



Projet de fin d'étude

Distribution et sécurisation de diplômes avec la Blockchain

Jury: Abdussalam GEMAL

Encadrant: Christophe OZCAN

Etudiants: Cédric CHHUON, Williams FUCHS, Nicolas HOVETTE

SOMMAIRE

1.	Les enjeux de la vérification des diplômes.....	4
1.1.	Un enjeu social	4
1.2.	Des écoles et des entreprises victimes.....	5
1.3.	Confiance et sécurité.....	6
2.	La vérification des diplômes.....	6
2.1.	La certification des diplômes à l'ESME Sudria	6
2.2.	La vérification des diplômes en France	6
2.3.	La blockchain une certification des diplômes dans le monde.....	10
3.	Objectifs du projet.....	11
3.1.	Architecture.....	11
3.1.1.	La blockchain	12
3.1.1.1.	Ethereum	12
3.1.1.2.	Pourquoi choisir Ethereum ?	13
3.1.2.	Le contrat intelligent	13
3.1.3.	Les JNF/NFT	14
3.1.3.1.	Définition	14
3.1.3.2.	Les standards NFT.....	14
3.1.3.3.	Les normes ERC	15
3.1.3.4.	Présentation de l'ERC-721	15
3.1.4.	Fleek et IPFS.....	16
3.1.4.1.	Création d'un système de stockage Fleek	17
3.2.	L'application Chain It	18
3.2.1.	Interface utilisateur	18
3.2.2.	Fonctionnalités	18
4.	Résultat du projet.....	19
4.1.	Front End - Interface utilisateur	19
4.1.1.	Page d'accueil	19
4.1.2.	Gestion des diplômés	22
4.1.3.	Authentification.....	23
4.2.	Back End – Serveur	25
4.3.	La base de données Fleek.....	27
4.3.1.	Architecture de la base.....	27

4.3.2.	Code « <i>upload.js</i> » - Ajout d'un diplômé dans Fleek Storage.....	29
4.4.	Authentification avec Magic Auth.....	31
4.4.1.	Présentation de Magic Auth.....	31
4.4.2.	Code d'implémentation de Magic Auth	33
4.5.	Ajout des diplômés dans la blockchain	33
4.5.1.	Développement du Smart contract.....	33
4.5.2.	Interaction avec le contrat intelligent	36
4.6.	L'organisation et la gestion du projet	37
4.7.	Axes d'améliorations	37
4.8.	Nos difficultés.....	38
4.9.	Certification d'un diplôme par un recruteur	39
5.	Conclusion	39
6.	Annexe.....	40
6.1.	Bibliographie.....	40
6.1.1.	Documentation de l'étude du sujet	40
6.1.2.	Documentation sur le développement de la solution au niveau de la blockchain	40
6.1.3.	Documentation sur le développement de la base de données Fleek Storage.....	40
6.1.4.	Documentation sur le développement de l'authentification avec Magic Auth	41
6.1.1.	Documentation sur le développement du QR Code	41
6.2.	Documents annexes	41

Introduction

De nos jours, pour décrocher le job ou l'école de nos rêves, il est impératif de créer un curriculum vitae. Le curriculum vitae ou CV résume notre expérience de travail, nos informations, et tout ce qui englobe nos compétences. Toutefois les informations sur les CV ne sont pas toujours vraies.

Selon une étude de la société Vérifdiploma, parmi 115 000 vérifications de diplômes effectuées en 2020, 6% sont des faux. Les faux diplômes se multiplient grâce au développement de l'imagerie et parce qu'il est très simple de s'en procurer un. Par exemple le site <https://www.faux-diplome.org> fournit des faux diplômes très ressemblants en quelques clics.

En France, bien que l'utilisation de faux diplômes soit sévèrement punie par le Code Pénal, la traque aux faux diplômes reste très difficile. Le candidat doit donner son consentement pour que la vérification du CV et des diplômes ait lieu. Ainsi la véracité des diplômes devient un véritable enjeu sociétal.

1. Les enjeux de la vérification des diplômes

1.1. Un enjeu social

Dans le monde du recrutement, le curriculum vitae est le cœur de la guerre de la recherche d'emploi. Pour se démarquer des autres CV, les candidats « gonflent » ou surestiment leurs compétences. Et dans certains cas cela devient un mensonge grave comme la falsification de diplôme. Cette concurrence déloyale a un impact beaucoup plus fort pour les étudiants en sortie d'école qui possède que très peu d'expérience professionnelle.

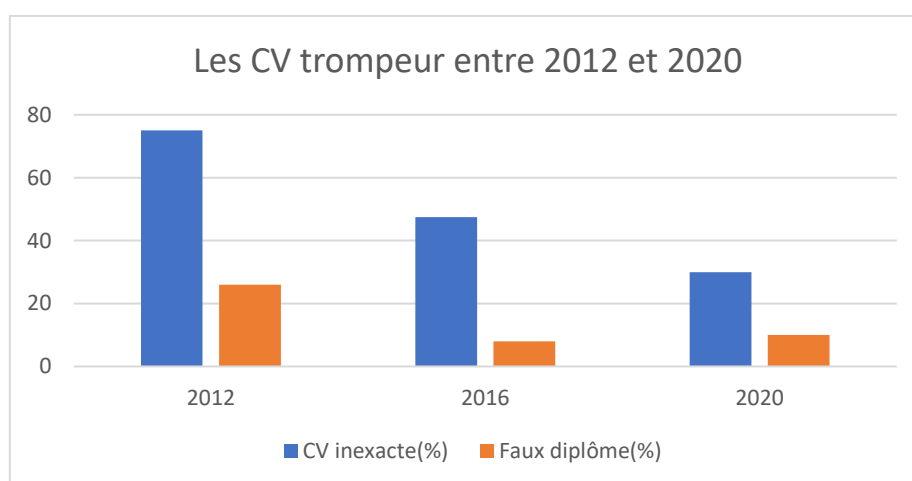


FIGURE 1.1.1 – POURCENTAGE DE CV INEXACTE ET DE FAUX DIPLOME EN FRANCE

SOURCE : CHAIN IT

Selon nos recherches, le phénomène de l'utilisation des faux diplômes s'est beaucoup développé dans les années 2000 grâce aux débuts d'internet et de la numérisation des documents. Ce phénomène qui était en régression jusqu'aux années 2012 avant d'exploser à nouveau. En 2020 selon une étude de la société VérifDiploma, 10% des CV transmis possède des diplômes qui n'ont jamais été obtenu. Le

développement des CV inexacte se montre aussi sur les chiffres. Selon un article publié en 2013 par le journal 20 minutes, seulement 54% juge que les CV sont fiables.

1.2. Des écoles et des entreprises victimes

L'usage de faux diplômes décrédibilise les établissements et porte atteinte à leur image. Selon le reportage du 13H en 2014, l'université Paris Sorbonne a recensé 80 plaintes de falsification de diplôme entre 1990 et 2014 soit environ 4 par an. Ce chiffre a fortement augmenté depuis pour atteindre 10 plaintes en 6 mois.

