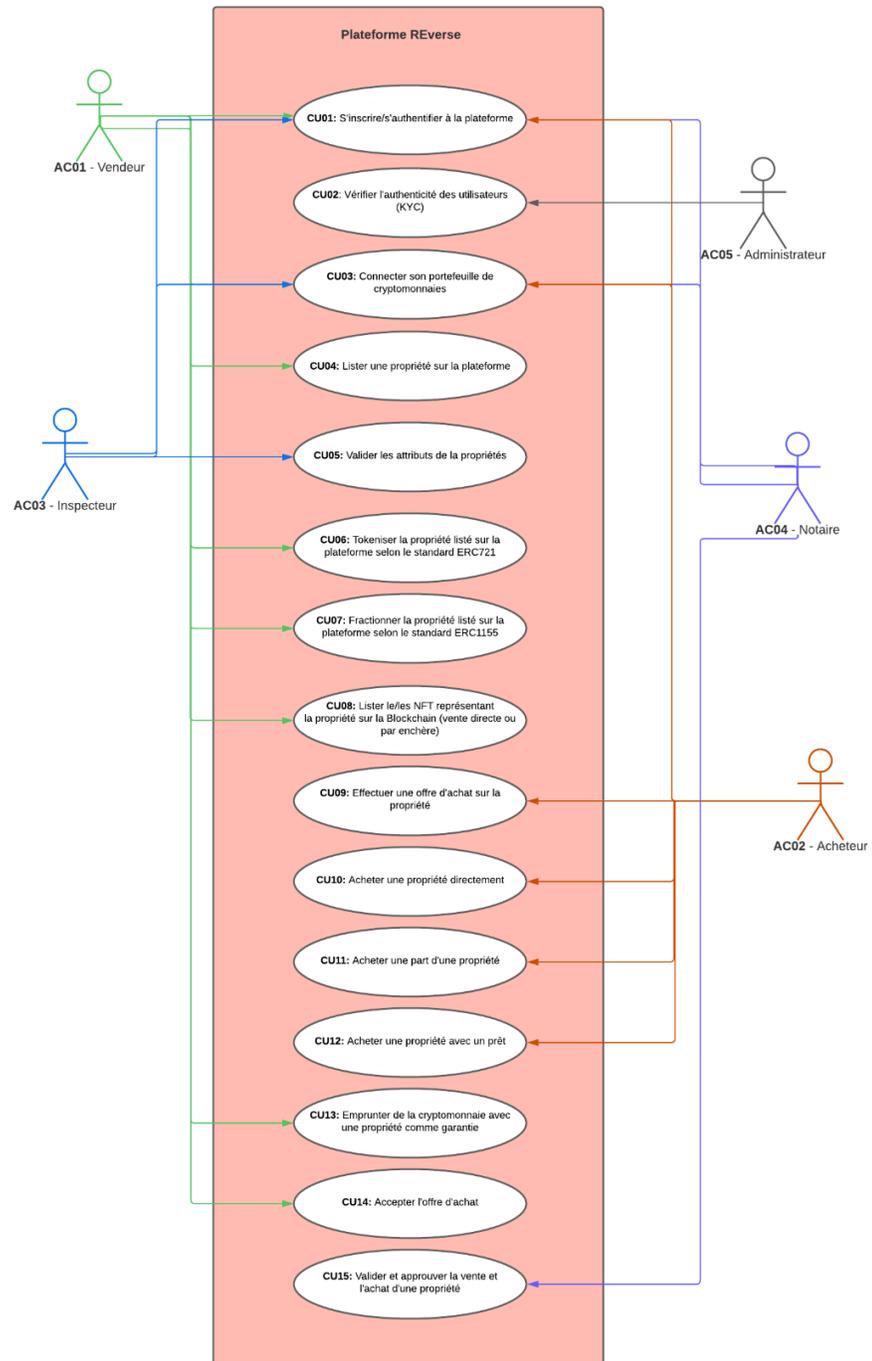


Figure 1 – Sommaire des cas d'utilisations

#ID	Nom	Complétion	Description	Commentaires
CU01	S'inscrire et s'authentifier à la plateforme	100%	Il est possible pour les utilisateurs qui souhaitent acheter ou vendre des propriétés de s'inscrire à la plateforme ou de s'authentifier directement à l'aide de leur compte Google, Github ou Facebook de manière sécuritaire à l'aide de OAuth 2.0.	Aucun
CU02	Vérifier l'identité des utilisateurs (Processus KYC)	0%	L'identification des utilisateurs est récupérée et validée au moyen d'un processus KYC qui extrait les informations d'une pièce d'identité gouvernementale et les valide automatiquement en utilisant une photographie portrait de l'utilisateur.	Voir ANNEXE I
CU03	Connecter son portefeuille de cryptomonnaies	100%	La plateforme offre aux utilisateurs la possibilité de connecter leur portefeuille de cryptomonnaies à la plateforme en détectant automatiquement les extensions présentes dans leur navigateur web. Les portefeuilles supportés comprennent MetaMask, CoinbaseWallet, WalletConnect ou toute autre extension de navigateur injectée tels que TrustWallet et d'autres.	Aucun
CU04	Lister une propriété sur la plateforme	100%	La plateforme offre la possibilité à l'utilisateur de lister sa propriété sur la plateforme. Pour ce faire, il doit fournir tous les détails pertinents sur la propriété, y compris son emplacement, sa taille, son prix, ses photos et toutes autres	Aucun

			informations pertinentes. À noter qu'il s'agit d'une étape préliminaire à la <i>tokenisation</i> de la propriété et sa mise en vente.	
<b>CU05</b>	Valider les attributs de la propriété	75%	Les attributs de la propriété nécessiteront une validation de la part d'un utilisateur exerçant le rôle d'inspecteur, qui s'assurera également de l'existence effective de la propriété. Une fois cette vérification effectuée, les visiteurs de la propriété constateront qu'elle a été soumise avec succès à l'inspection.	Actuellement, l'inspection est réalisée de manière automatisée. Cependant, bien que le concept de rôle ait été intégralement mis en œuvre, le processus d'inspection demeure partiellement implémenté.
<b>CU06</b>	Transformer la propriété listée sur la plateforme en NFT selon le standard ERC721	100%	L'utilisateur devra procéder à la <i>tokenisation</i> de sa propriété en créant un NFT (ERC721) unique qui la représente.	Aucun
<b>CU07</b>	Fractionner la propriété listée sur la plateforme selon le standard ERC1155	0%	L'utilisateur aura l'occasion de fractionner sa propriété, permettant ainsi aux acheteurs potentiels d'acquérir des parts de celle-ci.	Voir ANNEXE I
<b>CU08</b>	Mettre en vente le NFT	100%	Une fois le processus de <i>tokenisation</i> de la propriété terminée, les utilisateurs	Aucun

	représentant la propriété sur la Blockchain		souhaitant mettre leur NFT en vente auront la possibilité de le faire par le biais d'une enchère, laquelle sera définie par une date de début et une date de fin, ainsi que par la fixation d'un prix minimal d'offre et d'un prix d'achat immédiat. Par ailleurs, ils auront également l'option d'opter pour une vente directe de leur propriété à un prix préalablement déterminé.	
<b>CU09</b>	Effectuer une offre d'achat sur la propriété	100%	L'utilisateur désirant acheter une propriété à un prix différent de celui indiqué pour l'achat direct, il aura la possibilité de soumettre une offre d'achat pour ladite propriété.	Aucun
<b>CU10</b>	Acheter une propriété directement	100%	L'utilisateur désirant acheter une propriété au prix affiché pourra le faire à travers la plateforme.	Aucun
<b>CU11</b>	Acheter une part d'une propriété	0%	L'utilisateur pourra acquérir des parts d'une ou plusieurs propriétés correspondant à un pourcentage de la valeur du bien.	Étant donné que l'intégration du standard ERC1155 a été mise en attente, la mise en œuvre de ce cas d'utilisation a également été reportée.
<b>CU12</b>	Acheter une propriété avec un prêt	0%	L'utilisateur aura la possibilité d'acheter une propriété en obtenant un prêt par le biais de pool de prêt décentralisés. L'utilisateur pourra emprunter la somme nécessaire auprès de ces pools en utilisant d'autres actifs cryptographiques ou NFT	Lors de l'implémentation de ce cas, nous avons constaté qu'il exigeait la création d'une plateforme de

			comme garantie. Le prêt sera remboursé au fil du temps, soit par l'utilisateur ou le revenu généré par la propriété elle-même.	prêt indépendante, ce qui dépassait le cadre du présent projet.
<b>CU13</b>	Emprunter de la cryptomonnaie avec une propriété comme garantie	0%	Les utilisateurs de la plateforme pourront utiliser le NFT représentant sa propriété comme garantie pour un prêt lui permettant	Lors de l'implémentation de ce cas d'utilisation, nous avons constaté qu'il exigeait la création d'une plateforme de prêt indépendante, ce qui dépassait le cadre du présent projet.
<b>CU14</b>	Accepter l'offre d'achat	100%	Le vendeur d'une propriété sera en mesure d'accepter l'offre émise par un acheteur et recevra les fonds dans son portefeuille de cryptomonnaie.	Aucun
<b>CU15</b>	Valider et approuver la vente et l'achat d'une propriété	25%	Suite à l'acceptation de l'offre d'achat par le vendeur, le notaire entreprendra un ensemble de vérifications exhaustives pour s'assurer de la conformité de tous les éléments. Ces vérifications incluront la validation des documents légaux, la vérification de l'identité des parties impliquées, l'examen de l'historique de propriété et la confirmation de l'absence de litiges ou de réclamations en suspens. Après avoir mené à bien ces étapes, le notaire procédera à la validation formelle de la transaction. Cette validation	Actuellement, l'approbation de la vente est réalisée de manière automatisée. Cependant, bien que le concept de rôle ait été intégralement mis en œuvre, le processus de vérification et d'approbation des documents n'a pas

			déclenchera le transfert sécurisé des fonds depuis le portefeuille de cryptomonnaies de l'acheteur vers celui du vendeur, assurant ainsi la conclusion efficace et protégée de la vente de la propriété.	été implémenté, car cette étape implique une collaboration avec un notaire (hors porté).
--	--	--	--	--

