

# Wikipédia décentralisé

*Projet basé sur le dépôt github suivant:*  
<https://github.com/ghivert/decentralized-wikipedia>



# Sommaire:

<b>Sommaire:</b>	<b>2</b>
<b>Introduction:</b>	<b>3</b>
<b>Wikipédia décentralisé : le parcours utilisateur</b>	<b>4</b>
<b>Structure du projet</b>	<b>4</b>
Contrats intelligents	5
Organisation du site web	5
<b>Difficultés rencontrées:</b>	<b>6</b>
<b>Conclusion:</b>	<b>6</b>

# Introduction:

La blockchain telle qu'on la connaît aujourd'hui est en réalité une technologie relativement récente.

En effet, les premiers travaux d'études concernant cette méthode de transmission et de stockage d'informations par blocs datent de 1991, et ont été réalisés par Stuart Haber et W. Scott Stornetta, deux cryptographes.

Il s'agit initialement de protéger des documents contre toute forme de modification ou d'antidatage via un procédé à base de signatures cryptographiques.

Cependant, cette technologie ne connaîtra un véritable essor qu'à partir de 2008, lorsqu'une personne, ou un groupe de personnes, opérant sous le pseudonyme de [Satoshi Nakamoto](#) lance la monnaie virtuelle [Bitcoin](#) basée sur un concept de chaîne de blocs.

Un des points négatifs de cette technologie et qui devient de plus en plus notable de nos jours est la consommation énergétique qu'elle engendre.

Cette technologie est cependant déjà très utilisée dans certains domaines tels que la crypto-monnaie (le bitcoin étant l'exemple le plus connu), des bases de données sécurisées ou divers systèmes répartis, car elle permet des transactions publiques et infalsifiables.

# Wikipédia décentralisé : le parcours utilisateur

Le site web développé est très basique. Il consiste en une version très simplifiée du célèbre site Wikipedia.

La racine du site mène vers la page d'accueil dans laquelle plusieurs options s'offrent à nous:

- Voir tous les articles
- Voir un article via son ID
- Voir tout l'historique d'un article via son ID
- Ajouter un article
- Modifier un article

Chaque page est très basique, on n'aura donc pas la possibilité de voir ou choisir de titre d'article, mais on pourra les identifier par leur ID.

Etant donné que tous les articles sont stockés via un système de blockchain, il est ici impossible de supprimer un article. On pourrait cependant imaginer un système qui marque un article comme étant supprimé pour ne plus qu'il apparaisse dans les résultats de recherche, mais il resterait présent dans l'historique de la blockchain.

## Structure du projet

Pour réaliser ce projet, nous nous sommes appuyé sur les technologies suivantes:

- Reactjs pour le front, interface utilisateur permettant d'interagir avec les blockchains
- Ganache pour la blockchain, il s'agit d'un logiciel qui permet de créer une blockchain en local
- MetaMask pour gérer les comptes Ethereum et faire des transactions
- Solidity qui est un langage de programmation spécialisé dans la gestion de contrats intelligents

## Contrats intelligents

```
// mise a jour d'un article
function updateArticle(uint index, string memory content) public {
    Article memory updatedArticle = Article(content);

    // ajout de l'ancienne version dans l'historique
    historyById[id][nbModifsById[index]] = articlesById[index];
    nbModifsById[index] += 1;

    // mise a jour de l'article dans la liste d'articles
    articlesById[index] = updatedArticle;
}
```

Le langage Solidity a une syntaxe proche de celle de langages que nous connaissons bien tels que C et JAVA ce qui a permis une prise en main rapide. On voit sur l'exemple ci-dessus que le code est facilement compréhensible.

En s'appuyant sur le code fourni, nous avons écrit les méthodes nécessaires à l'ajout et à la mise à jour d'article ainsi que l'accès à l'historique des articles.

## Organisation du site web

Nous avons commencé par comprendre le code fourni puis nous avons ajouté un handler dans `NewArticle` pour transmettre l'article écrit au code Solidity. Nous avons ensuite repris le modèle pour les autres fonctions en adaptant à chaque fois l'affichage html retourné et les potentiels handlers (pour `UpdateArticle` par exemple).

La majorité des pages possède également des liens vers les autres pages du site, et `AllArticles` permet de visiter les pages de mise à jour de chaque article ainsi que leur historique.

Exemples de fonctions :

- `AllArticles` affiche tous les articles et leur id
- `Article` affiche un seul article
- `UpdateArticle` met à jour un article à partir de son id
- `History` affiche toutes les versions antérieures d'un article et leur numéro de version

On peut accéder aux méthodes instanciées via solidity avec l'instruction suivante :  
`contract.methods.articleContent(arguments).call(callback);`

## Difficultés rencontrées:

Nous avons tout d'abord rencontré des difficultés à comprendre ganache, en effet, l'exemple de base avec le gaz est un peu perturbant, et nous avons eu du mal à comprendre comment s'en servir pour gérer des articles.

Nous avons également rencontré des difficultés pour comprendre précisément le système de contrats avec Solidity et comment gérer le lien avec les différents composants du projet.

## Conclusion:

Ce projet nous a permis de découvrir la blockchain, et d'apprendre à l'utiliser. Même si nous avons rencontré des difficultés lors de l'utilisation des différentes technologies, nous avons pu néanmoins, grâce à ce projet, mieux comprendre les utilisations, les avantages et les contraintes liées à l'utilisation de cette technologie.