Certification de diplômes via la blockchain

Bachelor développement logiciel, 2e année, 2023-2024

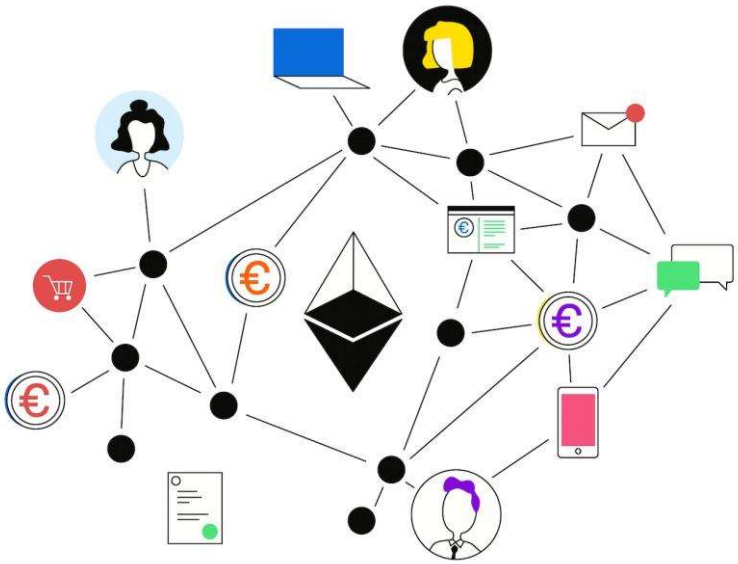

Figure 1 - Image page de garde - https://www.bitpanda.com/academy/fr/lecons/quest-ce-quethereum/

Travail groupe : 2281.1 Projet P2 SA IL

Auteurs : Villarejo Maxime - Vivone Tom - Bozier Luan

Date : 19.09.2023 – 22.01.2024

Ecole : Haute Ecole Arc, HES-SO

Encadrants : Ninoslav Marina - Benoit Le Callennec - Nicolas Minnig

Table des matières

1 Introduction

1.1 Contexte

Le module P2 du semestre d'Automne, de notre cursus de bachelor d'informatique de programmation logiciel, nous a donné l'opportunité d'explorer et de se spécialiser dans des domaines novateurs, et émergents.

Dans le cadre de ce projet, réalisé en équipe de trois étudiants, nous avons bénéficié d'une totale liberté de choix tant pour le sujet que pour la technologie.

Nous avons fait le choix audacieux de nous aventurer dans un domaine nouveau pour l'ensemble des membres du groupe, en exploitant des technologies récentes. Notre exploration s'est orientée vers le sujet captivant de la blockchain et des "smart contracts" basé sur Ethereum afin de mettre en place un système de certification de diplômes.

1.2 Description du sujet

1.2.1 Blockchain

Ceci est une brève introduction (simplifiée et vulgarisée) à cette technologie, que nous détaillerons davantage au chapitre 3.4.

Pour tout type de transactions, un tiers de confiance est requis en tant qu'intermédiaire entre le débiteur et le bénéficiaire. Prenons l'exemple d'un virement bancaire, la banque assume le rôle d'intermédiaire, contrôlant que tout est en règle et prélevant une commission.

La technologie des chaînes de blocs (« blockchain ») représente une alternative aux tiers de confiance traditionnels. On appelle le nouvel intermédiaire « vérification collective de l'écosystème », fonctionnant selon le principe simple suivant :

1. Une personne A (un « nœud ») possède un fichier de transactions sur sa machine (un « registre »).
2. Un certain nombre de comptables (des « mineurs ») possèdent également ce registre (« distribué »).
3. A effectue une transaction.
4. Ceci notifie tous les comptables, qui se précipitent pour être les premiers à vérifier que la transaction est valide.
    a. Cela leur permet de gagner leur salaire (« Cryptomonnaies »).

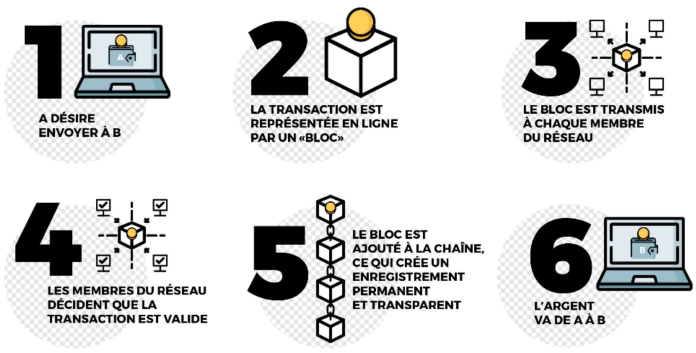

Figure 2 – Schéma simple explicatif d'une blockchain - https://www.digitvalue.fr/2018/07/24/la-blockchain-7-points-pour-faire-le-point/

Ainsi, les transactions sont collectivement vérifiées, assurant une sécurité, une rapidité et une traçabilité élevées dans le réseau. Enfin, ce principe n'est pas limité qu'à la finance ; il peut être étendu à toute transaction nécessitant traçabilité et visibilité, comme la signature de contrats, la vérification de la provenance d'un produit, la certification de documents officiels ou encore la mise en place d'une plateforme de vote.

1.3 Problématique actuelle

Dans notre démarche de création, nous avons identifié une problématique significative liée à l'authenticité des diplômes universitaires. Actuellement, un diplôme de bachelor se présente sous forme de documents papier officiels. Cependant, cette méthode pose un défi majeur : la facilité de falsification. En effet, un simple bout de papier ne fournit aucune garantie tangible de son authenticité, et il repose souvent sur la bonne foi de l'employeur qui, en cas de doute, doit contacter l'établissement émetteur pour vérifier la validité du diplôme.

C'est dans ce contexte que notre projet prend tout son sens. En exploitant la technologie de la blockchain et les contrats intelligents (smart contracts), notre objectif est de créer une application web pour palier à ce problème. Cette plateforme permettra d'émettre les diplômes de bachelor sous forme de métadonnées, et chaque transaction liée à ces diplômes sera signée numériquement par les parties autorisées, garantissant ainsi leur légitimité. Les diplômes, une fois validés, seront affichables sur notre application, sous forme de texte, grâce à un hash de transaction.

Ce procédé offre une transparence accrue. Un employeur, en cliquant sur le lien associé au diplôme, pourra vérifier instantanément la validité du document, ainsi que les détails de la transaction et les parties ayant approuvé le diplôme. Cette approche innovante vise à simplifier et à sécuriser le processus de vérification des diplômes, offrant ainsi une solution fiable et accessible à tous les acteurs concernés.