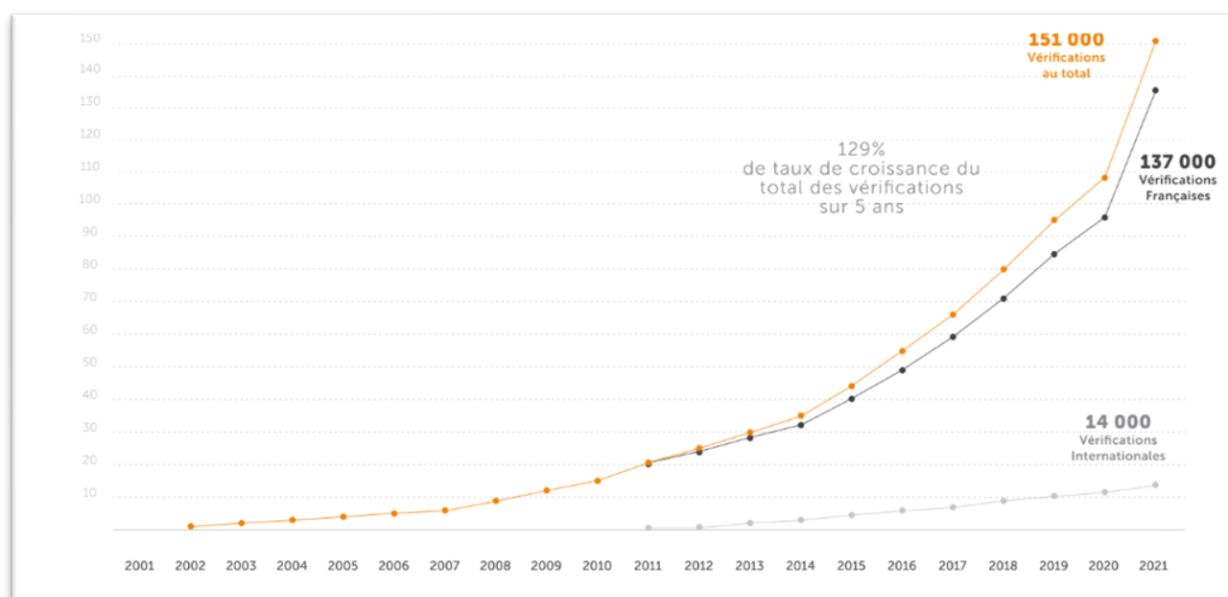


FIGURE 1.2.1 – GRAPHIQUE DU VOLUME DE VERIFICATION DES DIPLOME EN FRANCE
SOURCE : VERIFDIPLOMA

Pour les écoles et les entreprises la lutte des faux diplômes est un travail coûteux et très chronophage. Selon le responsable du suivi pédagogique Jean Maeso de l'école d'ingénieurs ESME Sudria, les appels de vérification de diplômes sont quotidiens. Les plus grandes enseignes sont même amenées à avoir un budget dédié et leur propre département de vérification de diplôme par exemple à l'université de Paris Dauphine ou à l'Université de Paris Sorbonne.

Pour les entreprises, les recruteurs ont pour devoir de vérifier la véracité des données incluses dans les CV selon le code du travail. Toutefois, pour le recruteur, la sélection des CV est une étape déjà très chronophage. Ainsi ajouter la vérification des diplômes peut s'avérer long et fastidieux. En effet, on note que seulement 33% des recruteurs vérifient les données dans les CV des candidats.

Par ailleurs, cette concurrence déloyale qui est très forte pour les jeunes diplômés engendre une perte de confiance auprès des recruteurs. Seulement un recruteur sur deux considère le CV fiable d'après un sondage du journal 20 minutes en 2013.

1.3. Confiance et sécurité

Le diplôme est une attestation de compétence qui est essentielle pour les postes à haut risque comme pour les métiers dans la santé. C'est un gage de confiance et de sécurité pour l'entourage de la personne exerçant le métier qui est parfois trompeur. En effet, les faux diplômes sont présents dans tous les domaines. Selon une étude de la société EveryCheck 30-40% des faux diplômes sont dans le domaine de la finance et des ressources humaines mais aussi 1-2% des faux diplômes sont dans le domaine de la santé. Les postes sensibles et à hautes responsabilités humaines telles que les chirurgiens sont aussi convoités et présentes d'autant plus danger sans avoir les connaissances nécessaires.

En 2020 dans le centre de santé de Montceau-les-Mines, Samantha Avril a exercé en tant que généraliste pendant plusieurs mois, avec un faux diplôme de médecin. Elle a été démasquée après avoir causé la mort de deux patients. Bien que ces actes soient punis par l'article 441-2 du Code Pénal par 5 ans de prison et 75 000 euros d'amende pour utilisation et possession de faux, cette jeune femme n'aurait jamais pu accéder à ce poste avec une simple vérification de diplôme.

2. La vérification des diplômes

2.1. La certification des diplômes à l'ESME Sudria

L'ESME Sudria reçoit des appels et des mails quotidiens pour certifier les diplômes. Et bien que les faux diplômes soient rares, l'ESME Sudria a connu un peu moins d'une dizaine de cas depuis sa création.

Pour certifier les diplômes, elle possède des bases de données référençant les procurations, les diplômes numérisés ainsi que diverses informations sur les diplômés. Ces bases de données très complètes permettent à l'établissement de certifier les diplômes en quelques minutes toutefois jusqu'à 30 minutes par jour sont consacrées à la vérification des diplômes.

Ainsi la certification des diplômes à l'ESME Sudria s'effectue en trois étapes :

- L'employeur envoie un mail à l'ESME Sudria avec le diplôme et les informations du candidat en pièce jointe.
- L'ESME vérifie dans ses bases de données si le diplôme correspond bien à un diplômé.
- L'ESME renvoie une réponse à l'employeur

2.2. La vérification des diplômes en France

Dans le marché de la vérification des diplômes, il existe de nombreuses agences de vérification dont la plus réputée est Vérifdiploma. Grâce à sa vaste base de données et ses nombreuses écoles partenaires, les recruteurs peuvent vérifier les diplômes des candidats parmi les 19 000 écoles référencées. On peut aussi citer EveryCheck qui propose des services de vérification de CV.

Les sites de vérification de diplômes ne sont pas tous payant ! En France, le gouvernement possède un service de certification des diplômes uniquement pour les attestations de baccalauréat, brevet, CAP, BEP ou BTS. L'accès au diplôme s'effectue grâce au nom et à une clé privée unique détenu par le diplômé. La société des Ingénieurs et Scientifiques de France (IESF) référence 1 141 225 ingénieurs et scientifiques qui est aussi accessible gratuitement.



Espace de vérification de diplome.gouv.fr


Vérifiez en un clic qu'une personne a bien obtenu le diplôme dont elle vous a indiqué la « clé de contrôle ».

Le périmètre des diplômes concernés évolue. [Plus d'informations](#)

Vérifier une attestation de diplôme

VÉRIFIER

FIGURE 2.2.1 – SITE DE VERIFICATION DE DIPLOME DU GOUVERNEMENT
SOURCE : DIPLOME.GOUV



IESF
SOCIÉTÉ DES INGÉNIEURS ET SCIENTIFIQUES DE FRANCE

[CONSULTATION](#) | [INSCRIPTION](#) | [CERTIFICAT D'INSCRIPTION](#) | [EURING](#) | [ESPACE ENTREPRISES](#) | [ESPACE ASSOCIATIONS](#) | [CODES](#) | [FAQ](#)

Répertoire des Ingénieurs et des Scientifiques

Il y a aujourd'hui **1 141 225** ingénieurs et scientifiques enregistrés dans le répertoire.

Pour effectuer une recherche **rapide**, saisissez des mots clés dans le champ ci-dessous (ex: "martin supelec 1984").
 Pour effectuer une recherche **avancée**, cliquez sur le bouton et remplissez les champs que vous souhaitez.

Recherche avancée

~ 5 000 résultats

Etat civil ▲	N°IESF	Etablissement	Promotion
AATZ Michel	712233	ESME	1960 (Sortie)
ABADIA Michaël	712234	ESME	2009 (Sortie)
d'ABBADIE Clément	714829	ESME	2003 (Sortie)
d'ABBADIE Jacques-Alban	1259952	ESME	2020 (Sortie)
ABBASOGLU Sara	1118088	ESME	2018 (Sortie)
ABBATI Marc	712235	ESME	1984 (Sortie)
ABBOUD Alain	712236	ESME	2000 (Sortie)
ABD ALI Mehdi	959483	ESME	2016 (Sortie)
ABDEKHALID Seddik	712237	ESME	1991 (Sortie)
ABDELLAOUI Amira	401672	ESM2 (École Centrale Marseille)	2004 (Sortie)

FIGURE 2.2.2 – SITE DE VERIFICATION DE DIPLOME DE L'IESF
SOURCE : IESF

En 2017 est fondé la société BCDiploma qui propose une solution novatrice en utilisant la blockchain Ethereum pour stocker et certifier les diplômes. Pour vérifier les diplômes, les recruteurs entre un jeton sur l'interface de BCDiploma qui va rechercher les données dans la blockchain Ethereum. Les clients ne sont plus les entreprises mais les établissements scolaires.

Interact with Contracts

Interact Deploy

Network

Ethereum

Contract

BCDiploma - EvidenZ
0x90bb...06f7

ABI / JSON Interface

```
[{"constant":false,"inputs":[{"name":"_address","type":"address"}, {"name":"_name","type":"string"}, {"name":"_legalReference","type":"string"}, {"name":"_intentDeclaration","type":"string"}, {"name":"_host","type":"string"}, {"name":"_KYB_hash","type":"string"}],"name":"addIssuer","outputs":
```

Interact with Contract

Read / Write Contract

validators READ

name string

Blockchain Certified Data

validatorAddress address

0x7332eA1229c11C627C10eB24c1A6F77BceD1D5c1

legalReference string

Blockchain Certified Data SAS - 104 avenue Albert 1er, 921

KYB_hash string

fb1fdd1090b68a8c8c161bc6022b8efb731ca7b378c47ca0c96fd9f3d4

webSite string

https://www.BCdiploma.com

logoURL string

https://gateway.ipfs.io/ipfs/QmNWkrTbW95Z7jCmyndms7QxXfR;

validatorID uint256

1

lastBlockValidity uint256

0

BACHELOR

John Doe

Date de naissance : 1 avril 1980 | Lieu de naissance : Paris

a rempli de façon satisfaisante toutes les exigences du diplôme, de 2017 à 2020.

