



RAPPORT DU PROJET 2 DE DAAR

# Decentralized Wikipedia

*Rédacteurs :*

Qiwei XIAN

Mehdi-Nassim KHODJA

Adel EL AMRAOUI

*Professeur :*

Prof. GUILLAUME-HIVERT

5 décembre 2020

## Table des matières

<b>1</b>	<b>Préface</b>	<b>2</b>
1.1	Objectif . . . . .	2
1.2	Techninologies utilisées . . . . .	2
<b>2</b>	<b>Architecture du projet</b>	<b>3</b>
2.1	Structure de l'application . . . . .	3
<b>3</b>	<b>Fonctionnalités de l'application</b>	<b>4</b>
3.1	Ajouter un article . . . . .	4
3.2	Consulter la liste des articles . . . . .	5
3.3	Rechercher un article . . . . .	6
3.4	Modification l'article . . . . .	7
<b>4</b>	<b>Difficultés techniques rencontrées</b>	<b>8</b>

# 1 Préface

## 1.1 Objectif

L'objectif du projet est de réaliser un wikipédia décentralisé. Tous les utilisateurs ont le droit d'ajouter des articles dans ce wikipédia, ainsi que de consulter et de modifier les articles existants.

## 1.2 Techninologies utilisées

1. React.js hook : Visualiser les résultats de nos fonctions back-end de recherche, d'ajout et de modification des articles.
2. API Web3 : Faire intéragir notre page web avec nos smart contracts. Cette API nous permet d'appeler les différentes fonctions qu'on a définit comme smart contract dans Wikipedia.sol.
3. Smart contract : Définir les classes qu'on veut stocker sur la blockchain ainsi que leurs opérations.
4. Ganache : Construire une blockchain locale. Ganache nous propose un environnement pour installer les smart contracts et générer des comptes virtuels.
5. Metamask : Plugin de chrome. Lorsque l'utilisateur envoie une transaction, Metamask affiche une fenêtre qui permet de signer la transaction.

## 2 Architecture du projet

### 2.1 Structure de l'application

1. `contracts/Wikipedia.sol` : Définition des méthodes de recherche, modification et d'ajout d'article dans notre wikipedia décentralisé.
2. `src/services/Ethereum.js` : Tous les appels des APIs web3 sont encapsulés dans ce fichier. Il permet les interactions entre notre partie front-end et nos smart contracts.
3. `src/App.js` : C'est la partie front-end de l'application, elle permet de générer dynamiquement le contenu des pages en utilisant des appels asynchrones.

## 3 Fonctionnalités de l'application

### 3.1 Ajouter un article

Pour ajouter un article il est nécessaire de spécifier un id entier et un contenu sous la forme d'une chaîne de caractères puis il faut cliquer sur "submit". Submit va faire appel à la fonction addArticle de Ethereum.js pour utiliser notre smart contract addArticle défini dans wikipedia.sol.

**Welcome to Decentralized Wikipedia**

account: 0x1fbadf7635de9fa45c4a45d4c2abf291b54110ab

[Home](#)   [Add an article](#)   [Modify articles](#)   [All articles](#)

**add article**

id:

This is article 101

FIGURE 1 – Saisir ID et l'article

wikipedia

Account 1

0x8775...06C1

合约交互

0

DETAILS DATA

GAS FEE 0.002124 无可用兑换率

Gas Price (GWEI) 20 Gas Limit 106200

AMOUNT + GAS FEE 0.002124 无可用兑换率

TOTAL 0.002124 无可用兑换率

拒绝 确认

All articles

Id	Article
0	This is yo ...
5	contract U ...
1	Lorem ipsu ...
3	Use our Wh ...
2	The progra ...
10	Welcome to ...
6	Fortmatic ...
4	Let's star ...
7	Sed ut per ...
101	This is ar ...

FIGURE 3 – Le résultat de l'addition

FIGURE 2 – Signer la transaction

### 3.2 Consulter la liste des articles

Chaque article est identifié par un Id sous la forme d'un entier et possède un contenu représenté par une chaîne de caractères. La liste des articles est récupérée en cliquant sur "All articles". All articles va faire un appel asynchrone à la fonction `getAllArticles` de `Ethereum.js` pour afficher la liste des articles.

**Welcome to Decentralized Wikipedia**

account: 0x1fbadf7635de9fa45c4a45d4c2abf291b54110ab

[Home](#)   [Add an article](#)   [Modify articles](#)   [All articles](#)

**Search article**

**All articles**

<b>Id</b>	<b>Article</b>
0	This is yo ...
5	contract U ...
1	Lorem ipsu ...
3	Use our Wh ...
2	The progra ...
10	Welcome to ...
6	Fortmatic ...
4	Let's star ...
7	Sed ut per ...
101	This is ar ...