2 Abstract

Dans un contexte où l'authenticité des diplômes universitaires est de plus en plus remise en question, notre projet, réalisé dans le cadre du module "FB_2281.1 - Projet P2 SA IL 2023-2024", propose une application utilisant les technologies de la blockchain Ethereum pour la certification des diplômes universitaires.

En exploitant un système de portefeuille numérique, spécifiquement MetaMask, lié à la blockchain et à notre application, nous avons permis la signature numérique des données des diplômes. Le processus commence par le remplissage d'un formulaire par les responsables des établissements d'enseignement, où ils saisissent les informations essentielles du diplôme. Ces données sont ensuite transmises sur la blockchain. Les responsables d'authentification des diplômes récupèrent les diplômes bruts sur la blockchain pour les signés numériquement, assurant ainsi l'intégrité et l'authenticité des diplômes. Une fois ces étapes complétées, les diplômes deviennent des enregistrements sécurisés et transparents sur la blockchain. Ils sont non-modifiables ni supprimables et accessibles à tous sur notre application. Au moyen d'un code QR un employeur peut instantanément vérifier la validité du diplôme et les détails de la transaction d'autorisation via un lien direct ou un QR code.

3 Etat de l'art

Dans le domaine relativement récent et innovant de la certification de documents grâce à la technologie blockchain, il est important de noter que malgré quelques initiatives notables, l'état de l'art demeure relativement limité en raison de la nouveauté du champ. Les résultats de nos recherches ne nous ont pas permis de confirmer si certaines écoles ou d'autres structures ont déjà adopté ce système. Malgré cela, certaines avancées ont déjà été réalisées, et nous allons les présenter.

Solutions actuelles et projets existants :

Signature Chain (SIGN) (2022) :

Signature Chain (SIGN) révolutionne la certification de documents en décentralisant le processus grâce à la blockchain. La plateforme, conçue par Christophe Verdot de Digital Chain LTD, utilise la signature numérique des utilisateurs via une fonction de hachage cryptographique. Cette approche garantit la sécurité et la simplicité dans la certification de divers actifs numériques.

SIGN trouve des applications variées, de la dématérialisation des diplômes à la certification médicale. Les institutions éducatives peuvent enregistrer des diplômes avec un hachage unique sur la blockchain, assurant la vérification facile de l'authenticité. Les groupes médicaux peuvent également émettre et vérifier des certificats médicaux, garantissant la validité des dossiers tout au long de la vie.

La plateforme s'étend à des domaines tels que l'enregistrement de brevets, la signature numérique de contrats, la génération simplifiée de factures, la certification d'e-mails cruciaux, et bien plus encore. En résumé, Signature Chain offre une solution complète, améliorant la sécurité et l'efficacité de la certification décentralisée dans divers secteurs

Dans cette partie, il faut détailler les contributions apportées par rapport à l'état de l'art (en plus d'améliorer ses propres connaissances et d'acquérir de l'expérience).

https://www.signature-chain.com/

Blockchain-based certificate authentication system with enabling correction (Dept. of Computer Science and Engineering Southeast University – Bangladesh) **:**

En février 2023, des étudiants de l'Université du Sud-Est ont mis en place un projet visant à valider l'authenticité des certificats scolaires et à prévenir la falsification (environ 2 millions de faux diplômes aux États-Unis). Voici un schéma (voir figure 3) tiré de leur rapport décrivant leur méthodologie :

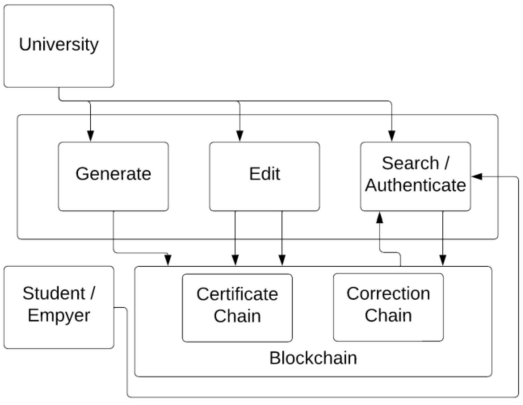

Figure 3 – Schéma projet southeast university - https://arxiv.org/pdf/2302.03877.pdf

En résumé, les personnes autorisées (administrateurs de l'université) génèrent des documents et les certifient dans la blockchain. Les étudiants et/ou employeurs peuvent visualiser le diplôme authentifié via une adresse (hash de transaction) ou un QR code.

Ils ont utilisé les technologies suivantes :

- Solidity (smart contracts) + remix (déploiement des smart contracts)
- Réseau Goerlitestnet (blockchain de test)
- Portefeuille Metamask pour les transactions
- Web app frontend

https://arxiv.org/pdf/2302.03877.pdf

Conclusion:

Tout au long de ce projet, notre objectif sera d'explorer ces différentes technologies pour élaborer un produit final novateur, offrant une solution complémentaire aux écoles désireuses de renforcer l'authenticité de leurs diplômes. Le projet de l'Université du Sud-Est constitue une source précieuse d'inspiration, nous motivant à développer une solution unique tout en tirant parti des enseignements de l'état actuel de l'art. Nous sommes déterminés à apporter notre propre contribution à ce domaine.

4 Analyse

Le projet vise à développer une application blockchain pour la certification numérique de documents. L'objectif est de fournir une solution plus sécurisée et économique que les systèmes existants.

En utilisant la technologie des smart contracts sur Ethereum, le projet promet d'améliorer la transparence et l'efficacité des processus de certification.

La motivation principale est de réduire les coûts et les délais associés à la certification traditionnelle, tout en augmentant la sécurité grâce à la blockchain. Cela devrait aboutir à une plateforme fiable et facile à utiliser pour les utilisateurs finaux, offrant un accès rapide à des services de certification vérifiables et immuables.

4.1 Faisabilité

4.1.1 Contraintes techniques

Discuter des défis techniques spécifiques liés à la blockchain, comme le développement de smart contracts, l'intégration avec Ethereum, et la gestion de la sécurité.