A obtenu le diplôme académique suivant :

Logistique


par décision du jury des examinateurs le 10 juin 2020


Professeur Dr. Jestovebs


Doyen des relations internationales
Leaston Europe
11 juin 2020

Professeur Dr. Salz Hügel


Doyen des affaires académiques
Leaston Europe


LE CERTIFICAT EST VALIDE


Attribué à
John Doe


Émis par
Leaston University

Adresse
0x7332eA1229c11C627C10eB24c1A6F77BceD1D5c1
[Voir sur la blockchain](#)
[Voir sur BCDiploma.com](#)


Émetteur vérifié par
Blockchain Certified Data

Adresse
0x7332eA1229c11C627C10eB24c1A6F77BceD1D5c1
[Voir sur la blockchain](#)

À propos des preuves

Conçu par BCDiploma ©2022



FIGURE 2..2.3 – INTERFACE DE VERIFICATION DE DIPLOME DE BCDIPLOMA

SOURCE : BCDIPLOMA

NOM	SOLUTION	TARIF	RAPIDITE DE VERIFICATION	SPECIFICITE	CONTRAINTE
Diplôme.gouv	Base de données public	Gratuit	Instantanée (Recherche dans une base de données)	Diverses écoles publiques	Base de données très limitée en contenu.
IESF	Base de données public	Gratuit	Instantanée (Recherche dans une base de données)	Axés sur les écoles supérieures	Base de données très limitée en contenu.
Verifdiploma	Service de vérification de diplôme	Abonnement	Environ 48H (Plus rapide pour les écoles partenaires)	Verifdiploma possède une base de données très conséquente grâce à son grand nombre de partenaire.	Une vérification de diplôme peut durer jusqu'à 48H.
Every check	Service de vérification de diplôme	Abonnement	Environ 48H	Everycheck se focalise sur la vérification de CV.	Une vérification de diplôme peut durer jusqu'à 48H.
Bcdiploma	Solution blockchain	Abonnement	Instantanée (Recherche dans une base de données)	Utilisation des propriétés des transactions de la blockchain.	Les données insérées sont non-modifiable. Problème au niveau de la sécurité des données personnelles. (RGPD)

FIGURE 2.2.4 – TABLEAU COMPARATIF DES PRINCIPALES SOLUTIONS EXISTANTES SUR LE MARCHÉ

2.3. La blockchain une certification des diplômes dans le monde

Pour certifier les diplômes, les établissements scolaires se tournent de plus en plus vers la blockchain notamment pour sa propriété d'immuabilité des objets.

L'établissement « Massachusetts Institute of Technology » connu aussi sous le nom de MIT utilise depuis 2018 une blockchain pour certifier les diplômes. L'ajout et la vérification du diplôme par l'employeur se déroule en cinq parties :

1. *Les étudiants ajoutent leurs diplômes dans la blockchain.*
2. *Le MIT vérifie le diplôme dans la blockchain et le certifie en créant un hash du block contenant le diplôme.*
3. *Le MIT envoie le hash au diplômé.*
4. *Le diplômé envoie ses identifiants blockchain à toute personne souhaitant vérifier l'authenticité de son diplôme d'État, par exemple un employeur.*
5. *L'employeur rentre les identifiants sur le site <https://credentials.mit.edu/> du MIT et vérifie ainsi l'authenticité du diplôme.*

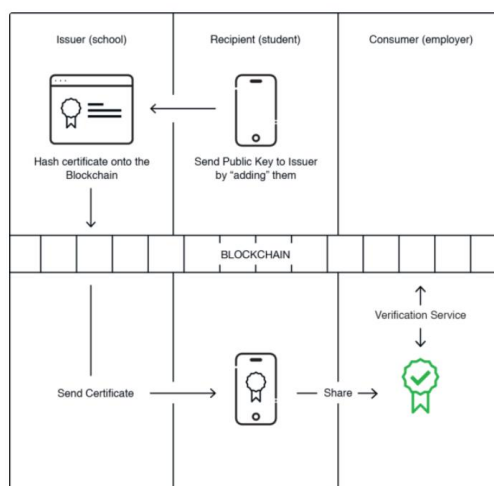


FIGURE 2.3.1 – DEROULEMENT DE LA VALIDATION DES DIPLOMES

SOURCE : BLOCKCERTS

L'utilisation de la blockchain pour certifier les diplômes se développe de plus en plus dans le monde notamment durant la période du COVID-19 où les cérémonies de remise de diplômes étaient compromises.

Par exemple, l'établissement POSTECH en Corée du sud a stocké les diplômes sur la blockchain publique « ICONLOOP ». L'établissement a par la suite envoyé un QRCode aux étudiants qui permettait d'accéder à leurs diplômes sur la plateforme de l'établissement.

3. Objectifs du projet

L'objectif de notre projet est de transformer les CV. Les CV du futur seront interactifs et constitueront un véritable dossier d'inscription. Des QR Codes ou des liens seront inclus dans les CV pour vérifier les données très rapidement. Dans un premier temps, nous voulons résoudre le problème des faux diplômes qui actuellement demande beaucoup de ressources humaines et financières pour les écoles et les entreprises. Les entreprises recrutant veulent s'assurer que le candidat est bel et bien diplômé et les établissements veulent garder leur image intacte.

Pour certifier, stocker et distribuer les diplômes de manière sécurisée, nous utiliserons la technologie blockchain. C'est l'opportunité que les entreprises et les écoles attendaient. La blockchain garantit l'intégrité de la donnée, sa sécurité et sa transparence. De plus, la blockchain est accessible à tous et permet de stocker les données dans un registre de manière immuable. Ainsi la blockchain offre un cadre idéal pour la certification des diplômes.

Pour ce faire, nous allons utiliser des contrats intelligents capables d'exécuter des instructions au sein même de la blockchain. Les instructions seront d'enregistrer toutes les informations relatives aux diplômes (propriétaire, établissement, année, niveau...), le diplôme sous forme d'un Non Fungible Token (ou NFT), de gérer les droits de propriétés du diplôme (son appartenance) ou simplement une lecture des différentes données associées au contrat intelligent.

Enfin une application permettra la simplification de l'émission et la consultation de diplômes. Elle exécutera les méthodes de notre contrat intelligent. Ainsi elle sera connectée à l'API Ethereum.

3.1. Architecture

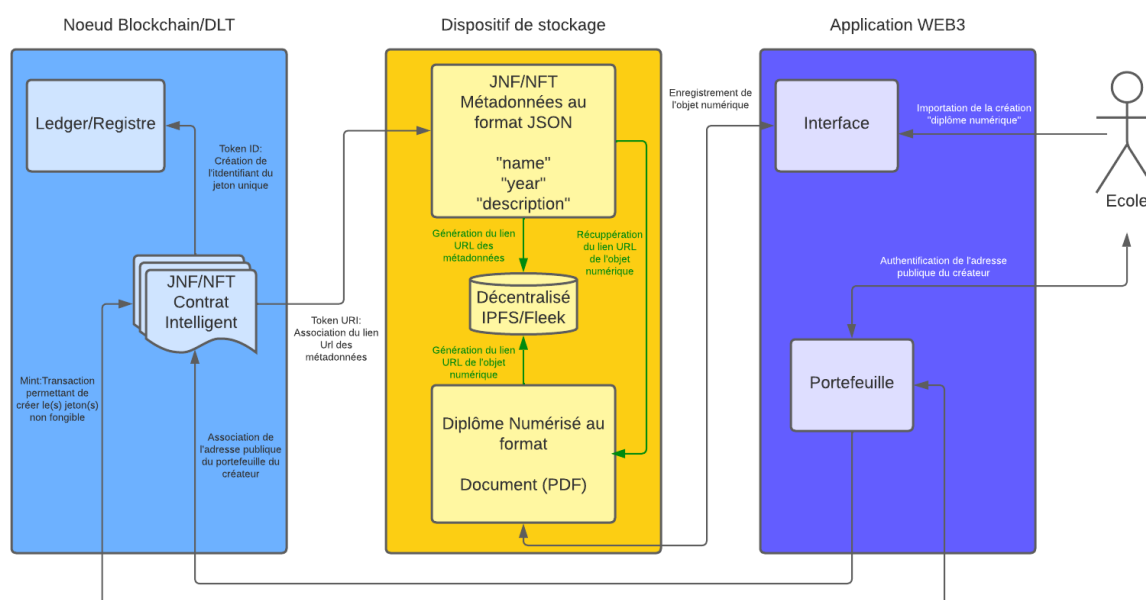


FIGURE 3.2.1 – ARCHITECTURE DU PROJET
SOURCE : CHAIN IT/CRYPTO4ALL

3.1.1. La blockchain

La blockchain est une technologie de stockage et de transmissions d'informations dans un intermédiaire centralisé. Elle garantit une haute sécurité et transparence pour les utilisateurs connectés en réseau. Elle est constituée d'une grande base de données sécurisée et distribuée où l'on y retrouve l'historique de tous les échanges entre utilisateur, depuis sa création, et elle est partagée par ses différents utilisateurs sans intermédiaire.