## 2.2 Caractéristiques de l'architecture et décisions de conception

### 2.2.1 Source de données

L'entreposage de données se fait principalement à travers trois plateformes soit, une base de données MongoDB centralisée, la Blockchain décentralisée d'Ethereum et IPFS, un protocole de partage de fichiers décentralisé. Dans le contexte de notre implémentation actuelle, il est difficile, voire impossible, d'entreposer l'ensemble des données de la plateforme de manière décentralisée. Par conséquent, nous avons adopté une approche hybride. Dans un premier temps, le propriétaire passera au travers d'un processus de vérification et d'authentification sur la plateforme, effectuées au moyen d'un système fictif de validation des données personnelles. Cette procédure garantira la sécurité du système. Les données reliées aux utilisateurs sont entreposées à la fois dans une base de données NoSQL (MongoDB) ainsi que sur la Blockchain. Lors du processus de mise en vente d'une propriété sur la plateforme, le propriétaire doit enregistrer les détails de la propriété qu'il souhaite transformer en un jeton non fongible. Les informations relatives à la propriété seront initialement sauvegardées à travers le réseau pair-à-pair de partage de fichiers appelé IPFS. Par la suite, le hachage de ce fichier, contenant les caractéristiques de la propriété, sera utilisé par les fonctions du contrat intelligent, entraînant ainsi la création d'un jeton non fongible associé à la propriété ainsi qu'à ses attributs, ce qui garantira son caractère immuable.

## 2.2.2 Application Web & Frameworks

### 2.2.2.1 Cadriceil Next.js 13

Pour la partie frontend de notre application, nous avons opté pour l'utilisation de la version 13 du cadriceil Next.js. Ce choix s'est avéré particulièrement judicieux en raison de ses capacités avancées, notamment en termes de rendu côté serveur (SSR) et de rendu côté client (CSR). Next.js s'appuie sur la bibliothèque React et offre un système de développement basé sur des composants réutilisables, qui englobent des sous-composants, garantissant ainsi une structure modulaire, une facilité de maintenance et une claire séparation des responsabilités.

L'un des éléments distinctifs mis en avant par la nouvelle version de Next.js est l'adoption du nouveau répertoire "app". Grâce à cette mise en œuvre, une partie essentielle de notre application est en mesure de fonctionner côté serveur. Cela signifie que même en cas d'erreurs sur certains composants, l'application reste opérationnelle et fonctionnelle. Cette approche assure une stabilité globale du système, évitant les interruptions majeures en cas de problèmes ponctuels.

### 2.2.2.2 Librairie Thirdweb

Reformuler et améliorer la syntaxe de ceci: Thirdweb est un outil spécialisé dans le développement de DApps. Face à la complexité inhérente à la conception d'une application décentralisée, Thirdweb cherche à simplifier le processus en offrant une panoplie de fonctionnalités dédiées au développement d'applications "web 3.0". Dans le contexte de ce projet, Thirdweb s'est révélé essentiel pour l'intégration des fonctionnalités de nos contrats intelligents dans notre DApp, grâce à des "hooks" adaptés à chaque fonction, étant donné notre utilisation de React et TypeScript. De plus, le SDK de Thirdweb renforce la modularité de notre application. Grâce à la diversité des outils qu'il propose, nous pouvons intégrer nos propres solutions, que ce soit pour le

stockage de données sur IPFS ou pour la mise en place de nouveaux hooks. Il est à noter que notre application n'est pas strictement dépendante de Thirdweb, puisque cette logique peut être dissociée du framework. Cela offre une flexibilité appréciable pour les évolutions futures de l'application.

### **2.2.2.3 Service de nœud décentralisé**

Pour faciliter l'accès à la blockchain et à IPFS, nous avons exploité les services d'Infura, qui met à disposition un ensemble complet d'outils gratuits destinés à la création d'applications décentralisées reliées à la blockchain Ethereum. Cette plateforme permet d'accéder rapidement et de manière fiable aux données et aux fonctionnalités de la blockchain, éliminant ainsi la nécessité de gérer la complexité d'une infrastructure de nœud complet. L'intégration à la blockchain Ethereum peut en effet s'avérer complexe et fastidieuse sans une solution telle qu'Infura.

Les services offerts par Infura sont accessibles via une clé API gratuite. Ils permettent d'effectuer une variété d'actions, notamment l'envoi de requêtes, la récupération d'informations relatives aux transactions et aux contrats intelligents, ainsi que les interactions avec IPFS, parmi d'autres fonctionnalités. Cette gamme d'outils essentiels s'avère particulièrement précieuse pour développer des DApps et d'autres services basés sur la blockchain.

### **2.2.2.4 Portefeuille de cryptomonnaie**

La plateforme prend en charge plusieurs portefeuilles, parmi lesquels figurent MetaMask, Coinbase Wallet, WalletConnect, ainsi que tout autre portefeuille accessible via une extension Chrome. Cette intégration est facilitée par l'utilisation de la bibliothèque Thirdweb, qui offre une fonction dédiée à l'établissement de cette connexion.

### **2.2.2.5 Blockchain – Testnet Ethereum**

L'ensemble de nos contrats ont été déployés sur le Testnet d'Ethereum, et nous avons opté pour le Testnet Sepolia pour des raisons spécifiques. Contrairement à d'autres testnets, Sepolia propose plusieurs Faucets, des sources de jetons d'une valeur nominale de 0 \$, qui ne requièrent pas de solde en Ethereum. Cette particularité offre la commodité d'acquérir aisément des SepoliaETH, ce qui facilite le déploiement et le test de nos contrats intelligents.

Un "faucet" agit comme une source de jetons fictifs, permettant aux développeurs de recevoir des tokens pour effectuer des tests sans avoir à détenir de véritable valeur en Ethereum. Cette approche pratique et économique assure un environnement propice à l'exploration et à la mise au point de nos contrats intelligents avant de les déployer sur le réseau principal.

### **2.2.2.6 Hardhat**

Hardhat est un framework de développement spécialisé pour Ethereum, qui est essentiellement conçu pour faciliter la création, le test et le déploiement de smart contracts. L'un des avantages de ce framework réside dans sa facilité de déploiement sur divers réseaux d'Ethereum. Dans le cadre de notre projet, nous avons exploité cette fonctionnalité pour déployer nos contrats intelligents principalement sur les testnets d'Ethereum.

## **2.2.3 Smart Contracts – Thirdweb et OpenZeppelin**

Les contrats intelligents jouent un rôle essentiel au sein de notre application, créant un mécanisme d'échange sécurisé et transparent pour les biens immobiliers dans un cadre décentralisé. Pour réaliser cette vision, nous nous sommes appuyés sur les bases de développement offert par Thirdweb. De manière tout aussi importante, nous avons choisi de

mettre en œuvre les normes rigoureuses d'OpenZeppelin, un pilier reconnu en matière de sécurité des contrats intelligents.

Cette approche offre une série d'avantages significatifs. En intégrant les standards d'OpenZeppelin dans le cadre de Thirdweb, nous bénéficions de solutions préétablies et sécurisées, réduisant les risques potentiels et accélérant le processus de développement. Cela nous permet de consacrer plus de temps et d'attention à la personnalisation fonctionnelle de notre application, en sachant que la sécurité et la robustesse de nos contrats intelligents sont soutenues par des normes largement éprouvées et reconnues.

### **2.2.3.1 MarketplaceV3 – Sommaire**

Le Marketplace V3 de Thirdweb, à l'origine destiné aux NFT, a été adapté pour faciliter les transactions immobilières dans notre contexte spécifique. Il offre aux utilisateurs la possibilité de mettre en liste des propriétés à un prix fixe ou via des enchères, tout en permettant également aux parties non listées, telles que les notaires et les inspecteurs, de soumettre des offres.

Cette adaptation du Marketplace V3 au secteur immobilier comporte des améliorations majeures en termes de conception et de fonctionnalités. Il conserve ses mécanismes d'échange principaux tout en améliorant l'expérience pour l'ensemble des acteurs du secteur, y compris les notaires et les inspecteurs.

Le contrat Marketplace V3 a été optimisé pour répondre aux exigences particulières de l'échange immobilier décentralisé. Dans le cadre de la logique de vente, nous avons choisi d'implémenter le module d'enchères appelé « EnglishAuction », permettant ainsi d'optimiser les processus d'enchères pour une expérience plus fluide et efficiente, au bénéfice de tous les intervenants, qu'ils soient acheteurs, vendeurs, notaires ou inspecteurs.

### 2.2.3.2 RealEstateBasic.sol

Ce contrat intelligent correspond à l'implémentation de l'interface ITokenERC721 proposée par Thirdweb. Il est utilisé pour la tokenization de propriétés immobilières sous forme de jetons non fongibles et offre un ensemble de fonctionnalités essentielles pour faciliter ce processus. Lors de son initialisation, le contrat reçoit divers paramètres essentiels tels que le nom et le symbole, ainsi que les adresses des récipiendaires des ventes primaires et des redevances. De plus, les pourcentages de redevances et de frais de la plateforme sont configurables, ainsi que d'autres paramètres importants.

Le cœur du contrat réside dans la création des NFTs, également appelée "minting". Différents rôles, tels que "MINTER\_ROLE" pour la création de NFTs et "TRANSFER\_ROLE" pour les transferts, sont attribués pour restreindre les actions des utilisateurs du contrat. En parallèle, des rôles administratifs sont définis pour la gestion opérationnelle. Le contrat s'intègre de manière fluide grâce à la mise en œuvre d'interfaces, notamment celles d'OpenZeppelin en matière de sécurité. Il propose également des fonctions spécifiques pour la gestion des opérateurs et des ventes primaires, simplifiant ainsi les interactions. Pour renforcer la sécurité des signatures et la validation des demandes de création, le contrat utilise des fonctions de récupération d'adresse, garantissant la robustesse du contrat.