- **Développement avec les Smart Contracts** : C'est le point clé de notre projet, le but et la raison. La programmation qui se fait avec Solidity, grâce à son langage proche du C et du C ++ il est relativement simple à aborder et à coder. Malgré, tout il faut bien comprendre ce qui agit derrière. Lorsqu'on compile, on ne compile pas un simple programme sur un ordinateur mais sur l'EVM. C'est une technique un peu différente de la programmation enseignée en cours qui requiert quelques recherches initiales pour comprendre ce que l'on fait.
- **Déploiement des Smart Contracts** : Une fois compiler l'idée est de déployer ces Smart Contracts sur la blockchain. Mais cela requiert des outils tiers pour le déploiement. Il est idéal de se familiariser avec l'environnement avant de passer sur un réseau live. C'est pourquoi il est recommandé de faire des tests sur des blockchains en local par exemple Ganache. Cela offre un environnement sûr et contrôlé avant de les déployé sur le réseau principal.
- **Gestion de la sécurité :** Une des points les plus importants du projet est la gestion de la sécurité. Le sujet sensible que l'on traite implique de bien comprendre ce que l'on fait. Malheureusement bien que nous puissions faire des recherches et appliquer un maximum des informations apprises, il est difficile dans le temps imparti de s'assurer que toutes étapes de sécurité son très bien sécurisées.

4.1.2 Faisabilité économique :

- **Coût de développement :** Le développement d'une application sur la blockchain implique des coûts dans le cas où l'on travaille sur le mainnet. Il y a les frais de transactions pour les interactions avec les smart contracts ou encore les frais de déploiements de smart contracts. Cependant si l'on travaille avec un testnet, les cryptomonnaies non pas de valeur monétaire. On peut en obtenir gratuitement pour effectuer des tests et travailler avec la blockchain. Alors dans l'ensemble notre projet ne coutera rien du moment que l'on reste dans un réseau de test.

4.2 Risques

4.2.1 Risque important : Coûts de transaction prohibitifs

- **Description** : Le projet vise à offrir une solution de certification numérique de documents plus économique que les systèmes actuels. Cependant, il existe un risque que les frais de transaction sur la blockchain Ethereum soient plus élevés que les coûts des services existants, ce qui pourrait rendre notre solution moins compétitive.
- **Probabilité** : Moyenne, dépendante des fluctuations du marché et de l'évolution des frais sur la blockchain.
- **Impact** : Grave, car cela pourrait compromettre l'attrait économique de notre projet pour les utilisateurs potentiels.
- **Mesures de prévention :**
  - Surveillance régulière des frais de transaction sur Ethereum.
  - Exploration d'alternatives comme les solutions de second layer pour réduire les coûts (d'autres blockchain basé sur Ethereum).
  - Ajustement du modèle économique pour intégrer des mécanismes de compensation ou des options de tarification flexibles.

4.2.2 Risque critique - Sécurité des Smart Contracts

- **Description** : Les smart contracts étant au cœur de l'application, une faille de sécurité pourrait compromettre l'intégralité du système et les données utilisateurs.
- **Probabilité** : Probable
- **Impact** : Grave
- **Mesures de prévention :** Implémentation de protocoles d'audit de sécurité rigoureux, réalisation de tests unitaires et d'intégration pour détecter les vulnérabilités.
- **Mesures de correction :** Mise en place d'un plan d'urgence réactif pour appliquer des mises à jour correctives ou des patchs de sécurité.

4.2.3 Risque acceptable - Dépendance aux services tiers

- **Description** : La dépendance à des fournisseurs externes pour des services clés peut introduire des points de défaillance uniques.
- **Probabilité** : Probable
- **Impact** : Faible
- **Mesures de prévention** : Établissement de partenariats avec divers fournisseurs, mise en place d'une infrastructure redondante.
- **Mesures de correction** : Développement d'un processus agile pour basculer rapidement vers des fournisseurs alternatifs en cas de défaillance.

4.2.4 Risque modéré - Interaction Utilisateur avec les Smart Contracts

- **Description** : Les utilisateurs sont responsables de la création de leur portefeuille Metamask et de la gestion de leurs clés privées. Une mauvaise gestion pourrait limiter leur capacité à interagir avec les smart contracts.
- **Probabilité** : Modérée
- **Impact** : Modéré
- **Mesures de prévention :** Utilisation d'un tiers de confiance pour la récupération des adresses chez les clients, support technique pour la création et la gestion du portefeuille.
- **Mesures de correction** : Assistance pour la récupération de compte, en collaboration avec les services de support Metamask, tout en respectant la confidentialité et l'autonomie des utilisateurs.

4.2.5 Risque - Fiabilité de l'hébergement web

- **Description** : Notre application web repose sur un hébergement centralisé, ce qui pourrait présenter des points de défaillance uniques.
- **Probabilité** : Modérée
- **Impact** : Grave (si l'hébergement est compromis, cela pourrait affecter l'accessibilité et la fonctionnalité du service).
- **Mesures de prévention :** Utilisation de services d'hébergement réputés avec des garanties de temps de fonctionnement élevé.
- **Mesures de correction :** Stratégie de redondance, un plan de récupération après sinistre pour restaurer rapidement les services en cas d'interruption.

4.3 Planification

Projet P2 - Haute école Arc

Planifié :
Réel :

NY

CertifDocs

Année : 2023 2024

étudiant: Mois: septembre octobre novembre décembre janvier février

No.	TÂCHE	Semaine	étudiant:	Fait
	Plannification + Cahier des charges			
1	Rédaction du cahier des charges	1	Tous	Oui
2	Création de la planification	1	Tous	Oui
	Conception de la certification			
7	Mettre en place un environnement de travail	1	Tous	Oui
8	Se familiariser avec les smartcontracts et la blockcha	2	Tous	Oui
9	Tester l'utilisation d'un smartcontract simple en local	1	Tous	Oui
10	Développer les smartcontracts de signature	5	Luan	Oui
11	écrire un script de validation de certification	1	Luan	Oui
12	Passage à une blockchain de test publique	2	Tom	Oui
	Développement de la web app			
13	Mettre en place un environnement de travail en local	1	Tous	Oui
14	Gérer l'autentification des connexions à la web app	2	Tom	Oui
15	Dév. Front-End.(Envoi/Réception des données avec la blockchain	3	Tom, Maxime	Oui
17	Valider le hash et vérifier le hash de transaction	1	Maxime	Oui
18	Envoyer un mail avec le docuemnt signé	1	-	non
	Phase de tests			
19	Tests avec utilisateurs + Modif. Si besoin	2	tous	
	Rédaction du rapport			Oui
20	Rédaction du rapport	4	tous	
	Présentation			
21	Réalisation de la présentation	2	tous	

Figure 4 – Diagramme de Gantt

4.4 Technologies utilisées

4.4.1 Fonctionnement de la blockchain

Il est intéressant d'utiliser un schéma pour cette partie afin de mieux se représenter le fonctionnement, comme une image vaut milles mots voici le schéma explicatif de la blockchain en anglais :