Registre distribué composé de blocs confirmés organisés en une chaîne séquentielle ayant des liens cryptographiques.

3.1.1.1. Ethereum

Ethereum est une technologie permettant de construire des applications et des organisations, de gérer des actifs, de traiter et de communiquer sans être contrôlé par une autorité centrale. Il n'est pas nécessaire de transmettre toutes vos données personnelles pour utiliser Ethereum - vous gardez le contrôle de vos propres données et de ce qui est partagé. Ethereum possède sa propre crypto-monnaie, l'Ether, qui est utilisée pour payer certaines activités sur le réseau Ethereum.

L'Ethereum fonctionne comme une plateforme de logiciels ouverts fonctionnant sur la base de la technologie de la Blockchain. Cette Blockchain est hébergée sur de nombreux ordinateurs à travers le monde, la rendant ainsi totalement décentralisée. Chaque ordinateur dispose d'une copie de la Blockchain et il doit obligatoirement y avoir un accord majoritaire avant que le moindre changement ne puisse être implémenté sur le réseau.

La Blockchain de l'Ethereum est similaire à celle du Bitcoin dans le sens où c'est un registre de l'historique des transactions. Cependant, le réseau de l'Ethereum permet également aux développeurs de construire et déployer des applications décentralisées (« dapps »). Elles sont également stockées sur la Blockchain au même titre que les registres de transactions.

La proof of work, ou POW, est aujourd'hui la principale façon de « miner », c'est-à-dire de valider des transactions et de créer de nouvelles unités de crypto-monnaies. Concrètement, il s'agit d'un protocole qui permet de valider l'intégrité de la blockchain, et de créer de nouveaux blocs. C'est notamment ce système qui fait que la blockchain est sécurisée.

Afin de valider ce nouveau bloc, le protocole POW demande à tous les mineurs de résoudre des opérations et des calculs mathématiques très complexes, qui requièrent des ordinateurs très puissants. Le premier ordinateur à répondre à ces calculs « gagne » le droit de miner le prochain bloc, et reçoit en retour une récompense sous forme de crypto-monnaie.

Depuis le 15 septembre 2022, Ethereum a repensé intégralement son infrastructure. En passant du PoW au PoS

La preuve d'enjeu est un type de mécanisme de consensus utilisé par les réseaux blockchain pour obtenir un consensus distribué.

Cela requiert que les utilisateurs misent leurs ETH pour devenir validateurs sur le réseau. Les validateurs sont responsables de la même chose que les mineurs dans le cadre de la preuve de travail comme ordonner les transactions et créer de nouveaux blocs afin que tous les nœuds puissent s'accorder sur l'état du réseau.

La preuve d'enjeu apporte un certain nombre d'améliorations au système de preuve de travail :

- Meilleure efficacité énergétique : vous n'avez pas besoin d'utiliser beaucoup d'énergie en minant des blocs.
- Réduction des barrières à l'entrée et des exigences matérielles : vous n'avez pas besoin de matériel haut de gamme pour avoir une chance de créer de nouveaux blocs.
- Plus grande immunité à la centralisation : la preuve d'enjeu devrait conduire à plus de nœuds sur le réseau.
- Meilleur support des chaînes fragmentées - un point clé dans le passage à l'échelle du réseau Ethereum

3.1.1.2. Pourquoi choisir Ethereum ?

Pourquoi avons-nous choisi la technologie d'Ethereum alors que de nombreuses autres technologies sont sur le marché. Dans un premier temps, Ethereum est la technologie offrant la plus haute sécurité d'un réseau décentralisé. En effet, il est extrêmement compliqué de pirater cette blockchain à cause du nombre de nœuds présents sur le réseau. De plus, elle utilise des EVM (Ethereum Virtual Machine) qui offre la possibilité de développer des Contrats intelligents, des jetons cryptographiques, ainsi que des applications décentralisées. Dans notre projet, nous allons avoir besoin de créer des jetons non fongibles grâce l'interface ERC-721 et aussi des créer une interface web qui utilisera des propriétés du Web3. Pour ajouter, Ethereum possède un écosystème très riche en outils de développement et de ressources. On peut y trouver Solidity, le langage de développement des contrats intelligents, Vyper, un autre langage de développement basé sur python, le célèbre RemixIDE, l'IDE de développement de contrats intelligents, Web3.js, une API de Javascript Ethereum, MetaMask, un portefeuille virtuel pour stocker et effectuer des transactions, et pour finir Ropsten, le Testnet network pour essayer son code de contrat intelligent avant le déploiement sur le réseau Ethereum.

Pour toute ses différentes raisons, nous avons choisi la blockchain Ethereum.

3.1.2. Le contrat intelligent

Un contrat intelligent est la transcription en langage informatique de conditions enregistrées sur un registre décentralisé et exécutées automatiquement (sur une machine virtuelle) en fonction d'un ou plusieurs évènements.

Le propriétaire du contrat intelligent sera la Blockchain et permettra la vérification du diplôme.

Le diplôme sera possédé par l'étudiant à tout moment et seulement l'école aurait les droits du diplôme.

Programme informatique intégré dans un système DLT dans lequel le résultat de toute exécution de ce programme est enregistré sur le registre distribué.

3.1.3. Les JNF/NFT

3.1.3.1. Définition

JNF est un acronyme de jeton non fongible. Il est plus souvent connu sous le nom NFT pour Non Fungible Token. Il désigne un jeton unique qui par définition ne peut être remplacé par d'autres jetons de la même nature, qualité ou quantité. Dans la blockchain, les NFT sont utilisés pour authentifier un bloc.

Non fongible veut dire ici que le jeton est unique et ne peut pas être confondu avec un autre bien. Dans la blockchain il est utilisé comme certificat d'authenticité d'un fichier numérique auquel il est rattaché.

Un NFT est une utilisation originale du contrat intelligent : un jeton stocké sur la blockchain.

Le fonctionnement de cette dernière permet d'attester la validité de toutes les transactions en son sein et donc la validité d'un jeton non fongible s'y trouvant. Il est également très simple de parcourir la blockchain pour y observer les transactions, ce qui permet à n'importe qui de vérifier l'origine et l'historique complet d'un NFT.

L'utilisation d'un NFT peut être diverse.

Associer un NFT à un élément physique, celui-ci devient lui aussi singulier et authentifiable par la même occasion.

C'est la raison pour laquelle les entreprises commencent à utiliser cette technologie afin de garantir l'authenticité de leur produit. En effet, il est tout à fait envisageable d'intégrer au sein d'un NFT des données, telles que le type/nom de l'objet, la marque, la signature du magasin, les mentions légales, le lieu de fabrication et d'autres...

De plus, par exemple, en explorant la blockchain, un client pourrait de cette façon voir que l'objet est bel et bien sorti d'une véritable usine.

De ce fait, en associant un objet et un NFT contenant toutes les informations d'authenticité, il n'est plus possible de douter de l'origine du produit. Ce procédé pourrait ainsi mettre un terme à la fabrication de contrefaçon, ou du moins en réduire sa portée.

3.1.3.2. Les standards NFT

Une norme de jeton définit le contrat intelligent et les fonctionnalités du jeton émis par celui-ci. Il existe de nombreuses normes différentes sur différentes blockchains. La catégorisation la plus simple serait entre les jetons fongibles et non fongibles.

Au fil du temps, Ethereum a gagné en popularité et maintenant la plupart des NFT sont émis sur cette blockchain. Les normes des jetons Ethereum commencent par l'abréviation « ERC » (Ethereum Request for Comments).

ERC-20 – jetons fongibles. ERC désigne un ensemble de règles qui aident les développeurs à améliorer le processus de création d'un jeton standard basé sur Ethereum, tandis que "20" est le numéro d'identification unique de la proposition.

En fait, les jetons ERC-20 sont des contrats intelligents qui offrent une grande flexibilité et fonctionnalité. Les règles de ce protocole doivent être suivies pour que le jeton interagisse avec d'autres jetons au sein du réseau. Ils peuvent agir comme des certificats, des crypto-monnaies ou des actions.