Méthodes	Description/responsabilité
<b>initialize</b>	Initialise les états du contrat lors du déploiement.
<b>contractType</b>	Retourne le type de module du contrat.
<b>contractVersion</b>	Retourne la version du contrat.
<b>owner</b>	Retourne l'adresse du propriétaire actuel du contrat.
<b>verify</b>	Vérifie la signature d'une demande de mint et le rôle MINTER_ROLE du signataire.
<b>tokenUR</b>	Retourne l'URI pour un ID de jeton donné.
<b>mintTo</b>	Cette méthode est accessible uniquement aux détenteurs du rôle "MINTER_ROLE" et permet de créer un NFT. Chaque NFT créé est

	associé à un URI (Uniform Resource Identifier) contenant les métadonnées détaillées décrivant l'objet tokenisé
<b>recoverAddress</b>	Récupère l'adresse du signataire à partir de la signature d'une demande de mint.
<b>_encodeRequest</b>	Encode les données d'une demande de mint pour la vérification de la signature.
<b>verifyRequest</b>	Vérifie et traite une demande de mint en vérifiant la signature, l'horodatage et le destinataire.
<b>collectPrice</b>	Collecte et distribue le prix de vente primaire du NFT en fonction des informations de la demande.
<b>supportsInterface</b>	Vérifie les interfaces prises en charge par le contrat.

### 2.2.3.3 EnglishAuction

Ce contrat intelligent, intitulé "EnglishAuctionLogic", met en œuvre la logique et les fonctionnalités associées aux enchères anglaises pour les tokens non fongibles (NFTs) conformes aux standards ERC-721. Il gère la création d'enchères, la soumission d'offres, la clôture des enchères, le paiement des redevances et d'autres fonctionnalités associées aux enchères. Le contrat utilise la structure de données du contrat "EnglishAuctionStorage" afin d'entreposer les informations sur les enchères, les offres gagnantes et l'état des paiements des enchères.

Méthodes	Description/responsabilité
<b>createAuction</b>	Permet au créateur d'enchères de créer une nouvelle enchère avec des paramètres spécifiés.
<b>bidInAuction</b>	Permet aux utilisateurs de soumettre des offres pour une enchère en cours.
<b>approveWinnder</b>	Permet au rôle NOTARY d'approuver le gagnant de l'enchère et de transférer les jetons au gagnant.
<b>cancelAuction</b>	Permet au créateur de l'enchère d'annuler une enchère avant sa clôture.
<b>isNewWinningBid</b>	Vérifie si une nouvelle offre est la meilleure offre gagnante.
<b>totalAuctions</b>	Récupère le nombre total d'enchères.
<b>getAuction</b>	Récupère les détails d'une enchère spécifique.

<b>getAllValidAuctions</b>	Récupère toutes les enchères valides en cours.
<b>getWinningBid</b>	Récupère les informations sur l'offre gagnante d'une enchère.
<b>isAuctionExpired</b>	Vérifie si une enchère est expirée.

#### 2.2.3.4 EscrowV4

Ce contrat intelligent combine les fonctionnalités des contrats "MarketplaceV3" et "EnglishAuctionLogic". Il initialise des autorisations spécifiques pour une adresse associée aux biens immobiliers.

### 2.3 Décision de conception et d'implémentation

#### 2.3.1 Choix du standard de "tokens" (Ethereum Request for Comments)

Dans notre démarche de structuration des smart contracts, nous avons envisagé d'adopter le standard ERC-1155 pour assurer la gestion des rôles ainsi que la fractionnalisation des jetons. Cependant, et comme évoqué précédemment, ce standard a été écarté suite aux défis qu'il présentait vis-à-vis des fournisseurs de portefeuilles. En effet, la non-énumérabilité des jetons ERC-1155 compliquait leur traçabilité, nous obligeant à surveiller en permanence les événements de transfert. Cette complexité s'est accentuée avec la considération d'un "subgraph" du protocole "The Graph", d'autant plus que la majorité de notre équipe manquait d'expérience avec les technologies du web 3.0. Face à ces enjeux, notre choix s'est orienté vers le standard ERC-721, apprécié pour sa simplicité et sa compatibilité.

#### 2.3.2 Choix de bibliothèques

Lors de la phase d'implémentation de notre projet, nous avons débuté avec l'utilisation de la bibliothèque WAGMI et d'ethers.js. Nous avons également été confrontés à des défis avec la bibliothèque WAGMI. Sa nouveauté, combinée à un faible soutien, a entraîné des

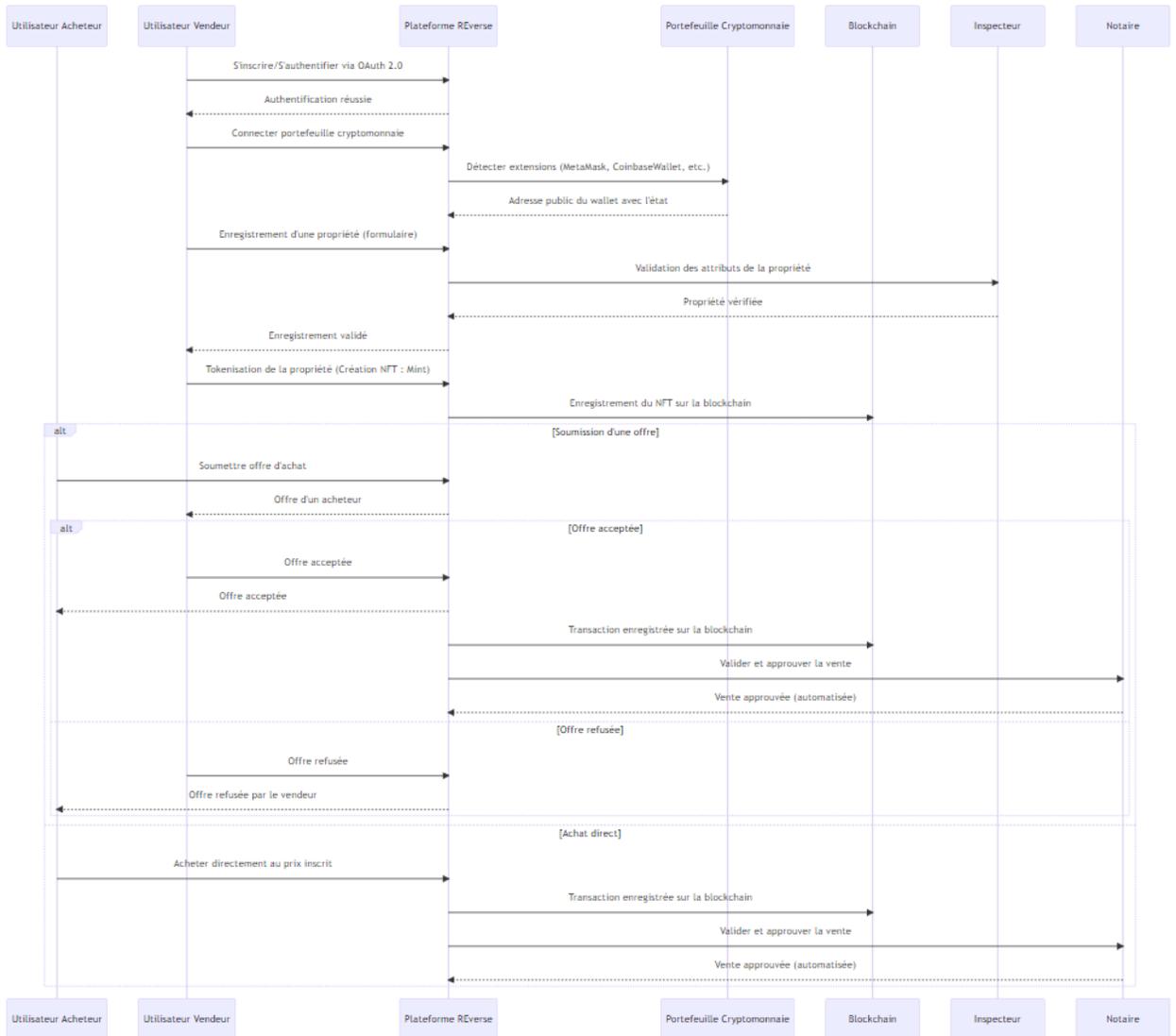
problèmes de stabilité et de fiabilité. Notre manque d'expérience avec cette librairie a rendu les diagnostics difficiles, nous laissant souvent incertains quant à la source de nos problèmes - WAGMI ou nos smart contracts. Cette incertitude a généré d'importants retards dans notre mise en œuvre.

Après avoir surmonté ces obstacles, notre attention s'est tournée vers Thirdweb, un framework spécialisé dans le développement web 3.0. Ce pivot s'est révélé fructueux. Non seulement nous avons pu éliminer certains concepts compliqués, mais nous avons aussi amélioré notre application. En particulier, le token ERC721 proposé par Thirdweb surpassait le nôtre en maturité. De plus, leur contrat "MarketplaceV3" se démarquait par sa sécurité et sa modularité, nous permettant, par exemple, d'y intégrer des rôles spécifiques tels que le notaire et l'inspecteur. Grâce à Thirdweb, notre vision initiale a été concrétisée en un temps record. Leur interface, offrant une visualisation des contrats et des exemples de code, a ajouté une valeur inestimable à notre projet.