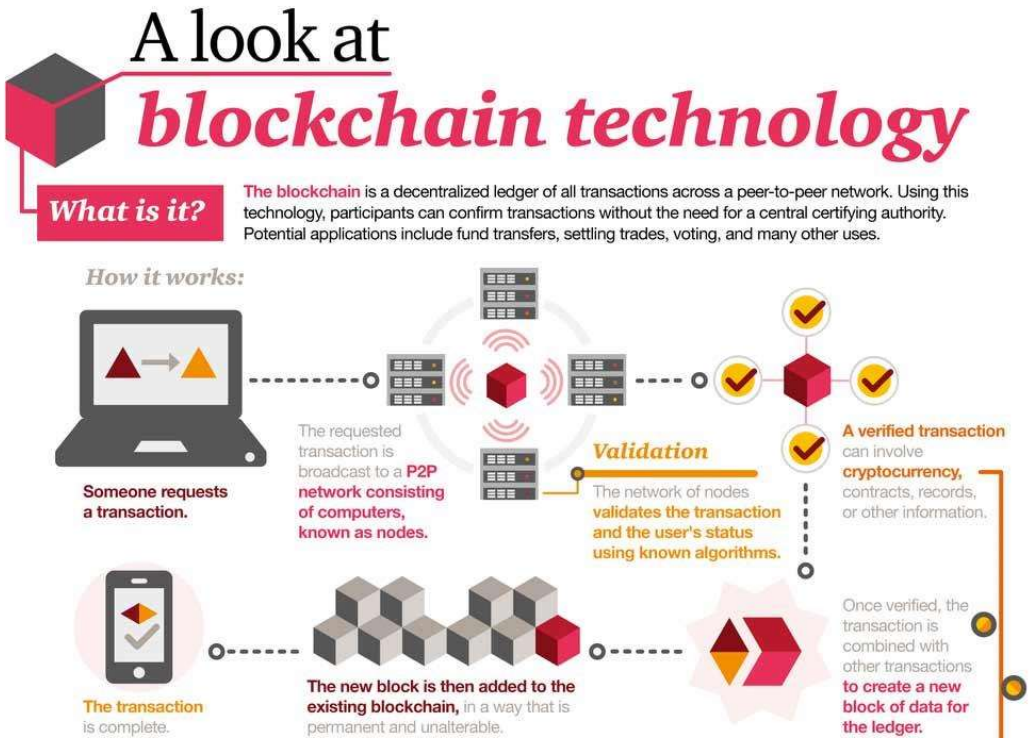

Figure 5 – Explication de ce qu'est la blockchain et comment ça marche

4.4.2 MetaMask

MetaMask est un portefeuille de cryptomonnaies logiciel utilisé pour interagir avec la blockchain Ethereum. On accède à celui-ci via un navigateur ou sur son application mobile. MetaMask permet de stocker et gérer des clés de comptes, d'effectuer des transactions, d'envoyer / recevoir des cryptomonnaies. Mais pas seulement, il peut aussi interagir avec des applications liées à la blockchain comme la nôtre grâce à des codes JavaScript, on peut faire des requêtes d'actions, des signatures ou encore des transactions.

4.4.3 Ganache

Ganache est une blockchain personnelle pour le développement rapide d'applications distribuées Ethereum et Filecoin. Elle peut être utilisée toute au long du cycle de développement, ce qui permet notamment de développer, déployer et tester nos applications dans un environnement sûr et déterministe.

Ganache facilite grandement la vie au développeur avec ses fonctionnalités. Elle permet de facilement simuler un réseau Ethereum sans avoir à passer par des étapes de configurations complexes. Elle permet aussi de simuler le minage des blocs induit des transactions sur la blockchain ce qui permet de tester rapidement des applications.

Un autre avantage est la simulation de comptes, il n'y a pas besoin de travailler avec des mots de passes et de créer de multiples utilisateurs manuellement. Elle s'occupe de fournir directement des comptes fictifs auxquels ont peu se raccrocher pour effectuer des tests.

Cependant pour notre application, cela ne représentait pas la réalité des choses en termes d'utilisation, scalabilité ou encore fonctionnalités que l'on voulait mettre en place. C'est pourquoi nous avons décidé de migrer nos smart contracts sur une blockchain de test (testnet). Ce qui nous permet de mieux visualiser et estimer les coûts d'une telle application.

4.4.4 Pourquoi un testnet et lequel choisir ?

Goerli Testnet est un testnet basé sur Ethereum sur la « proof-of-stake » (PoS), il est connu pour sa stabilité, sa vitesse et ses faibles frais de transactions. Cependant il faut savoir que Goerli a été déprécié le 1 janvier 2024 ce qui fait que le réseau n'est plus activement maintenu. Lors de nos recherches il semblait être le « go to » pour faire développer des applications sur un testnet mais il faudrait aujourd'hui le migrer sur le testnet Sepolia.

Pourquoi avoir choisi Goerli : Pour notre projet, Goerli est préféré pour sa popularité et sa stabilité actuelle, ce qui en faisait un choix stable pour le développement et les tests. À l'heure actuelle (janvier 2024) il est encore raisonnablement stable mais pour continuer nos tests il faudrait migrer notre projet vers Sepolia testnet.

Sepolia est aussi un testnet Ethereum basé sur la « proof-of-stake » et est aujourd'hui (2024) le testnet par défaut pour le développement de smart contracts. Il offre un réseau un peu plus prévisible. Il a l'avantage d'avoir des temps de synchronisation plus rapides et des exigences de stockage plus faibles pour exécuter son propre nœud comparé à Goerli.

Qu'est-ce qu'un nœud ? :

Un « nœud » dans le contexte des réseaux blockchain comme Ethereum est un ordinateur participant au réseau, il tient à jour une copie de la blockchain. Il joue un rôle clé en validant et relayant les transactions pour maintenir l'intégrité et le consensus du réseau. On s'en sert pour récupérer des détails de transactions, envoyer des nouvelles transactions, interagir avec des smart contracts enfin de manière générale, communiquer avec la blockchain.

Il existe 2 façons principales de travailler avec des nœuds sur la blockchain Ethereum. Premièrement, il est possible de configurer et maintenir son propre nœud mais cela implique qu'on doit gérer une infrastructure nécessaire pour rester synchronisé avec la blockchain et c'est complexe et cela peut demander beaucoup de ressources.