La principale différence entre ERC-20 et les autres normes de crypto-monnaie est qu'il est lié à Ethereum et ne peut être utilisé qu'au sein de ce réseau.

Les jetons ERC-20 fonctionnent comme une crypto-monnaie ordinaire, mais ce sont toujours des jetons. ERC-20 travaille spécifiquement sur la blockchain Ethereum ; ainsi, les commissions sont amorties lors des transactions et les frais de gaz dépendent directement de la charge du réseau.

ERC-20 peut être utilisé comme actif financier, actif utilitaire ou devise.

3.1.3.3. Les normes ERC

- ERC-721 – jetons non fongibles
- ERC-223 - un peu comme ERC-20 mais avec une fonctionnalité qui garantit que les jetons ne sont envoyés qu'à des adresses compatibles. Cela empêche la perte d'accès aux jetons puisqu'ils ne peuvent pas être récupérés à partir d'adresses incompatibles.
- ERC-827 - permet l'approbation des transferts de jetons fongibles afin que les jetons puissent être dépensés par un tiers sur la chaîne.
- ERC-777 - une amélioration par rapport à ERC-20. Les utilisateurs peuvent envoyer des jetons au nom de différentes adresses.
- ERC-1155 - un contrat intelligent qui permet aux utilisateurs de gérer des jetons Ethereum de nombreux types. Il peut contenir des jetons ERC-20 ou ERC-721 et fonctionne pour tous les types d'actifs : fongibles et non fongibles.
- ERC-1137 - une norme de jeton conçue pour les paiements récurrents. Cela fonctionne bien pour les abonnements nécessitant des paiements à certains intervalles.
- ERC-998 – un contrat intelligent qui permet aux utilisateurs de fusionner plusieurs NFT en un seul NFT.
- ERC-875 – un contrat intelligent qui permet aux utilisateurs de transférer plusieurs NFT en une seule transaction.
- ERC-865 - un contrat intelligent qui permet aux utilisateurs de payer une transaction avec des jetons au lieu de gaz.

3.1.3.4. Présentation de l'ERC-721

L'ERC-721 introduit une norme pour les NFT. En d'autres termes, ce type de jeton est unique et peut avoir une valeur différente de celle d'un autre jeton du même contrat intelligent, peut-être en raison de son âge, de sa rareté ou du visuel qui lui est associé.

Tous les NFT ont une variable uint256 appelée tokenId ainsi, pour tout contrat ERC-721, la paire contract address, uint256 tokenId doit être globalement unique. Cela étant dit, une dApp peut intégrer un « convertisseur » qui utilise le tokenId comme entrée et affiche une image de quelque chose de cool, comme des zombies (référence à la collection ainsi qu'au jeu).

3.1.4. Fleek et IPFS

Fleek est un outil qui utilise IPFS et nous sert pour créer un hash à nos diplômes pour notre projet. Il est un moyen simple et performant pour télécharger, épingler, et récupérer des fichiers sur IPFS et il fournit sa propre passerelle IPFS afin que tout le monde puisse accéder aux fichiers sur IPFS.

IPFS (InterPlanetary File System) est un système de fichiers décentralisé pour garantir la sécurité et la confidentialité de nos données. C'est un protocole pair à pair du web 3.0. IPFS est donc un protocole pair à pair décentralisé dont le but est de rendre le Web plus rapide, plus sûr, plus ouvert et moins cher pour le stockage. Il est conçu pour stocker sur plusieurs nœuds (serveurs) tous types de données : fichiers, sites Internet, applications ou encore des métadonnées de NFTs.

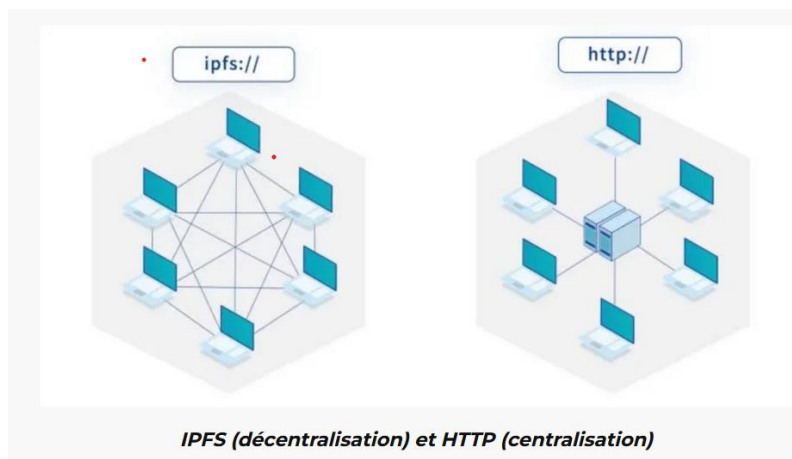


FIGURE 3.2.4.1 – SCHEMA COMPARANT LE SERVICE IPFS ET HTTP

SOURCE : CRYPTOAST

Il y a deux utilisations de Fleek qui pourrait nous intéresser : Fleek Hosting et Fleek Storage.

- Fleek Hosting nous permettrait de développer notre site web automatiquement avec une base de données des documents sous hash. Hosting utilise GitHub et IPFS pour stocker les documents. Avec Fleek, on peut déployer de manière transparente des sites statiques sur l'ordinateur Internet de DFINITY ("IC"). L'ensemble du processus de déploiement, de la création à la gestion du cycle, est abstrait et automatisé afin que nous puissions créer des sites rapides sur l'infrastructure IC sans confiance, sans autorisation et ouverte en quelques clics.
- Fleek Storage est similaire à Hosting, mais il n'utilise pas GitHub et ne crée pas de site internet. On aura alors accès qu'aux documents et leur hash associé sur l'interface IPFS. Tous les fichiers téléchargés sont publiés sur DNS et peuvent être visualisés et référencés via l'URL Fleek Storage et/ou directement sur n'importe quelle passerelle IPFS.

Nous utiliserons Fleek Storage pour sa simplicité et pour ce qu'on veut faire du projet.

3.1.4.1. Création d'un système de stockage Fleek

Dans l'espace Storage, on pourra d'abord commencer par créer un dossier, que j'ai appelé ici « Diplôme ».

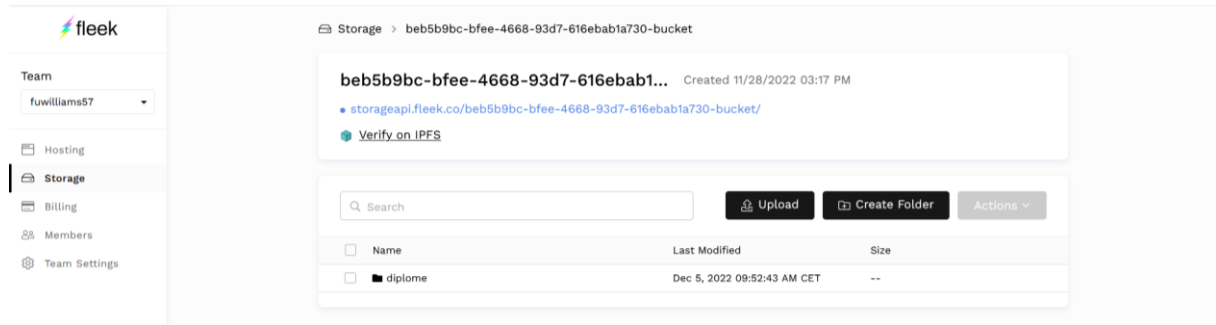


FIGURE 3.2.4.1.1 – CAPTURE D'ECRAN DE FLEEK STORAGE

SOURCE : CHAIN IT

Dans ce dossier, on y déposera les fichiers (les diplômes). Pour l'exemple, j'ai déposé un fichier texte dans mon dossier diplôme. On voit qu'un hash lui a été attribué avec IPFS et qu'il est en « pending », c'est-à-dire en attente d'être mis sur le IPFS.