FIGURE 4 – Afficher tous les articles existants

### 3.3 Rechercher un article

Pour rechercher un article on spécifie l'Id de l'article dans la barre de recherche puis on clique sur submit. Cette action va appeler la fonction `getArticleById` de `Ethereum.js` et va afficher le contenu de l'article qui possède l'Id spécifiée. Dans le cas où l'article n'existe pas elle affichera un contenu vide.

**Welcome to Decentralized Wikipedia**

account: 0x1fbadf7635de9fa45c4a45d4c2abf291b54110ab

[Home](#)      [Add an article](#)      [Modify articles](#)      [All articles](#)

**Search article**

Welcome to the DAAR project. The idea will be to implement a complete Wikipedia in a decentralized way, on Ethereum. This will have cool side effects, like not be forced to pay for servers.

FIGURE 5 – Chercher l'article par ID

### 3.4 Modification l'article

Pour modifier un article il suffit de cliquer sur "Modify articles" puis de saisir l'id de l'article et le nouveau contenu sous la forme d'une chaîne de caractères puis submit. Cela aura pour effet d'appeler de manière asynchrone la fonction `modifyArticle` de `Ethereum.js` qui va à son tour appeler le contrat `modifyContent` de `Wikipedia.sol` en précisant comme argument l'index de l'article correspondant et le nouveau contenu. La modification d'un article dont l'id n'existe pas n'aura donc aucun effet.

**Welcome to Decentralized Wikipedia**

account: 0x1fbadf7635de9fa45c4a45d4c2abf291b54110ab

[Home](#)
[Add an article](#)
[Modify articles](#)
[All articles](#)

**modify article**

id:

FIGURE 6 – Saisir ID et le nouvel article

MetaMask Notification

wikipedia

Account 1 → 0x8775...06C1

合约交互

0

DETAILS DATA

GAS FEE 0.00096 无可用兑换率

Gas Price (GWEI) 20 Gas Limit 48004

AMOUNT + GAS FEE 0.00096 无可用兑换率

TOTAL 0.00096 无可用兑换率

拒绝 确认

**Welcome to Decentralized Wikipedia**

account: 0x1fbadf7635de9fa45c4a45d4c2abf291b54110ab

[Home](#)
[Add an article](#)
[Modify articles](#)
[All articles](#)

**Search article**

Modify article 101

FIGURE 8 – Le résultat de la modification

FIGURE 7 – Signer la transaction



## 4 Difficultés techniques rencontrées

Pour les fonctionnalités d'ajout d'article et de modification d'article, parfois l'appel ne produit pas directement de demande de signature avec MetaMask ce qui nous contraint à recommencer une nouvelle fois la demande. En analysant l'exécution, on remarque pourtant que la fonction demandée est bien appelée malgré le fait que MetaMask n'envoie pas la demande de signature.

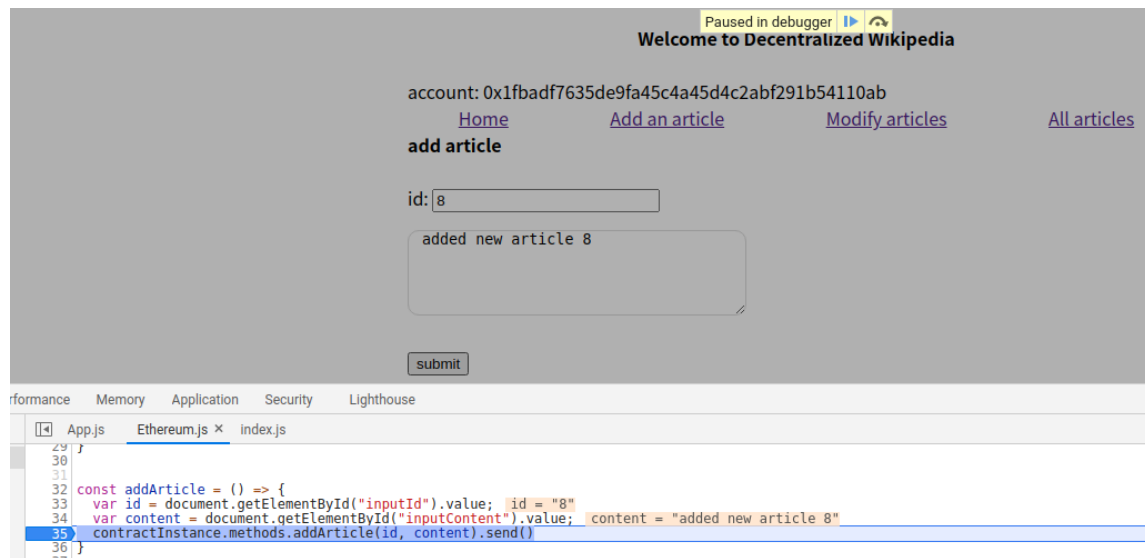


FIGURE 9 – La fonction de demande est bien appelée