Deuxièmement, nous pouvons utiliser un fournisseur de services qui offre un accès à un nœud distant. C'est que nous avons exploité dans notre projet. De cette manière on peut interagir avec la blockchain sans les défis liés à la maintenance d'un nœud personnel, simplifiant ainsi le développement et les tests.

NB : Nous avons utilisé infura, plus d'infos ici : https://www.infura.io/networks/ethereum

4.4.5 Smart contracts et Programmation Solidity

Les smart contracts sont au cœur de notre projet et constituent une partie essentielle de l'écosystème Ethereum. Ils sont des programmes autonomes stockés sur la blockchain qui s'exécutent lorsque des conditions prédéfinies sont remplies. Ces contrats intelligents permettent des transactions automatisées et sécurisées sans nécessiter d'intermédiaires, offrant ainsi une fiabilité et une transparence accrues.

Solidity est un langage de programmation orienté objet créé spécifiquement par le développeur du réseau Ethereum pour créer et implémenter des smart contracts sur la blockchain.

Solidity a beaucoup de similarité avec C et C++ et est raisonnablement « simple » à comprendre et apprendre. On a par exemple le « main » en C qui est l'équivalent d'un « contract » sur Solidity. On retrouve comme sur d'autres langages, des variables, fonctions, classes, opérations arithmétiques, la manipulation de String et autres.

Avec Solidity, on crée du code au niveau machine qui est compiler sur la Machine Virtuelle Ethereum (**EVM**).

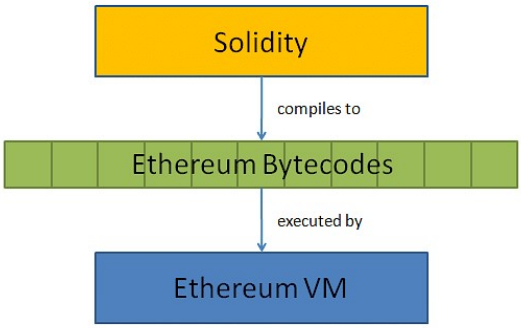

Figure 6 – Schéma de déploiement

L'EVM est un élément central de l'écosystème Ethereum. Celle-ci agit comme couche d'exécution globale qui permet le fonctionnement des smart contracts sur la blockchain. Voici quelques points clés pour comprendre son rôle :

Environnement d'exécution intelligent : Chaque nœud du réseau Ethereum exécute une instance de l'EVM, qui garantit ainsi l'exécution uniforme des smart contracts dans le réseau Ethereum.

Sécurité et Isolation : L'un des rôles majeurs de l'EVM est d'assurer la sécurité. Pour se faire elle exécute les smart contracts dans un environnement isolé, empêchant ainsi les contrats de corrompre ou d'interférer avec d''autres contrats du réseau ou même avec lui-même. Cela permet de protéger le réseau contre les attaques malveillantes ou les bugs de certains smart contracts.

Prévention des DDOS : Tout d'abord il faut vite comprendre ce que sont les gas cost Ethereum : Chaque opération faite sur la blockchain possède un coût fixe en gas déterminé à l'avance. Il s'agit simplement du coût relatif à la puissance de calcul nécessaire pour effectuer une opération.

Reprenons, pour empêcher les attaques par déni de services, Ethereum utilise se mécanisme de gas pour limiter la quantité de calculs et de stockage qu'un smart contract peut utiliser, ce qui empêche l'abus du réseau. Ex. Pour saturer le réseau de transactions, l'attaquant devrait payer un coût de plus en plus cher au fur et à mesure de l'attaque.

4.4.5.1 Etherscan et Remix

Etherscan est un outil essentiel pour observer les transactions et les smart contracts sur la blockchain Ethereum, tandis que Remix est un environnement de développement intégré (IDE) utilisé pour écrire, déployer et tester des smart contracts en Solidity.

4.4.6 Node JS

Pour notre projet, nous avons utilisé Node.js qui est une plateforme logicielle open source qui permet l'exécution de Javascript côté serveur. Ce qui distingue principalement Node.js d'autres environnements comme Apache par exemple, c'est sa capacité à gérer des applications en temps réel avec un haut niveau de trafic de données de manière efficace.

Il est conçu pour être léger, efficace grâce à son modèle d'entrée-sortie non-bloquant et asynchrone. Pas besoin d'attendre que chaque tâche soit complétée, ce qui le rend bien adapté pour gérer de multiples connexions simultanément. Ce qui est dans notre cas un atout intéressant pour notre application web interactive qui nécessite des interactions récurrentes avec la blockchain Ethereum.

Il facilite également l'accès à un grand choix de bibliothèques via le Node Package Manager (NPM). Parmi elles, nous utilisons Web3.js qui est essentielle pour intégrer les smart contracts Ethereum, assurant des interactions sécurisées avec la blockchain.

5 Conception

5.1 Maquette

Nous n'allons pas nous attarder sur la maquette de l'application web disponible en annexe car le visuel final est assez éloigné de notre idée de base. Cela s'explique notamment par le changement de type de données que l'application permet de signer. Notre première idée était de pousser un document pdf puis de le renvoyer avec une signature à l'utilisateur. Et finalement n'importe qui pouvait charger un pdf pour s'assurer qu'il n'ait pas eu de modifications depuis sa signature. Après discussion avec nos encadrants, nous avons décidé de mettre en place un simple formulaire permettant d'entrer des métadonnées qui seront signées par la suite, et de vérifier la signature via les informations des transactions récupérées sur Etherscan.