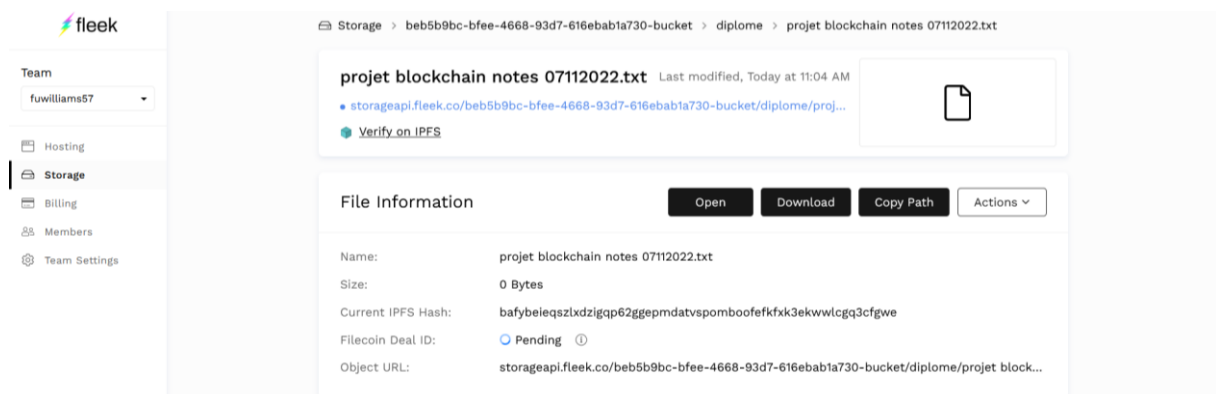


FIGURE 3.2.4.1.2 – CAPTURE D'ECRAN D'UN FICHIER SUR FLEEK STORAGE

SOURCE : CHAIN IT

Une fois le fichier importé, on peut utiliser « verify on IPFS » afin d’explorer notre document et ainsi nos fichiers, avec leur hash associé.

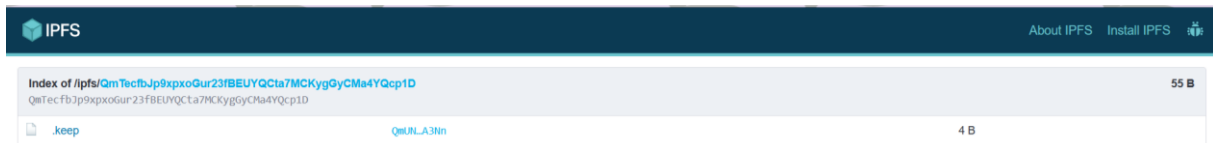


FIGURE 3.2.4.1.3 – CAPTURE D’ECRAN DE IPFS
SOURCE : CHAIN IT

Ici sur l’image, le fichier n’a pas encore été importé, cela prend entre 24 et 48 heures.

3.2. L’application Chain It

3.2.1. Interface utilisateur

L’interface utilisateur offre un contact avec les utilisateurs. Les recruteurs pourront rechercher un diplôme, les diplômés pourront consulter leurs informations et les écoles pourront y ajouter de nouveaux diplômes. Elle doit être dynamique dans son contenu, riche et attrayant, informatif et surtout, orienté vers le client avec des éléments ergonomiques et « user friendly ».

3.2.2. Fonctionnalités

Notre application aura pour fonctionnalités :

- Une solution d’authentification pour les administrateurs
- Un formulaire d’ajout d’un étudiant dans la base de données Fleek et dans la blockchain Ethereum
- Une page d’accueil avec une possibilité de rechercher un diplômé par un identifiant
- Une interface « User Friendly » pour faciliter la navigation sur le site

4. Résultat du projet

Notre application est composée d'un Front End codé en HTML, CSS, Javascript et d'un Back end en Node JS. Notre application respecte et remplit les objectifs fixés par le projet en proposant une application simplifiant l'insertion et la vérification des diplômes dans la blockchain.

4.1. Front End - Interface utilisateur

Notre interface graphique se décompose en trois pages : l'accueil, la page de gestion des diplômés et la page d'authentification.

La page d'accueil est formée des éléments suivants :

- Un bandeau de recherche
- Un formulaire de recherche
- Un formulaire de contact
- Des informations sur Chain It et sur nos services

4.1.1. Page d'accueil

La barre de recherche permet de rediriger vers les éléments de la page d'accueil ou vers la page de gestion des diplômés.

Le formulaire de recherche fonctionne avec l'identifiant des étudiants. Chaque étudiant a un identifiant unique associé à ses informations :

- Nom
- Prénom
- Filière
- Jury
- QR Code
- Diplômes

La recherche des informations passe directement par le contrat intelligent qui renvoie l'URI c'est-à-dire l'URL des données de l'étudiant.

Un QR Code qui renvoie le diplôme de l'étudiant est aussi implémenté. La génération du QR Code utilise la librairie qrcodejs provenant du site CloudFlare.

```
new QRCode(document.getElementById("qrcode"),
{
  text:"https://fleek.ipfs.io/ipfs/"+data.diplome,
  width:100,
  height:100
});
});
```