## **2.4 Diagrammes de classes et de séquences**

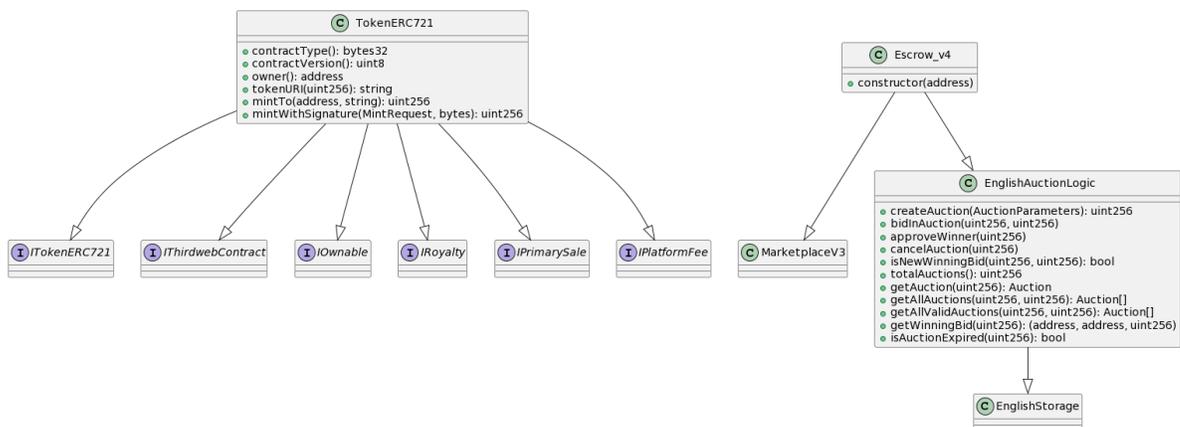
### **2.4.1 Diagramme de séquence**

Le diagramme de séquence ci-dessous représente la grande majorité des cas d'utilisation. On débute par l'inscription (si c'est la première connexion) ou l'authentification de l'utilisateur vendeur sur la plateforme REverse via OAuth 2.0. Une fois authentifié, l'utilisateur vendeur connecte son portefeuille de cryptomonnaie (MetaMask ou CoinbaseWallet) afin de laisser la plateforme détecter l'extension et l'associer au compte. Ensuite, l'utilisateur vendeur procède à l'enregistrement d'une propriété en remplissant le formulaire. Également, l'inspecteur vérifie les attributs de la propriété afin de la valider. L'utilisateur vendeur peut alors tokeniser cette propriété en créant un NFT (Token Non Fongible) qui sera ensuite enregistré sur la blockchain. Ce NFT se met en vente dans un prix fixe et avec une offre minimum. L'utilisateur acheteur a alors deux options : soumettre une offre d'achat (plus que le minimum) ou acheter directement au prix inscrit. L'utilisateur vendeur peut donc accepter ou refuser. Si l'offre est acceptée, une transaction est enregistrée sur la blockchain et ensuite validée par un notaire. Sinon, si l'utilisateur acheteur procède pour un achat direct, la confirmation de l'utilisateur vendeur n'est plus nécessaire. Le processus reprend donc normalement.



## 2.4.2 Diagramme de classe

Le contrat TokenERC721 est une implémentation du standard ERC721, utilisé pour gérer des NFT sur la blockchain Ethereum. Il offre une multitude de fonctions pour créer et interagir, permettant ainsi la gestion des redevances, des ventes primaires et des frais de plateforme. Parallèlement, le contrat Escrow\_v4 sert de facilitateur entre le marché et le système d'enchères, notamment via le contrat EnglishAuctionLogic. Ce dernier, axé sur les enchères anglaises, propose un ensemble complet d'outils pour créer, gérer et conclure des enchères, tout en interagissant avec EnglishStorage pour le stockage des données relatives aux enchères.



## CHAPITRE 3

### IMPLÉMENTATION

#### 3.1 Configuration requise

Pour exécuter avec succès notre projet, il est essentiel de prendre en compte plusieurs prérequis. Tout d'abord, il faut avoir installé la version la plus récente de Node.js, qui est l'environnement d'exécution JavaScript nous permettant de faire fonctionner le projet. De plus, la version la plus récente de MongoDB est nécessaire, car cette base de données NoSQL est au cœur de la gestion de nos données utilisateurs. Il est aussi nécessaire de posséder un compte développeur sur NextAuth.js, une solution d'authentification spécialement conçue pour les applications Next.js. Par ailleurs, un compte sur Thirdweb avec une clé API active est essentiel, étant donné que nous utilisons cette plateforme pour le développement web 3.0. De même, un compte sur Infura avec une clé API active est nécessaire, car Infura offre des outils cruciaux pour interagir avec la blockchain Ethereum. Enfin, pour déployer les contrats ainsi que pouvoir acheter ou vendre des propriétés, un compte sur MetaMask ou CoinbaseWallet est indispensable, et il devrait contenir des SepoliaETH, idéalement au moins 0.1.

Nom de la variable d'environnement	Description
<b>DATABASE_URL</b>	Chaîne de connexion de la base de données MongoDB, contient le nom d'utilisateur, le mot de passe et l'adresse du cluster.
<b>NEXTAUTH_URL</b>	URL de base pour l'authentification avec NextAuth.
<b>NEXTAUTH_SECRET</b>	Clé secrète utilisée par NextAuth pour la signature et la vérification des tokens.
<b>GITHUB_ID</b>	Identifiant client pour l'authentification OAuth avec GitHub.
<b>GITHUB_SECRET</b>	Clé secrète client pour l'authentification OAuth avec GitHub.

<b>GOOGLE_CLIENT_ID</b>	Identifiant client pour l'authentification OAuth avec Google.
<b>GOOGLE_CLIENT_SECRET</b>	Clé secrète client pour l'authentification OAuth avec Google.
<b>NEXT_PUBLIC_CLOUDINARY_CLOUD_NAME</b>	Nom du compte Cloudinary pour la gestion des médias.
<b>INFURA_API_KEY</b>	Clé API pour accéder à la plateforme Infura utilisée pour interagir avec la blockchain Ethereum.
<b>SEPOLIA_PRIVATE_KEY</b>	Clé privée associée à un portefeuille (Metamask dans ce cas).
<b>NEXT_PUBLIC_THIRDWEB_API_KEY</b>	Clé API (ClientId) obtenu sur la plateforme Thirdweb.
<b>NEXT_PUBLIC_THIRDWEB_API_SECRET</b>	Clé secrète permettant d'accéder aux fonctionnalités avancées de Thirdweb.

Le fichier "README.md", situé à la racine de notre projet, contient toutes les instructions détaillées pour faire fonctionner l'application localement ou via un conteneur Docker (optionnel). Toutefois, en guise de synthèse, voici les étapes principales à suivre :

1. Créer un fichier .env comprenant les variables d'environnements (valeurs associées).
2. Utiliser la commande " yarn install " pour installer les dépendances nécessaires.
3. Utiliser la commande " yarn run dev " pour lancer l'application.

## 3.2 Patron de conception

### 3.2.1 Patron proxy

Dans le contexte de contrats intelligents, le patron proxy fait référence à la conception d'architecture où un contrat externe (appelé proxy) agit comme une façade pour un contrat logique interne. Cette manière de procéder permet de séparer la logique et l'implémentation du contrat principal du contrat de proxy qui gère les appels et les interactions avec le contrat principal. Le contrat proxy agit donc comme un intermédiaire pour les appels aux fonctions

du contrat principal ce qui nous permet de mettre à jour le contrat principal sans perturber l'utilisation des fonctionnalités du contrat. Dans notre projet, le contrat intelligent "MarketplaceV3" est considéré comme étant le contrat principal et il va effectuer des appels de fonctions au contrat "EnglishAuctionLogic". Ce dernier agit comme un contrat de gestionnaire pour la logique des enchères anglaises. Il définit des fonctions pour créer des enchères, enchérir, gérer les paiements, etc.

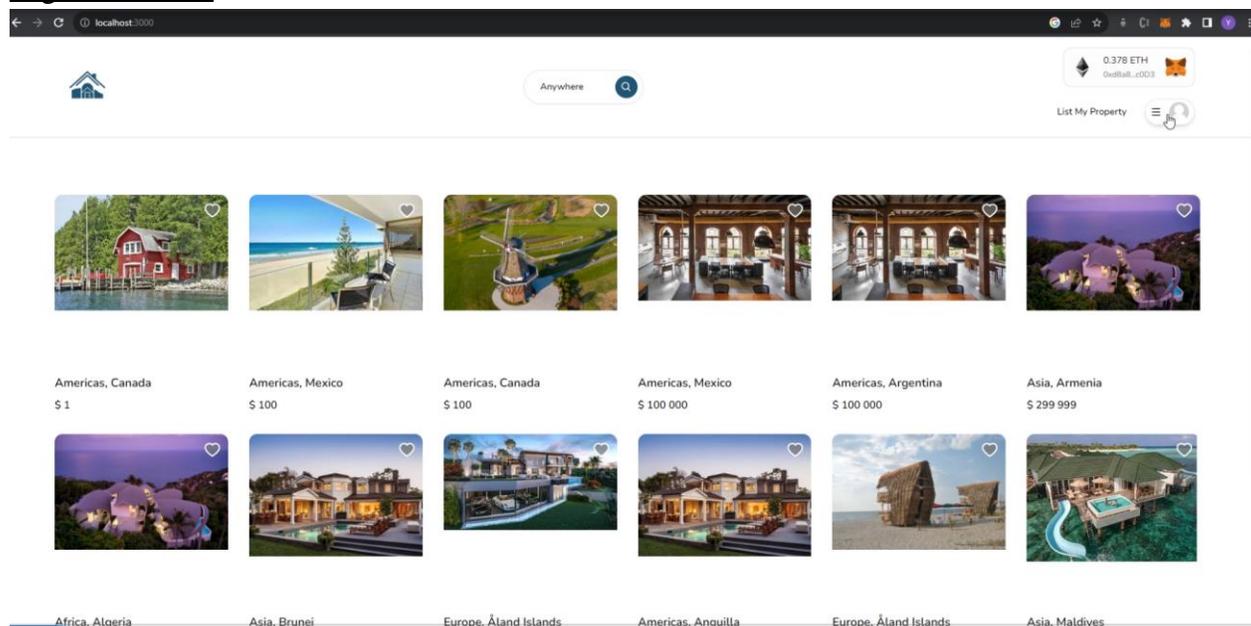
### **3.2.2 Patron composite**

Les différentes composantes de l'application web sont organisées dans une hiérarchie arborescente. Ainsi, chaque composant peut être réutilisé à plusieurs endroits dans l'application, ce qui facilite la gestion de l'interface utilisateur et la maintenance du code. Cela permet également de créer une structure claire et organisée pour notre application tout en tirant parti de la puissance de React pour la manipulation des composants et de l'état.