5.2 UML

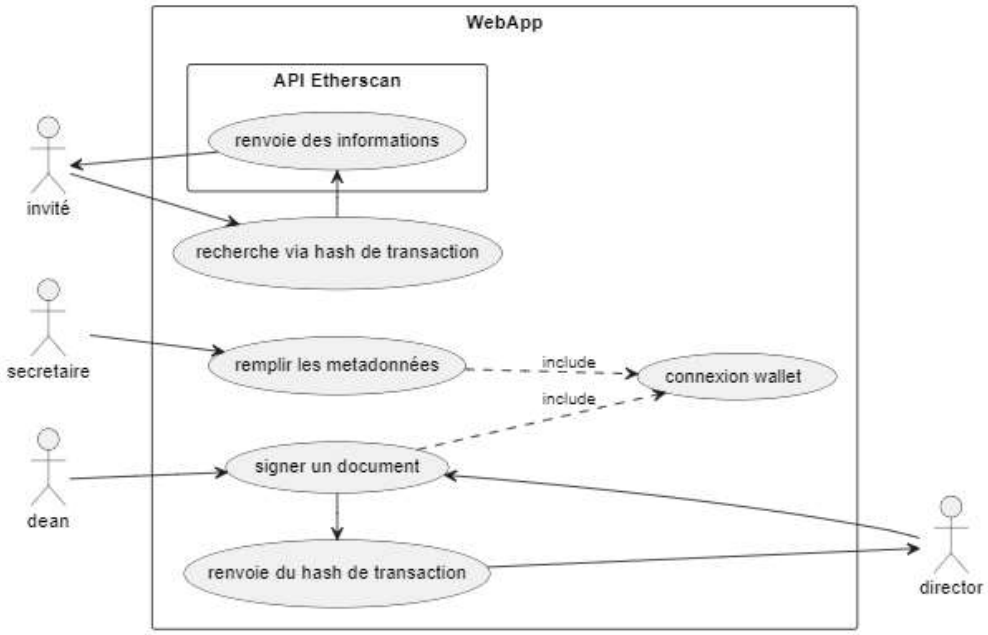

Figure 7 – Schéma UML Use Case

Comme on peut le voir sur le schéma ci-dessus, il y a quatre acteurs différents capable d'interagir avec notre système, un invité qui représente en fait n'importe qui accédant au site web et qui est autorisé à rechercher si un document est valide ou non, une secrétaire autorisée à remplir le formulaire et envoyer les données sur la blockchain, un doyen qui peut sélectionner les données qu'il veut signer et un directeur qui établit une deuxième signature et qui reçoit le hash de la transaction sur la blockchain permettant via Etherscan de rechercher et valider les informations signées.

5.3 Séquence de signature

Le diagramme de séquence de signature en annexe montre que la méthode d'authentification passe seulement par Metamask. Pour chaque action disponible dans l'application (à part la vérification) elle va demander de se connecter à un portefeuille Metamask d'abord. Ensuite l'utilisateur sera en mesure de remplir des données ou d'en signer. C'est dans les smart contracts qu'on va vérifier que le portefeuille connecté est bien celui autorisé à appeler la fonction demandée. Pour l'ordre du processus, il faut évidemment que la secrétaire rentre d'abord des données pour qu'elles apparaissent ensuite dans la liste du doyen. Les signatures se font aussi dans un ordre précis, c'est le directeur qui finit par valider totalement le processus il faut donc que les données soient d'abord signées par le doyen.

5.4 Séquence de validation

Comme nous l'avons précisé précédemment et comme le montre le diagramme en annexe, n'importe qui accédant au site à la possibilité de vérifier des données, car finalement notre système, fait seulement le lien entre l'utilisateur et l'API Etherscan en recherchant des informations via un hash de transaction et en les affichant de manière lisible.

6 Implémentation

6.1 Smart contract

Nous avons décidé lors de la conception, de découvrir le plus de fonctionnalités et d'éléments que peuvent offrir la blockchain et les smart contracts quitte à perdre un peu d'ergonomie sur l'application web en elle-même. Notre projet a pour but de découvrir cette nouvelle technologie. C'est pour cela que l'application par exemple ne possède pas de base de données, de système d'authentification direct, ... Tout cela est géré par les contrats.

Nous avons séparé le travail entre deux contrats, un premier contrat qui va seulement être appelé par le second pour gérer les accès des différentes fonctions d'ajout et de signature du deuxième. De cette manière nous avons aussi pu expérimenter l'échange d'informations entre deux contrats distincts.

Une explication brève de la mise en place de ce processus : il faut créer dans le contrat de certification une instance de celui d'autorisation en lui passant en paramètre son adresse sur la blockchain.

```
contract CertificationContract {
    // Authorisation contract's address
    address addressAuthorisationContract = 0xCBf9F13dBE1a332040E13a24394f8cc23a836c5F;
    AuthorisationContract authorisationContract;

    constructor() {
        authorisationContract = AuthorisationContract(addressAuthorisationContract);
    }
```

Ensuite on est capable d'appeler les fonctions présentes dans ce contrat :

```
modifier onlySecretary() {
    require(authorisationContract.isSecretary(msg.sender) == true, "Vous n'etes pas autorise a rentrer un nouveau document.");
    _;
}
```

La capture ci-dessus permet aussi de découvrir deux options particulièrement utiles pour la gestion des autorisations. La 1ère est le type "modifier" : c'est une fonction qui va toujours être appelée avant l'exécution d'une ou plusieurs fonctions permettant par exemple de vérifier si l'appelant a le droit de l'exécuter. Pour se faire on utilise la deuxième option qui est "require". Cela permet de rejeter la transaction en cours si le premier paramètre est faux en passant un message d'erreur.

```
function pushDoc(string memory _name, string memory _firstName) external onlySecretary
```

On peut voir ici que le "modifier" *onlySecretary* va être appelé avant chaque appel de la fonction *pushDoc*. On peut aussi remarquer le mot-clé *external* qui définit la fonction. Il en existe plusieurs, voici ceux qu'on a pu découvrir et utiliser le long de notre projet :

- external, la fonction est appelée depuis l'extérieur du contrat
- internal, la fonction est appelée seulement depuis le contrat lui-même
- view, la fonction ne modifie pas l'état du contrat et de la blockchain il n'y a pas de coûts engendrés pour appeler la fonction, surtout utilisé pour la lecture et récupération de données
- returns([type, ...]), définit quel type de données la fonction retourne

6.2 Application web

Nous l'avons déjà dit, le projet ne portait pas spécialement sur l'application web donc il n'y a pas vraiment de code spécifique à part l'utilisation de la librairie web3.js pour la communication avec Metamask et les contrats et les fonctions asynchrones pour interagir avec la blockchain afin d'attendre certaines réponses de requêtes pour continuer dans l'exécution du processus :

```
const connectContract = async () => {
    //the name : pushDoc represent in function once ran on Solidity Remix.
    //So it's used the point 3 below "Send data to the smart contract"
    window.web3 = await new Web3(window.ethereum);
    window.contract = await new window.web3.eth.Contract(ABI, Address);
}
```

Cette fonction permettant de se connecter à notre smart contract de certification est asynchrone grâce au mot-clé *async*, lors de son appel dans le code on peut décider d'attendre sa réponse pour l'attribuer dans une variable comme on le fait à la première ligne avec le mot clé *await*.

7 Résultats

Le projet maintenant terminé, il est temps de reprendre les objectifs indiqués dans notre cahier des charges et voir quels objectifs ont pu être atteints dans le temps imparti et lesquelles n'ont pas été réalisés et pourquoi.