FIGURE 4.1.1.1 – CREATION DU QR CODE
SOURCE : CHAIN IT

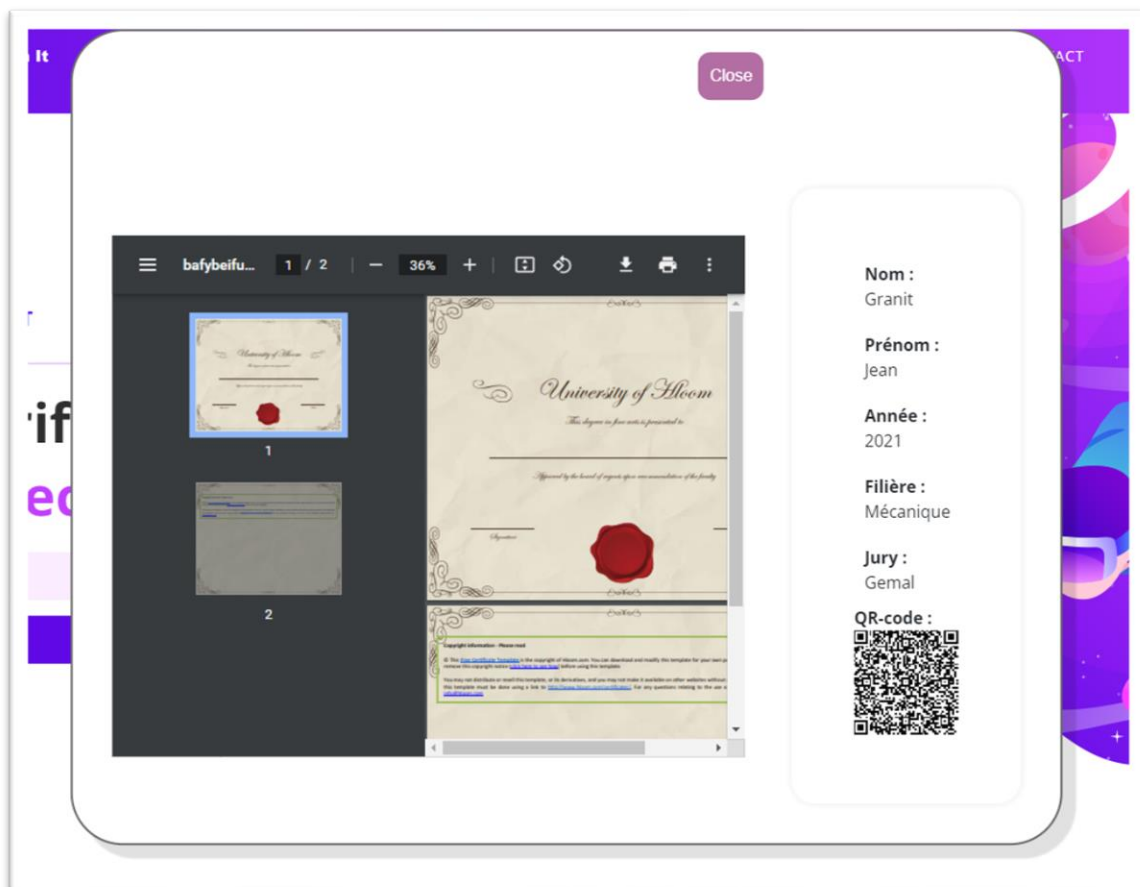


FIGURE 4.1.1.2 – RECHERCHE D'UN DIPLOME
SOURCE : CHAIN IT

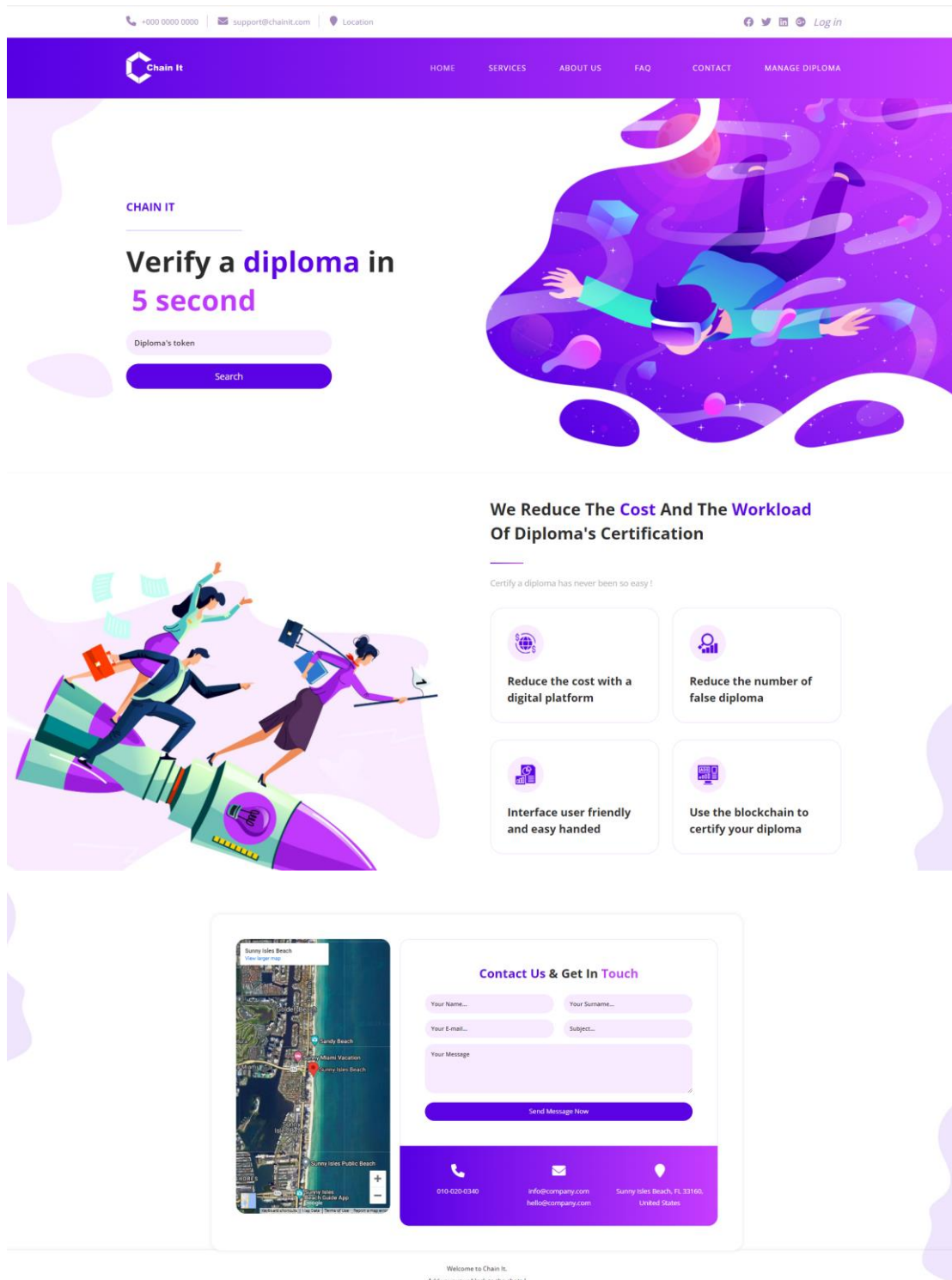
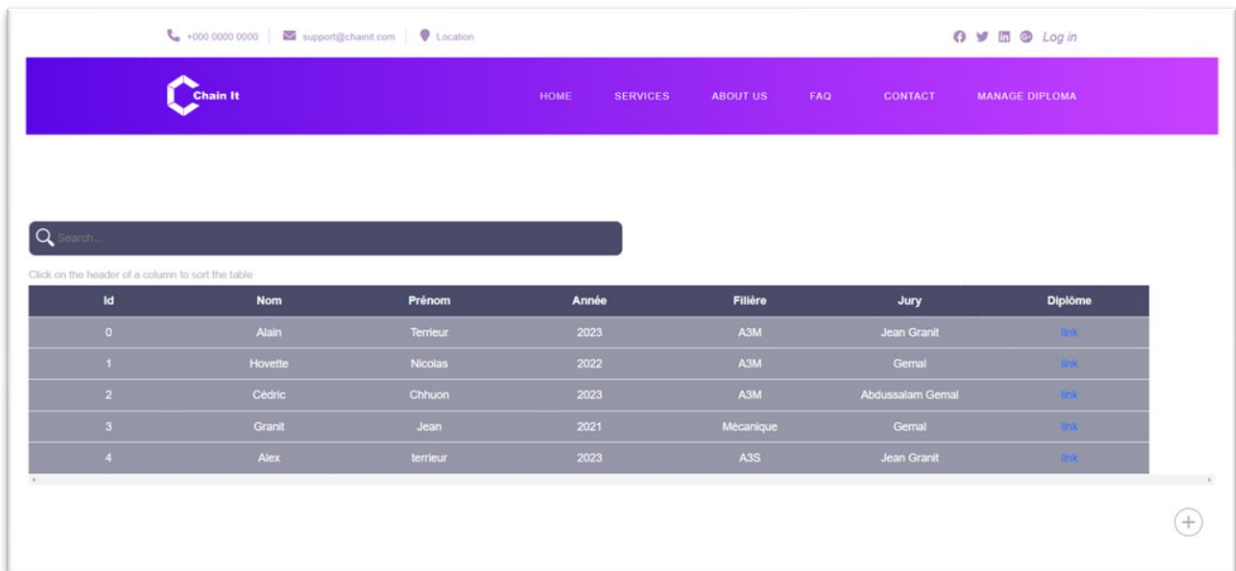


FIGURE 4.1.1.3 – PAGE D'ACCUEIL DU SITE CHAIN IT
SOURCE : CHAIN IT

4.1.2. Gestion des diplômés

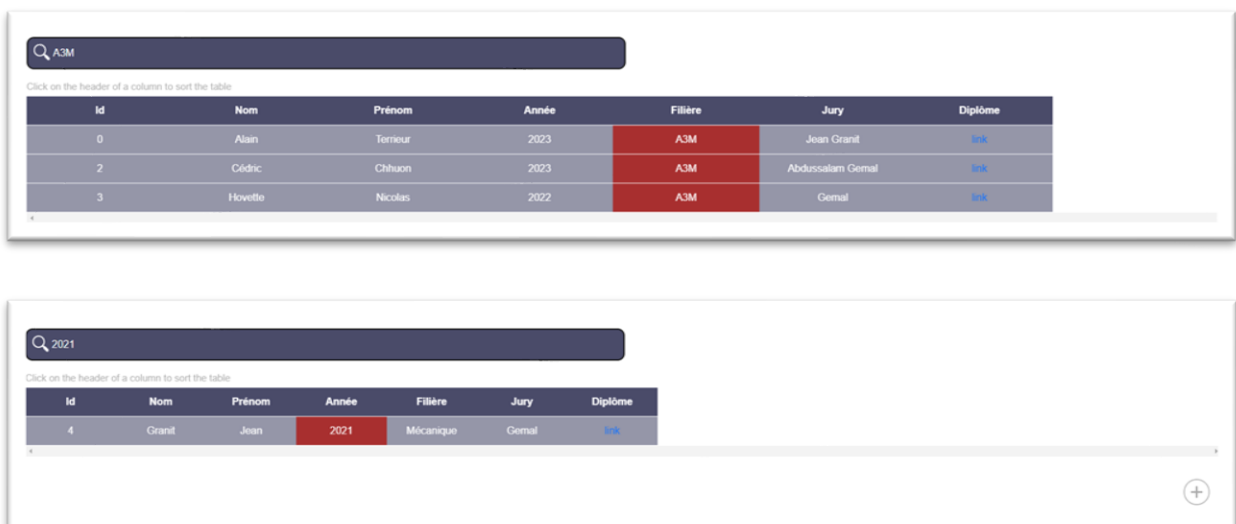
La deuxième page de notre application est la page de gestion des diplômés. Elle est constituée d'un tableau de filtrage et d'un formulaire d'ajout qui apparaît en cliquant sur le bouton « + ».