### 3.3 Liste des vues de l'application

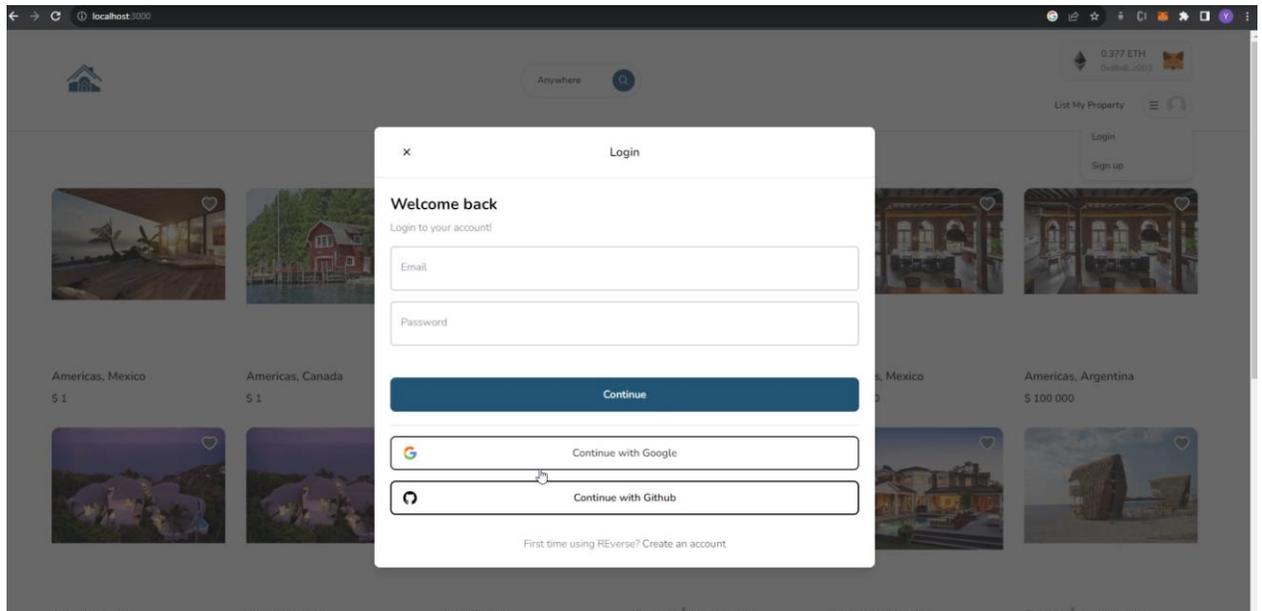
Cette section regroupe toutes les pages principales de l'application Reverse :

#### Page d'accueil :



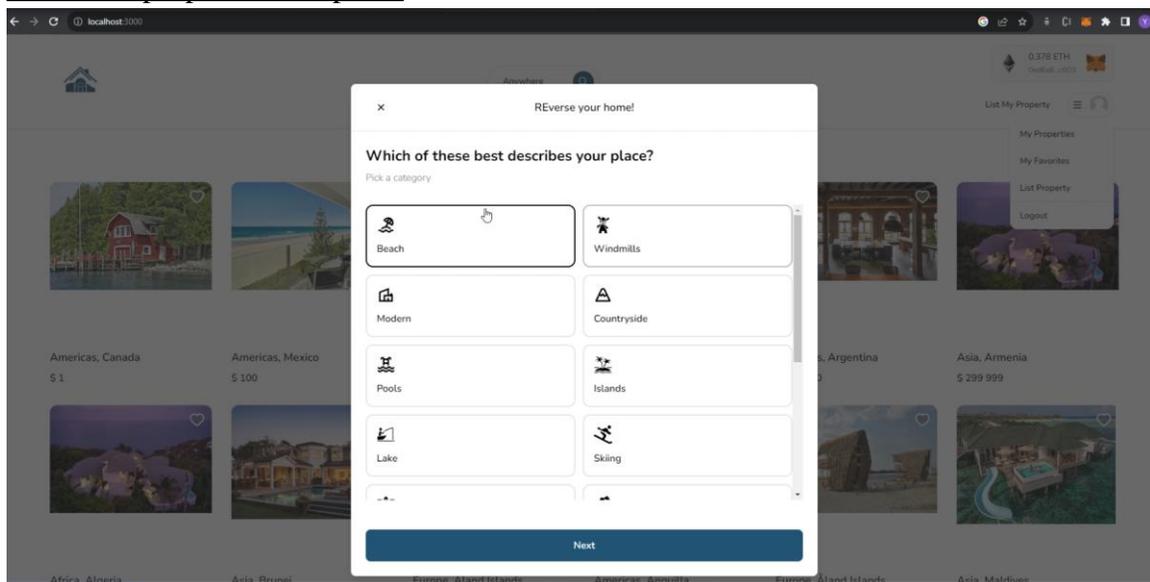
Lorsqu'un utilisateur se rend sur la page d'accueil de l'application, les différentes propriétés qui ont été listées par des utilisateurs vont être affichées sur cette page avec une image, l'emplacement et le prix.

## Authentication :



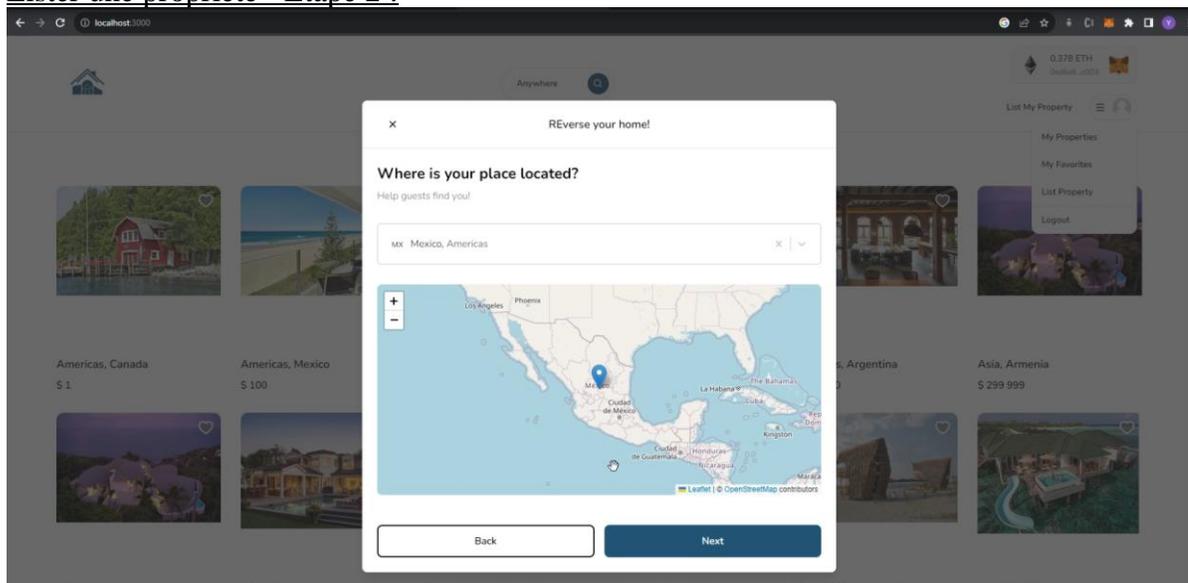
L'utilisateur pourra s'authentifier avec un courriel et un mot de passe d'un compte existant. De plus, il pourra aussi se connecter s'il le souhaite avec un compte google ou avec un compte GitHub. Pour terminer, il aura le choix de créer un compte s'il n'en possède pas déjà un.

## Lister une propriété – Étape 1 :



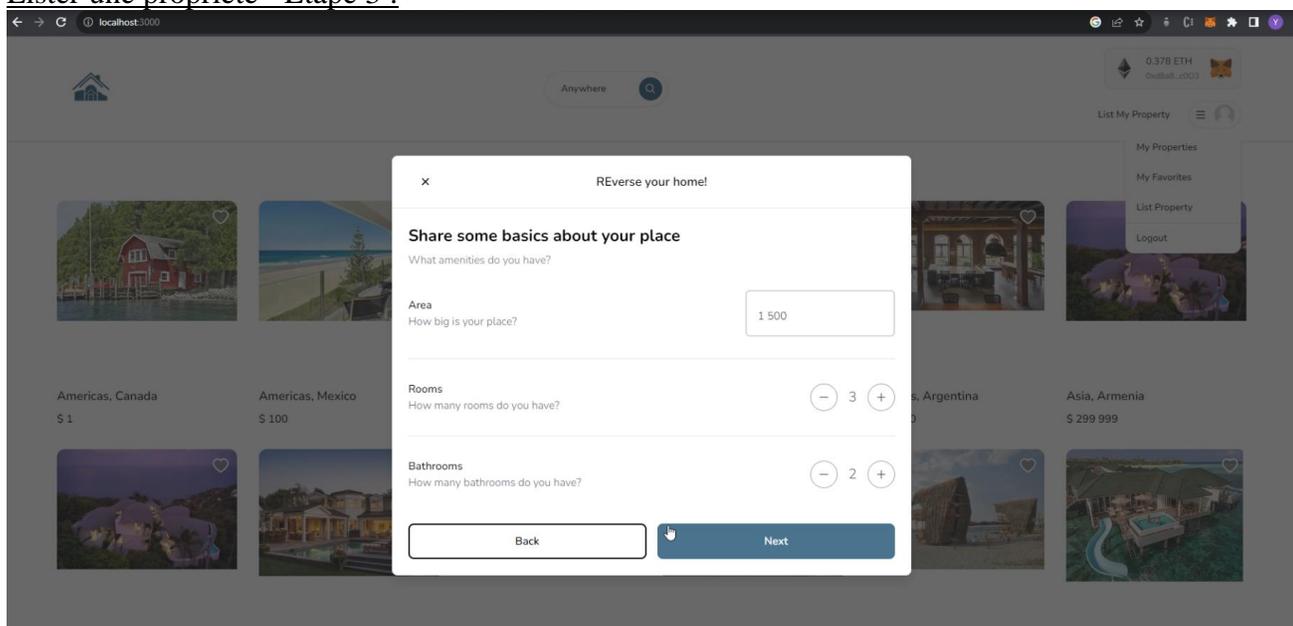
L'étape 1 pour lister une propriété est de choisir qu'elle est le type de maison qu'on veut lister.

