

# Rapport technique – Projet Web3

Sujet

**Application décentralisée de gestion et de traçabilité de ressources sur blockchain**

**Réalisé par le groupe : 12**

Ndiambe Gueye

Isabelle Marie Caroline Coly

Ibrahima Awl

**Projet académique**

**Encadré par :**

**M. Vangu Elie**

**Année : 2025-2026**

# 1. Introduction

Le projet **ResourceChain** s'inscrit dans le cadre du développement d'une application décentralisée (DApp) basée sur la blockchain. L'objectif principal est de concevoir un système permettant la **gestion décentralisée de ressources numériques tokenisées**, en respectant des contraintes métiers précises (limites, délais, échanges contrôlés).

La blockchain Ethereum a été choisie comme support technologique afin de garantir la transparence, la traçabilité et l'immuabilité des données. Le projet repose sur des **smart contracts**, un **frontend Web3**, et l'utilisation d'**IPFS** pour le stockage des métadonnées.

---

## 2. Cas d'usage et justification de la blockchain

### 2.1 Cas d'usage

ResourceChain permet de représenter des ressources numériques (exemples : maison, gare, bâtiment, actif symbolique) sous forme de **tokens non fongibles (NFT)**. Chaque ressource est unique, possède des attributs propres et peut être échangée entre utilisateurs selon des règles définies.

### 2.2 Justification de l'utilisation de la blockchain

L'utilisation de la blockchain est justifiée par les besoins suivants :

- **Unicité des ressources** : chaque ressource doit être identifiable et non duplicable.
- **Traçabilité** : l'historique des propriétaires doit être conservé de manière fiable.
- **Décentralisation** : aucune autorité centrale ne contrôle les échanges.
- **Sécurité et immuabilité** : les règles métiers sont inscrites directement dans le smart contract.

Ces exigences correspondent parfaitement au modèle des **NFT (ERC-721)** sur Ethereum.

---

## 3. Architecture globale du projet

### 3.1 Vue d'ensemble

L'architecture du projet est divisée en trois couches principales :

1. **Smart contract (Blockchain Ethereum)**
2. **Stockage décentralisé IPFS**
3. **Frontend Web3**

Chaque couche a une responsabilité bien définie afin de garantir la modularité et la maintenabilité du projet.

## 3.2 Technologies utilisées

- **Blockchain** : Ethereum (réseau local Ganache)
  - **Smart contracts** : Solidity
  - **Framework blockchain** : Truffle
  - **Frontend** : HTML, CSS, JavaScript, Web3.js
  - **Stockage off-chain** : IPFS (via Pinata)
  - **Environnement de test** : Ganache
- 

# 4. Smart Contract

## 4.1 Choix du standard

Le smart contract principal repose sur le standard **ERC-721**, plus précisément sur l'extension **ERC721Enumerable** fournie par OpenZeppelin.

Ce choix permet :

- l'unicité des tokens
- l'énumération des tokens possédés par un utilisateur
- la compatibilité avec l'écosystème Ethereum

## 4.2 Structure des données

Chaque ressource est représentée par une structure Solidity :

- `name` : nom de la ressource
- `resourceType` : type ou catégorie
- `value` : valeur associée
- `ipfsHash` : hash IPFS des métadonnées
- `previousOwners` : liste des anciens propriétaires
- `createdAt` : date de création
- `lastTransferAt` : date du dernier transfert

Cette structure permet de répondre aux exigences fonctionnelles du sujet.

---

## 5. Règles métiers implémentées

### 5.1 Tokenisation des ressources

Chaque ressource est créée via une fonction de mint. Une ressource correspond à un NFT unique, non fongible.

### 5.2 Limite de possession

Un utilisateur ne peut pas posséder plus de **4 ressources simultanément**. Cette règle est vérifiée directement dans le smart contract avant chaque mint.

### 5.3 Contraintes temporelles

Deux types de contraintes temporelles sont implémentées :

- **Cooldown entre deux créations** : un délai minimum de 5 minutes est imposé entre deux mint successifs pour un même utilisateur.
- **Lock après acquisition** : un délai de 10 minutes empêche le transfert immédiat d'une ressource nouvellement acquise.

Ces règles sont appliquées via des timestamps stockés on-chain.

### 5.4 Échanges de ressources

Les ressources peuvent être transférées entre utilisateurs à condition que :

- l'expéditeur soit le propriétaire
- le délai de verrouillage soit écoulé

Lors de chaque transfert, l'adresse de l'ancien propriétaire est ajoutée à l'historique `previousOwners`.

---

## 6. Stockage des métadonnées avec IPFS

### 6.1 Motivation du stockage off-chain

Le stockage direct de données volumineuses sur la blockchain étant coûteux, les métadonnées des ressources sont stockées **hors chaîne** sur IPFS.

Seul le **hash IPFS (CID)** est conservé dans le smart contract.

### 6.2 Implémentation

Les métadonnées sont stockées sous forme de fichiers JSON, contenant les informations descriptives de la ressource. Ces fichiers sont hébergés sur IPFS via le service **Pinata**.

Le lien entre la blockchain et IPFS est assuré par l'enregistrement du hash IPFS lors du mint.

---

## 7. Frontend Web3

### 7.1 Rôle du frontend

Le frontend permet aux utilisateurs de :

- visualiser leur compte actif
- créer de nouvelles ressources
- transférer des ressources
- consulter les ressources possédées

### 7.2 Interaction avec la blockchain

Le frontend utilise **Web3.js** pour communiquer avec le smart contract déployé sur le réseau Ganache.

Les appels sont réalisés de manière asynchrone et les transactions sont envoyées avec une limite de gas définie afin d'éviter les erreurs d'exécution.

---

## 8. Tests et validation

### 8.1 Tests fonctionnels

Les fonctionnalités suivantes ont été testées avec succès :

- création de ressources (mint)
- respect des limites de possession
- respect des délais temporels
- transfert de ressources
- mise à jour de l'historique des propriétaires

### 8.2 Environnement de test

Les tests ont été réalisés sur un réseau Ethereum local à l'aide de **Ganache**, permettant de simuler plusieurs comptes utilisateurs.

---

## **9. Conclusion**

Le projet ResourceChain démontre l'intérêt de la blockchain pour la gestion décentralisée de ressources numériques. Grâce à l'utilisation des NFT, d'IPFS et d'un frontend Web3, l'application répond pleinement aux contraintes techniques et fonctionnelles imposées.

Les règles métiers sont garanties par le smart contract, assurant sécurité, transparence et traçabilité. Ce projet constitue une base solide pour des évolutions futures, telles qu'un déploiement sur un réseau public ou l'intégration de portefeuilles comme MetaMask.