L'idée initiale du projet était de répondre à la problématique actuelle de la vérification de l'authenticité des diplômes universitaires, souvent sujette à falsification.

7.1 Objectifs principaux

- Développer un moyen de signer numériquement des documents ou métadonnées via les Smart Contracts et la blockchain.
    - Grâce à la mise en place d'un portefeuille de cryptomonnaies MetaMask lié à la blockchain qui comporte une clé privée et une clé publique. Il est possible de signer numériquement avec son adresse privée des données grâce à l'interaction avec les Smart Contracts et la blockchain. Ce qui rend unique chaque diplôme et certifie son authenticité.
- Mettre en place un système permettant de valider l'authenticité de ces informations signées.
    - Nous avons réussi à récupérer les informations des bachelor de chaque étudiant, ayant été signés uniquement par les personnes autorisées. Toutes tierces personnes se voient bloquées si une tentative de signature est effectuée.
    - La disposition des informations reste simple sous forme de texte brut mais le contenu est signé et authentique.

7.2 Objectifs secondaires

- La création d'une plateforme web pour rassembler la certification et la validation des informations avec signature numérique.
    - Cet objectif a évolué durant le projet car il s'est vu plus complexe que sa description. Les interactions entre le navigateur et les smart contracts de la blockchain sont gérés grâce à la mise en place de MetaMask pour identifier les personnes utilisant l'application et nos smart contracts. Puis les fonctionnalités de la bibliothèque Web3Js fourni par le framework NodeJS pour gérer les interactions entre MetaMask, l'application web et les smart contracts.
- Notifier les propriétaires des informations sur la réalisation de la signature et la présence des données sur la blockchain.
    - En l'état cet objectif n'est pas rempli, cependant une alternative a été mise en place pour accéder facilement à la page de vérification grâce à un lien direct ou un QR Code obtenu par le directeur après sa signature du bachelor.
    - La notification automatique aux étudiants est un axe d'amélioration pour nos développement futures.

7.3 Conclusion résultats obtenus

Notre projet a réussi à développer un système efficace de signature numérique et de validation de documents sur la blockchain, répondant ainsi à l'enjeu important de l'authentification des diplômes. Bien que la fonctionnalité de notification automatique n'ait pas été implémentée, l'alternative mise en place reste fonctionnelle. Globalement, le projet a atteint ses objectifs principaux, il serait intéressant de poursuivre les recherches pour obtenir un produit concret et opérationnel.

8 Limitations et perspectives

Tout d'abord, nous sommes très satisfaits d'avoir pu livrer en temps voulu un produit final fonctionnel qui répond à nos objectifs. Bien qu'il soit opérationnel, il présente également certaines contraintes et limitations, mais offre de nombreuses perspectives et possibilités d'amélioration.

Listes non exhaustives des limitations de notre projet :

- **Expérience utilisateur :**
  - Notre objectif principal était d'explorer les technologies et les aspects techniques du domaine. Par conséquent, nous n'avons pas mis l'accent sur la simplicité d'utilisation du produit final ni sur la présentation de l'interface graphique.
    - De plus, nous nous interrogeons sur la capacité des utilisateurs du programme à se familiariser avec des domaines qui peuvent sembler complexes, tels que l'utilisation de Metamask et le paiement de transactions avec des crypto-monnaies.
    - Metamask est le portefeuille numérique le plus couramment utilisé dans des projets similaires. Les deux projets présentés dans l'état de l'art exigent également que leurs utilisateurs disposent d'un portefeuille numérique pour effectuer des transactions.
  - Cependant, en dehors de l'aspect graphique de l'application et de l'acquisition d'un portefeuille virtuel, notre programme demeure très simple d'utilisation. De plus, nous avons intégré la possibilité de visualiser le bachelor à partir d'un lien ou d'un QR code.
- **Gestion des utilisateurs autorisés à certifié des diplômes**
  - Limité par le temps et désireux d'explorer au maximum les possibilités des smart contracts, la gestion des utilisateurs autorisés à certifier (secrétaire, doyen et directeur) se fait directement à l'aide des adresses publiques Metamask de ces personnes, et la validation est gérée par les smart contracts.
  - On aurait pu envisager de le faire avec une base de données et un système de connexion, cependant, le jour où le projet prendrait vie, il serait nécessaire d'introduire un intermédiaire (tiers de confiance) entre les écoles et notre système. Cet intermédiaire serait chargé de délivrer les autorisations de certification et d'authentifier les personnes, que ce soit par adresse Metamask ou par des identifiants de connexion.
- **Diplômes au format métadonnées**
  - Nos encadrants nous ont rapidement orientés et conseillés à maintenir un format de données simple, afin de concentrer nos efforts sur l'aspect technique garantissant l'authenticité et la transparence de ces informations. C'est pourquoi les bachelors sont enregistrés en tant que métadonnées et peuvent être visualisés au format texte directement sur notre application.
  - Ceci pourrait, à première vue, ne pas inspirer confiance, mais c'est en réalité une fausse impression. Les données sont les mêmes ; simplement, elles ne sont pas mises en forme ou présentées dans un format « jolie » ou habituel. Notre objectif premier était d'assurer la sécurité des données fondamentales.

Listes non exhaustives des perspectives et améliorations possibles :

- **Déploiement sur la blockchain mainnet** (la chaîne de blocs réelle en production, contrairement à des environnements de test simulés tels que Goerli)
    - Bien que des simulations comme Goerli reproduisent de manière assez fidèle les coûts et les temps d'attente du réseau mainnet, il serait envisageable, même avec un budget restreint, de réaliser des tests et des analyses en conditions réelles.
    - Cela permettrait d'effectuer une analyse approfondie du marché ainsi que des coûts associés à l'utilisation de ce système. Il est important de prendre en considération la volatilité du marché et la possibilité de fluctuations des coûts dans le temps.
    - De plus, cela ouvrirait la possibilité d'effectuer une comparaison financière avec le système actuel (coûts d'impression et de papier de qualité pour les diplômes) et de développer une stratégie marketing. Cela contribuerait également à renforcer la sécurité des diplômes, les rendant ainsi non falsifiables.
- **Amélioration de l'expérience utilisateurs**
    - L'amélioration de l'expérience utilisateur n'a pas été le centre de nos efforts dans le projet actuel, mais elle représente une perspective d'amélioration à considérer pour rendre l'aspect graphique plus attrayant, dynamique et convivial dans le futur
    - Une recherche supplémentaire pourrait être entreprise pour déterminer s'il existe une solution simplifiée permettant aux utilisateurs d'éviter l'utilisation d'un portefeuille Metamask. Toutefois, étant donné l'état actuel de l'art, l'utilisation de Metamask demeure la solution principale. Il convient d'explorer la possibilité de simplifier son utilisation dans le cadre de futures améliorations
    - Une autre suggestion serait d'envisager l'envoi automatique du QR code, accompagné du lien pour la visualisation du diplôme, à la personne concernée.
- **Mise en place d'une correction des données en cas d'erreur**
    - Actuellement, il n'est pas possible de rectifier les métadonnées une fois qu'elles ont été signées et envoyées sur la blockchain. Intégrer cette fonctionnalité constituerait une amélioration appréciable.
- **Mise en forme graphique du diplôme**
    - L'aspect graphique jouant fréquemment un rôle essentiel dans la confiance que les gens accordent aux informations, il serait bénéfique, à l'avenir, de formater les métadonnées dans un style plus esthétique et attrayant, se rapprochant du format habituel des fichiers PDF contemporains. Cette initiative pourrait renforcer l'impact visuel des informations, contribuant ainsi à instaurer une confiance accrue de la part des utilisateurs.
        - Bien que l'aspect graphique ne joue aucun rôle dans l'authenticité et la sécurité des données, il exerce une influence notable sur le jugement des utilisateurs.