### Lister une propriété - Étape 2 :



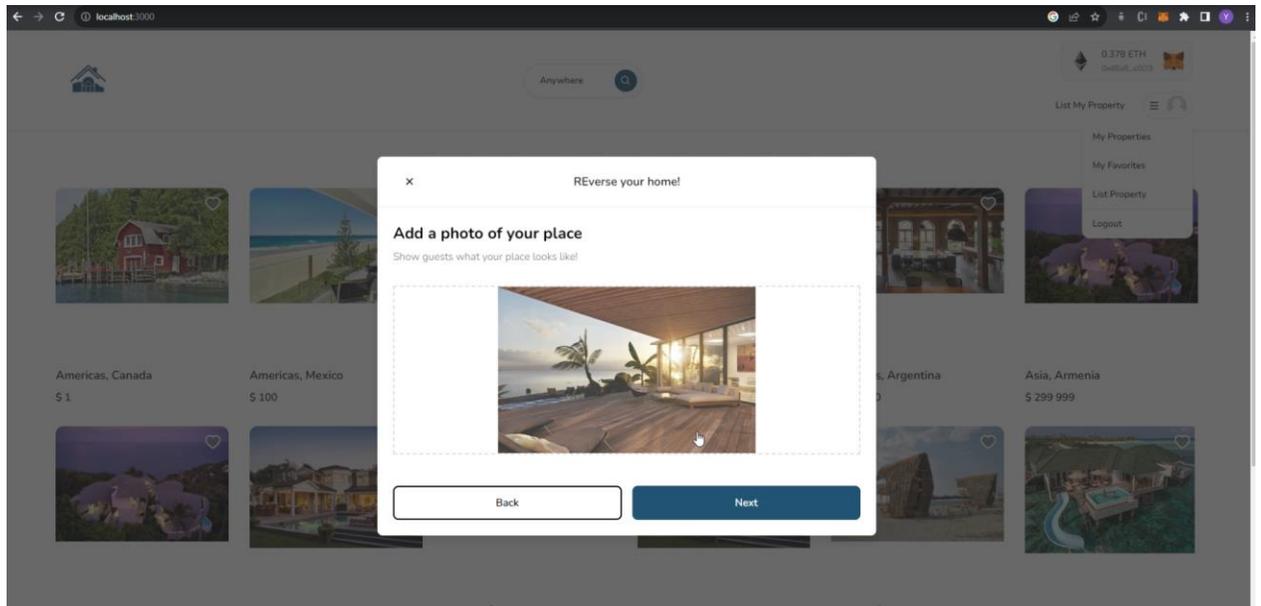
L'étape 2 consiste à choisir l'emplacement de la propriété qu'on veut lister.

### Lister une propriété - Étape 3 :



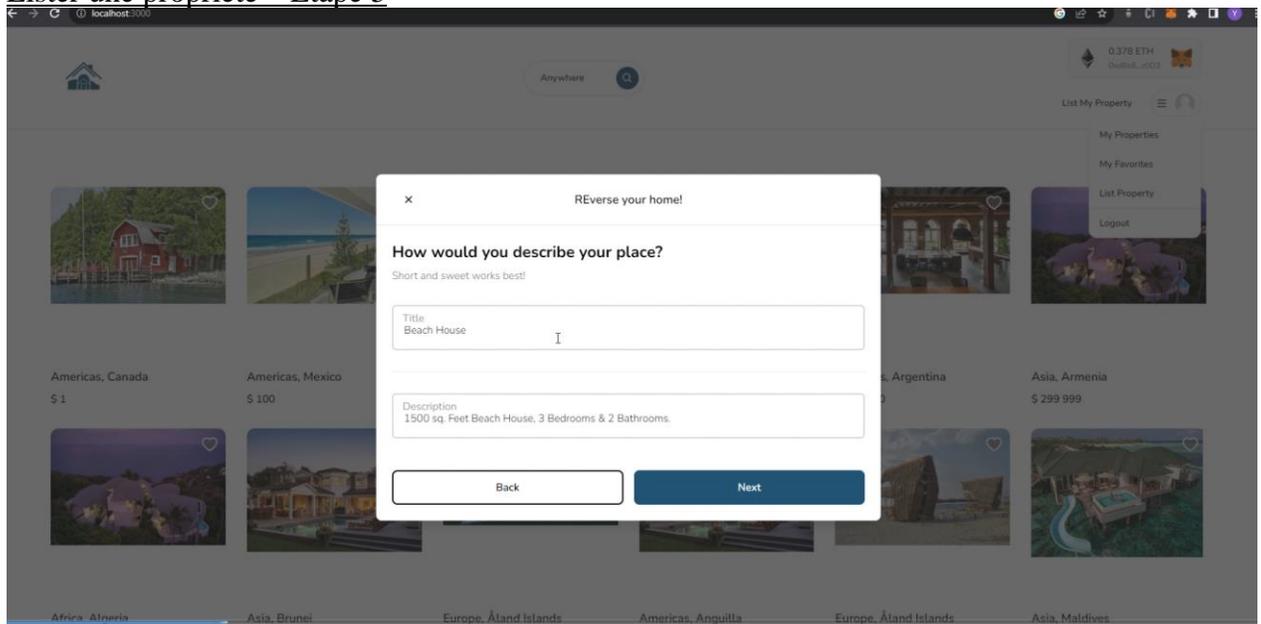
L'étape 3 pour lister une propriété est de choisir la grosseur de notre maison en pied carré, ainsi que de mettre le nombre de chambres et salles de bain.

### Lister une propriété - Étape 4 :



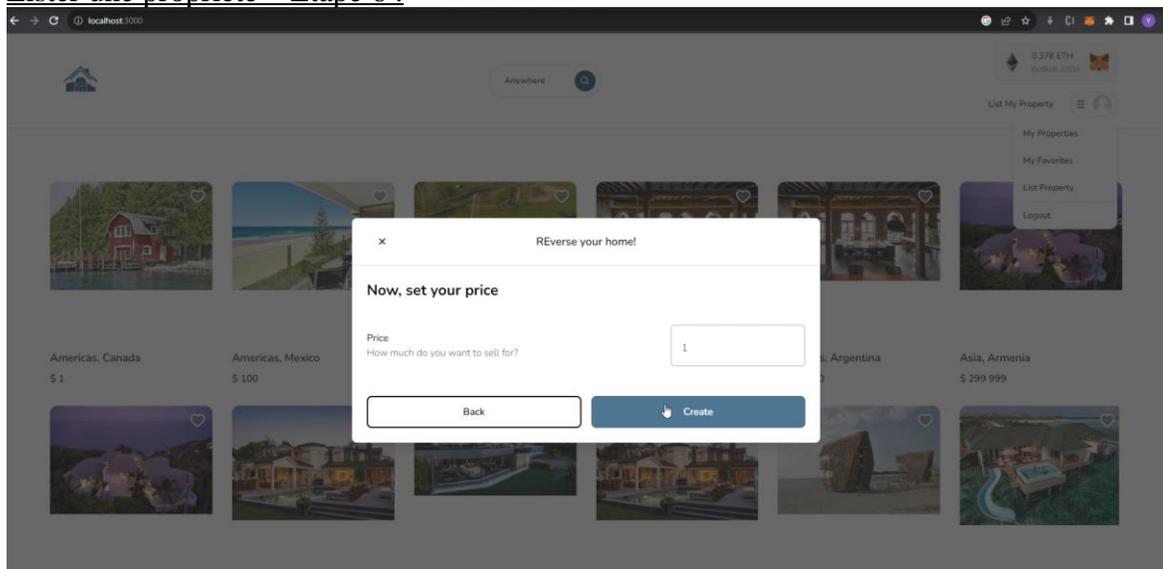
L'étape 4 pour lister une propriété consiste à ajouter une image représentative de la propriété qu'on souhaite lister.

### Lister une propriété – Étape 5



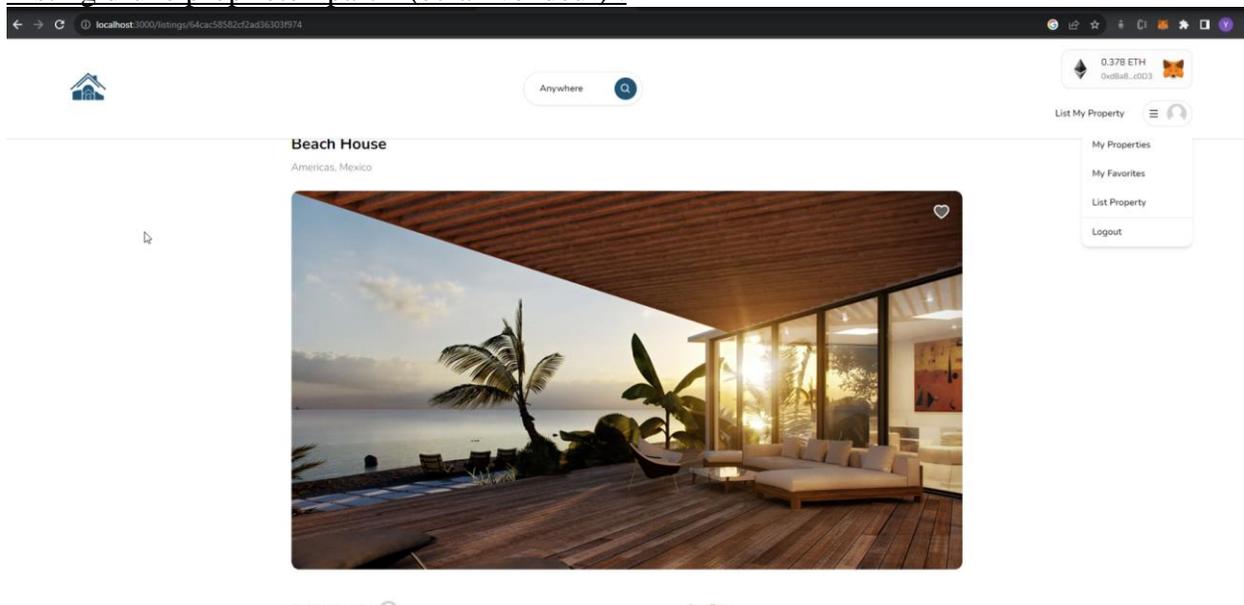
De plus, nous pouvons mettre un titre ainsi qu'une description à notre propriété.

## Lister une propriété – Étape 6 :



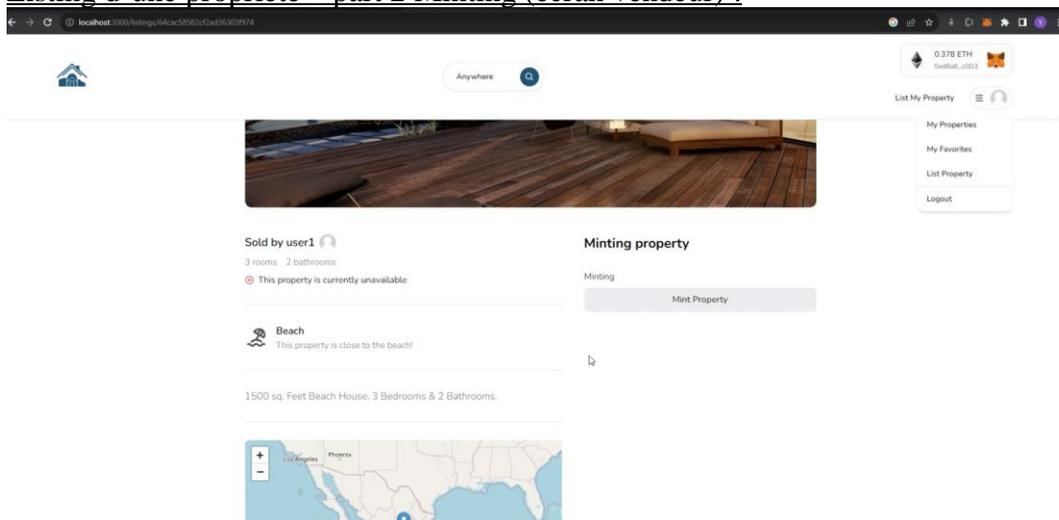
L'étape 6 est très importante, car c'est à cet endroit qu'on pourra mettre le prix qu'on souhaite lister notre propriété.

## Listing d'une propriété - part 1 (écran vendeur) :



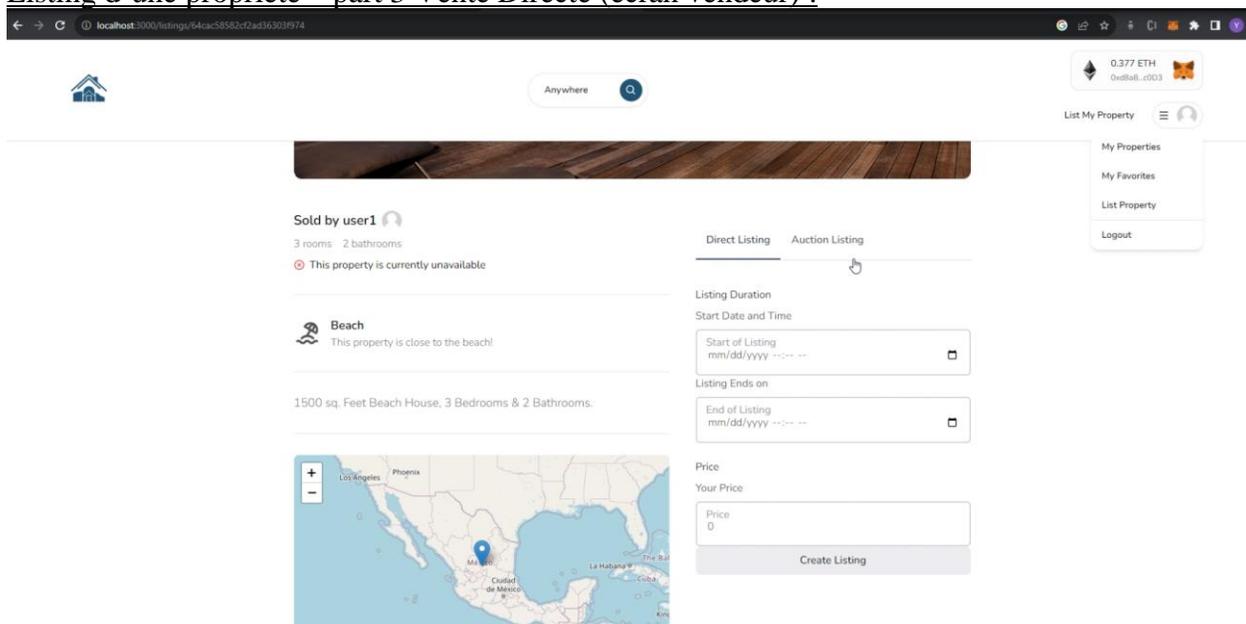
Une fois la mise en vente créée, on est amenée directement vers la page de notre propriété qui s'affiche avec toutes les informations que nous avons fournies.

### Listing d'une propriété – part 2 Minting (écran vendeur) :



On aura accès au bouton « Mint Property » qui va permettre de transformer le listing en un NFT unique

### Listing d'une propriété – part 3 Vente Directe (écran vendeur) :



Deux choix s'affichent au propriétaire du NFT, soit de lister la propriété avec un prix directement ou de créer une enchère. Dans le cadre d'une vente directe, un début et une fin sont demandés ainsi que le prix demandé.

### Listing d'une propriété – part 4 Vente aux enchères (écran vendeur) :

The screenshot shows a web browser window with a URL starting with 'localhost:3000'. The page features a search bar with 'Anywhere' and a user profile with '0.377 ETH'. The main content area is titled 'Sold by user1' and lists property details: '3 rooms 2 bathrooms', 'This property is currently unavailable', and 'Beach' with the note 'This property is close to the beach!'. Below this, it states '1500 sq. Feet Beach House, 3 Bedrooms & 2 Bathrooms.' and includes a map of Mexico. On the right, there are two tabs: 'Direct Listing' and 'Auction Listing', with the latter selected. Under 'Auction It', there are 'Begin' and 'End' date fields (mm/dd/yyyy). Below that, 'Set the Prices' section includes 'Allow bids starting from' with a 'Minimum bid price' field set to '0', and a 'Buyout price' field also set to '0'. A 'Create Auction Listing' button is at the bottom right.

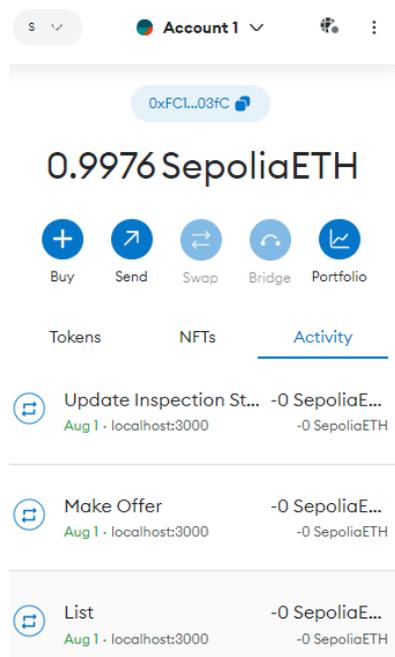
Deux choix s'affichent au propriétaire du NFT, soit de lister la propriété avec un prix d'achat directe ou de créer une enchère. Dans le cadre d'une enchère, un début et une fin sont demandés, le prix minimal ainsi que le prix d'achat direct sont aussi demandés.

### Listing d'une propriété – part 5 (écran acheteur) :

The screenshot shows the same web browser window from the seller's perspective. The main image is a large photograph of a modern interior with a wooden deck and a white sofa. The 'Sold by user1' section is visible. On the right, the 'Price' is listed as '0.01 ETH'. Below it, 'Bids starting from' is '0.001 ETH'. There are two buttons: 'Buy at asking price' and 'Place bid'. The 'Place bid' button is active, and a text input field contains '0.001'. The 'Buy at asking price' button is disabled.

Un acheteur pourra soit acheter au prix demandé, ou placer une offre plus haute que le prix minimal demandé.

### Metamask avec Sepolia Network :



Cette image représente les différents appels à la blockchain qui peuvent être lancés à partir de notre plateforme, faire une offre, lister une propriété, etc. De plus on peut voir que nous utilisons le network de Sepolia pour effectuer des tests.

## CHAPITRE 4

### VALIDATION ET VÉRIFICATION

#### 4.1 Plan de tests, évaluation de performance et résultat

Depuis le début de la création de la plateforme, nous avons constamment cherché à optimiser et améliorer nos performances, en particulier en ce qui concerne les coûts de transaction sur la blockchain. Au début, nos transactions avaient un gas limit de 2 480 897, ce qui entraînait des frais de transaction de 0.00372 ETH comme démontré sur la figure ci-dessous. C'était un coût significatif pour nos utilisateurs. Cependant, grâce à des efforts et à l'adoption de nouvelles méthodes, notamment le déploiement avec le proxy, nous avons réussi à réduire considérablement ces chiffres. Aujourd'hui, nous sommes fiers d'annoncer que notre gas limit a été réduit à 876 805, avec des frais de transaction de seulement 0.00127672 ETH. Cette amélioration permet d'offrir aux utilisateurs une expérience optimale et plus économique. Nous cherchons constamment à optimiser nos contrats intelligents en réduisant la complexité et l'adoption de proxies.