- **Tests scénarios approfondis**
    - En effet, notre projet bien que fonctionnel, ne pourrait pas être déployé directement sur le mainnet d'Ethereum. Il faudrait faire une batterie de tests de sécurités afin de s'assurer que de bout en bout des opérations réalisées avec l'application soient sécurisées. Par manque de temps, le projet sert de base de test et de recherche à un produit qui cible un domaine très sensible. Pour s'assurer sa robustesse nous devrions approcher des experts en applications liées à la blockchain qui nous indiqueraient si le projet a du potentiel ou s'il existe déjà une preuve que ce n'est pas possible. Une information que l'on n'aurait manqué durant nos recherches. Une campagne de test par des utilisateurs sans connaissances travaillant avec notre application nous donnerait une idée de la réalité de l'utilisation et le cas échéant des problèmes qu'il en ressort.

9 Conclusion

9.1 Objectifs et résultats

Revenons brièvement sur les objectifs que nous avons fixés au début de notre projet. Notre mission était de résoudre la problématique de la falsification des diplômes universitaires en exploitant la technologie blockchain et les Smart Contracts. Nos objectifs principaux, tels que le développement d'une signature numérique via les Smart Contracts et la validation d'authenticité, ont été globalement atteints.

Notre application web, déployée avec succès sur la blockchain de test Goerli, permet la certification des diplômes universitaires. Grâce aux Smart Contracts, le système gère efficacement les autorisations d'écriture et signature, impliquant la secrétaire, le doyen et le directeur, utilisant Metamask et les signatures numériques liées à leurs clés privées. Le déploiement sur Goerli garantit l'authenticité et la transparence des données offerts par la technologie blockchain d'Ethereum, facilitant l'entrée de nouveaux diplômes et leur double signature.

9.2 Limitations et perspectives

Notre projet présente des limitations à souligner. L'expérience utilisateur peut être complexe pour ceux qui ne sont pas familiers avec des concepts tels que les systèmes de portefeuille numérique et les transactions en crypto-monnaie. La gestion des utilisateurs autorisés à certifier des diplômes dépend actuellement des adresses publiques Metamask, suggérant la possibilité d'envisager une solution basée sur une base de données.

Le choix de conserver les diplômes sous forme de métadonnées, bien que garantissant la sécurité des données, pourrait être amélioré visuellement pour renforcer la confiance des utilisateurs. Plusieurs axes d'amélioration sont envisageables, notamment le déploiement sur la blockchain mainnet, l'amélioration de l'expérience utilisateur, la simplification de l'utilisation de Metamask, l'envoi automatique de QR codes, et la correction des données en cas d'erreur.

Des tests de scénarios approfondis sont nécessaires pour assurer la sécurité totale des opérations de l'application. L'avis d'experts en blockchain et des campagnes de tests par des utilisateurs novices seront cruciaux pour évaluer la robustesse de notre produit.

9.3 Mot final

En conclusion, bien que notre projet ait atteint ses objectifs principaux, il offre des pistes d'amélioration importantes et ouvre la voie à des développements futurs pour mieux répondre aux besoins des utilisateurs. La technologie blockchain constitue un terrain propice à l'innovation, et notre projet n'est qu'une première étape dans cette direction. Nous restons enthousiastes quant aux opportunités futures d'optimisation et d'expansion de notre solution.

Neuchâtel 22.01.2024

Bozier Luan, Vivone Tom, Villarejo Maxime

10 Table des figures

11 Bibliographie

- 20.01.2024: Entité de certifications de documents officiels, Signature Chain: https://www.signature-chain.com/

- 20.01.2024: projet de la Southeast University pour certifications de diplômes, Blockchain-based certificate authentication system with enabling correction : https://arxiv.org/pdf/2302.03877.pdf

- 20.01.2024 : introduction à la blockchain, La Blockchain - 7 points pour faire le point : https://www.digitvalue.fr/2018/07/24/la-blockchain-7-points-pour-faire-le-point/

- 20.01.2024: Ganarche website, Truffle Suite: https://trufflesuite.com/docs/ganache/#what-is-ganache

- 20.01.2024 : Node.js, Wikipedia : https://en.wikipedia.org/wiki/Node.js

- 20.01.2024: NodeJs about, NodeJs: https://nodejs.org/en/about

- 20.01.2024 : What is solidity programming , Simplilearn : https://www.simplilearn.com/tutorials/blockchain-tutorial/what-is-solidity-programming#:~:text=Solidity%20is%20an%20object%2Doriented,records%20in%20the%20blockchain%20system.

- 20.01.2024 : Qu'est-ce que le gas Ethereum ?, Stradoji stan53 (Rédacteur financier) : https://www.stradoji.com/quest-ce-que-le-gas-ethereum/#:~:text=A%20quoi%20sert%20le%20gas%20sur%20Ethereum%20%3F,DDOS)%20d'%C3%AAtre%20efficaces.

- 21.01.2024 : Goerli vs Sepolia testnet, QuickNode, Ferhat Kochan: https://www.quicknode.com/guides/ethereum-development/getting-started/goerli-vs-sepolia-a-head-to-head-comparison

- 21.01.2024 : goodbye goerli hello sepoli, Third Web, Rahul Menon & Juan Leal : https://blog.thirdweb.com/goodbye-goerli-hello-sepolia/