Transaction Hash:	0xa2716eba28713c745e3a563688ef19c25316b43456e0492df1196baba429fe76
Status:	Success
Block:	3864898 243932 Block Confirmations
Timestamp:	38 days 3 hrs ago (Jul-10-2023 05:00:12 PM +UTC)
From:	0xaBB6320CD93977BFbCf5b4dD4b1F28D550Eb7A0E
To:	[ 0xad44ca225473b69022fed05de921b810b81a5ab0 Created ]
Value:	0 ETH (\$0.00)
Transaction Fee:	0.00372134564885382 ETH 90.00
Gas Price:	1.50000006 Gwei (0.0000000150000006 ETH)
Gas Limit & Usage by Txn:	2,480,897   2,480,897 (100%)
Gas Fees:	Base: 0.00000006 Gwei   Max: 1.50000008 Gwei   Max Priority: 1.5 Gwei
Burnt & Txn Savings Fees:	Burnt: 0.0000000014885382 ETH (\$0.00) Txn Savings: 0.0000000004961794 ETH (\$0.00)
Other Attributes:	Txn Type: 2 (EIP-1559) Nonce: 0 Position In Block: 19

Figure 2 : Déploiement du contrat ERC821URISStorage sans proxy 2.4m de gas limit

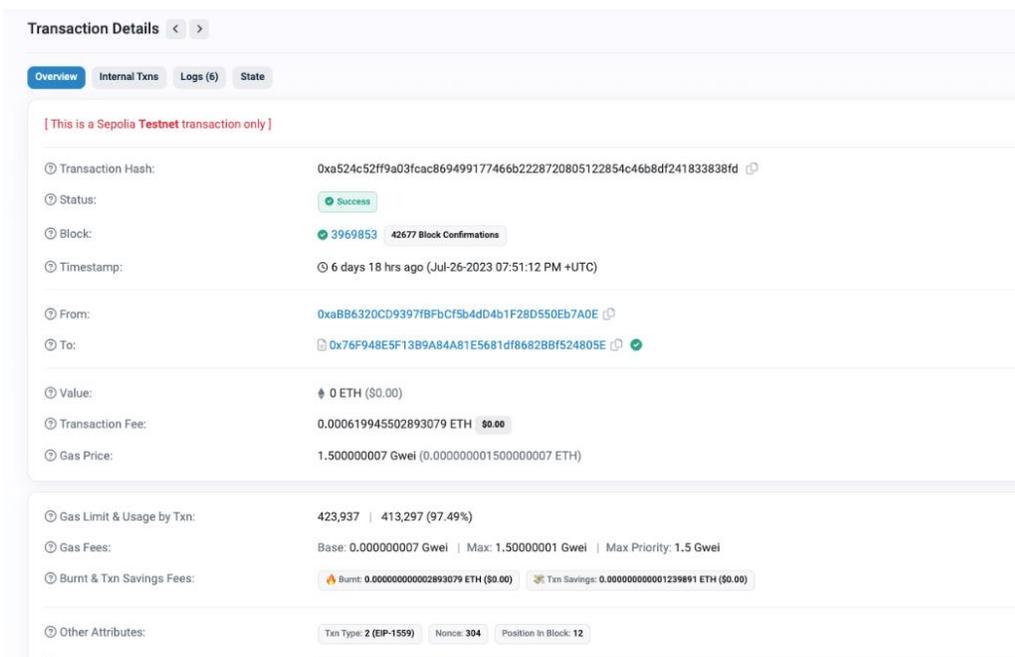


Figure 3 : Déploiement du contrat ERC821URISStorage avec proxy 423k de *gas limit*

## 4.2 Recommandations d'améliorations futures

Les améliorations futures envisagées pour la plateforme se concentrent principalement sur l'ajout de nouvelles fonctionnalités qui étaient prévus au départ du projet et l'introduction de nouvelles capacités pour améliorer l'expérience utilisateur. Voici un résumé des améliorations prévues :

### Processus KYC :

Bien que le processus d'approbation des utilisateurs soit actuellement automatisé, une mise en œuvre complète du processus KYC est envisagée pour garantir une vérification d'identité plus robuste.

### Intégration du standard ERC-1155 :

En raison des défis rencontrés avec le standard ERC-1155, des efforts supplémentaires seront déployés pour intégrer pleinement cette fonctionnalité, permettant ainsi la fractionnalisation

des propriétés, car cette fonctionnalité est très importante dans le cadre de la vision de notre projet.

#### Plateforme de prêt :

Les fonctionnalités permettant d'acheter une propriété avec un prêt ou d'emprunter de la cryptomonnaie en utilisant une propriété comme garantie nécessiteront la création d'une plateforme de prêt indépendante.

#### Validation par le notaire :

Bien que l'approbation de la vente soit actuellement automatisée, une collaboration avec des notaires est envisagée pour permettre une vérification et une approbation manuelles des documents, garantissant ainsi une plus grande sécurité et conformité.

#### Amélioration de l'interface utilisateur (UI) :

L'interface utilisateur actuelle sera revue et améliorée pour offrir une expérience plus intuitive et fluide. Cela inclura la simplification des processus, l'amélioration de la navigation et l'ajout de guides interactifs pour aider les utilisateurs à travers les différentes étapes.

#### Support pour d'autres portefeuilles de cryptomonnaies :

Bien que plusieurs portefeuilles soient déjà supportés, l'ajout de nouveaux portefeuilles populaires est envisagé pour offrir une plus grande flexibilité aux utilisateurs.

En guise de conclusion, ces améliorations visent à rendre la plateforme plus robuste, sécurisée et conviviale, tout en permettant d'offrir des nouvelles fonctionnalités qui offriront aux utilisateurs diverses façons d'investir.

## CONCLUSION

En conclusion, l'application Reverse permet de se positionner comme une solution révolutionnaire pour le marché immobilier traditionnel puisque celle-ci assure un certain niveau de centralisation grâce à l'adoption de la technologie blockchain. La tokenisation des propriétés via les NFTs promet une transparence accrue, une réduction des coûts et une accélération des transactions.

L'équipe a déployé beaucoup d'effort dans l'objectif d'optimiser les coûts de transaction. Les améliorations futures envisagées, quant à eux, témoignent de notre engagement et de notre volonté d'offrir une expérience utilisateur de qualité. Effectivement, plusieurs méthodes ont été détaillées telles que le déploiement avec un proxy. Ces méthodes innovantes démontrent une certaine capacité d'adaptation de la plateforme qui prouve le fait que l'application possède un large éventail de possibilité. D'ailleurs, les améliorations prévues telles que l'intégration complète du processus KYC, de la collaboration avec des notaires ou de l'amélioration de l'interface utilisateur, reflètent cette possibilité d'accentuer les retours positifs dans le secteur immobilier.

En somme, il est possible d'analyser le secteur immobilier au point de réaliser des points de comparaisons entre le système REverse et le système actuellement centralisé afin de vérifier si le fonctionnement théorique du produit est réellement applicable au monde d'aujourd'hui. Les avantages sont nombreux et nous sommes convaincus que cette source de technologie apportera une vision plus saine du monde dans son aspect économique.

## LISTE DE RÉFÉRENCES

- [1] Erdeljac, Antonio. 2023. « GitHub - AntonioErdeljac/next13-airbnb-clone ». In GitHub. En ligne. <<https://github.com/AntonioErdeljac/next13-airbnb-clone>>. 1 juin 2023
- [2] thirdweb-dev. 2023. « contracts/contracts/marketplace/english-auctions/EnglishAuctionsLogic.sol at main · thirdweb-dev/contracts ». In GitHub. En ligne. <<https://github.com/thirdweb-dev/contracts/blob/main/contracts/marketplace/english-auctions/EnglishAuctionsLogic.sol>>. 1 juin 2023
- [3] FAUCETLINK. 2023. « SEPOLIA ». In FAUCETLINK. En ligne. <<https://faucetlink.to/sepolia>>. 1 juin 2023
- [4] Thirdweb. 2023. « Templates ». In Thirdweb Portal. En ligne. <<https://portal.thirdweb.com/templates>>. 1 juin 2023
- [5] Wagmi. 2023. « useContractRead ». In Wagmi. En ligne. <<https://wagmi.sh/react/hooks/useContractRead>>. 1 juin 2023
- [6] Overleaf. 2023. « LaTeX, Evolved: The easy to use, online, collaborative LaTeX editor ». In Overleaf. En ligne. <<https://www.overleaf.com/>>. 1 juin 2023
- [7] Overleaf. 2023. « LaTeX, Evolved: The easy to use, online, collaborative LaTeX editor ». In Overleaf. En ligne. <<https://www.overleaf.com/>>. 1 juin 2023
- [8] IPFS. Date de publication non spécifiée. « IPFS ». In IPFS. En ligne. <<https://ipfs.tech/>>. 1 juin 2023

## ANNEXE I

### CU02 :

Présentement, l'approbation des utilisateurs sur la plateforme est réalisée automatiquement, évitant ainsi le passage par le processus KYC. La mise en œuvre de ce cas d'utilisation, en raison de sa complexité, aurait pu être l'objet d'un projet dédié. Il convient donc de souligner que ce cas d'utilisation a été mis de côté en raison des contraintes temporelles entourant la réalisation du projet.

### CU09 :

Lors de la phase d'implémentation, le standard ERC-1155 a présenté des défis en raison des difficultés de compatibilités avec les fournisseurs de portefeuilles, car les jetons de type "ERC-1155" n'étaient pas directement énumérables. Par conséquent, il était difficile de déterminer quel portefeuille détenait quelle quantité de jetons sans avoir recours à la surveillance des événements lors des transferts. Cette approche nécessitait d'écouter attentivement les événements de transfert pour suivre la quantité de jetons liés à un NFT.

Une option envisagée pour résoudre ce problème était la mise en place d'un "subgraph" au sein du protocole "The Graph". Cependant, l'intégration de cette solution introduisait un niveau supplémentaire de complexité dans son processus de mise en œuvre. Cette complexité était d'autant plus significative étant donné que la majorité des membres de notre équipe était peu familière avec les technologies du web 3.0. Face à ces défis et à la complexité accrue liée à l'adoption du standard ERC-1155 au sein de notre projet, nous avons opté pour d'autres approches afin de mieux répondre à nos exigences et à titre de complexité réduire du projet (implémentation).