Id	Nom	Prénom	Année	Filière	Jury	Diplôme
0	Alain	Terrieur	2023	A3M	Jean Granit	link
1	Hovette	Nicolas	2022	A3M	Gemal	link
2	Cédric	Chhuon	2023	A3M	Abdussalam Gemal	link
3	Granit	Jean	2021	Mécanique	Gemal	link
4	Alex	terrieur	2023	A3S	Jean Granit	link

FIGURE 4.1.2.2 – PAGE DE MANAGEMENT

SOURCE : CHAIN IT



Id	Nom	Prénom	Année	Filière	Jury	Diplôme
0	Alain	Terrieur	2023	A3M	Jean Granit	link
2	Cédric	Chhuon	2023	A3M	Abdussalam Gemal	link
3	Hovette	Nicolas	2022	A3M	Gemal	link

Id	Nom	Prénom	Année	Filière	Jury	Diplôme
4	Granit	Jean	2021	Mécanique	Gemal	link

FIGURE 4.1.2.3 – FILTRAGE DU TABLEAU

SOURCE : CHAIN IT

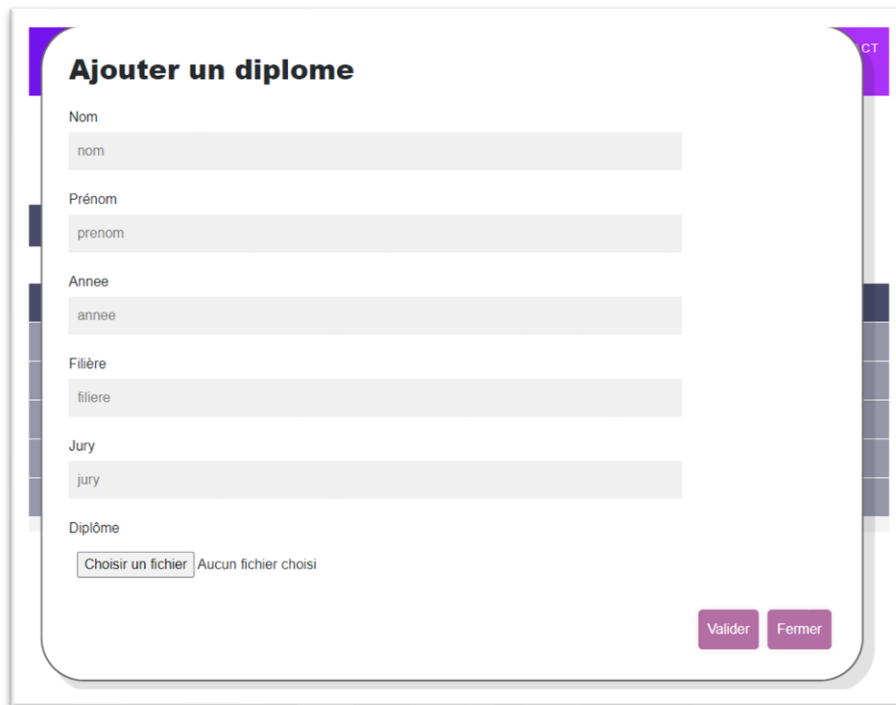


FIGURE 4.1.2.4 – FILTRAGE DU TABLEAU

SOURCE : CHAIN IT

4.1.3. Authentification

L'authentification s'effectue par la solution sans mot de passe de Magic Auth. Lors de la connexion, un e-mail est envoyé avec un code. Le code est envoyé si l'e-mail renseigné dans les utilisateurs autorisés. Une fois le code entré, l'interface de connexion nous redirige vers la page de gestion des diplômés. L'implémentation et la gestion de la solution sera détaillée dans les parties suivantes.

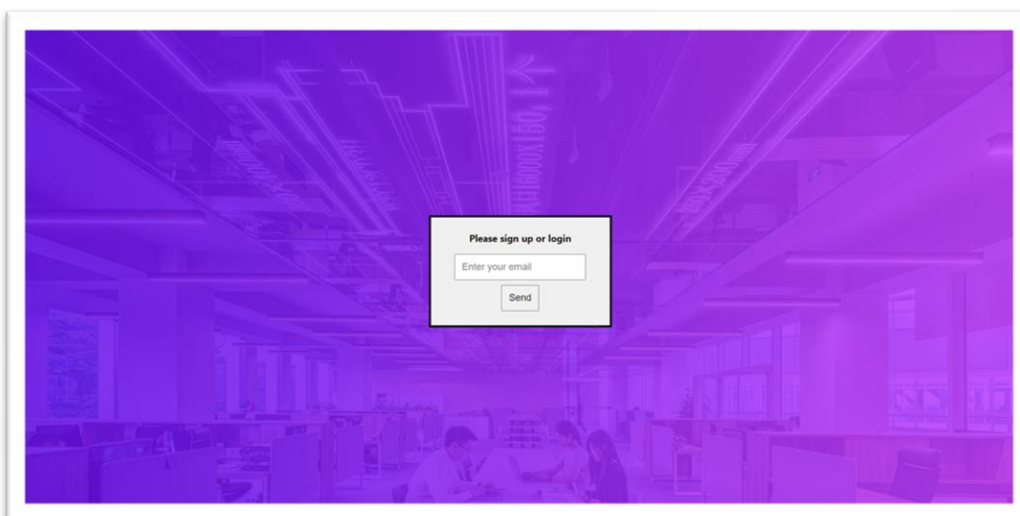


FIGURE 4.1.3.1 – FORMULAIRE DE CONNEXION

SOURCE : CHAIN IT

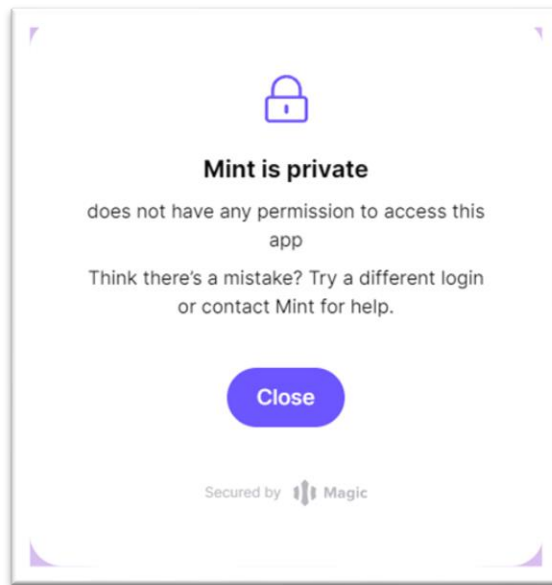


FIGURE 4.1.3.2 – CONNEXION AVEC UN MAUVAIS E-MAIL
SOURCE : CHAIN IT

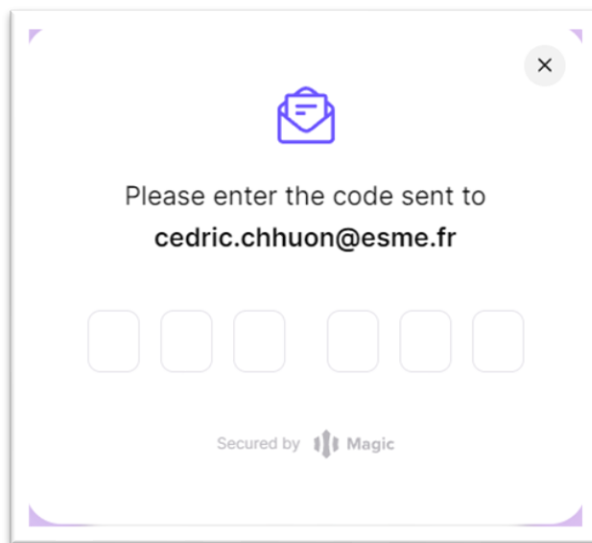


FIGURE 4.1.3.2 – CONNEXION AVEC UN E-MAIL CORRECT
SOURCE : CHAIN IT

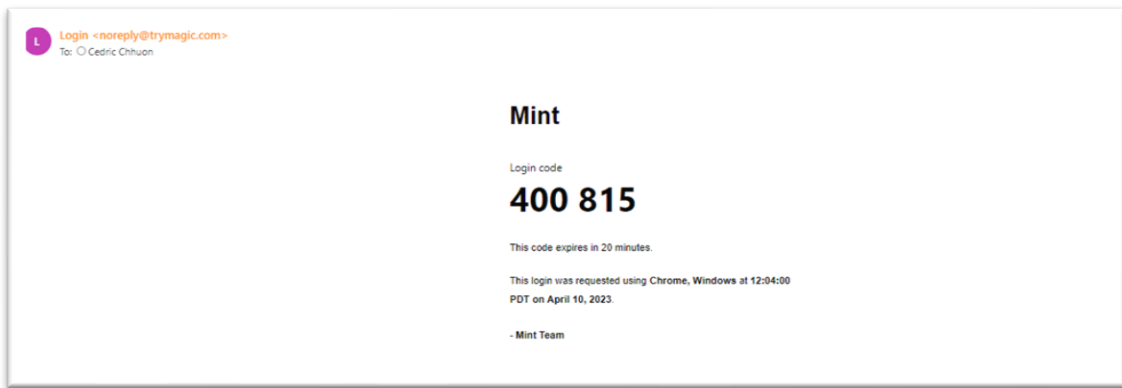


FIGURE 4.1.3.3 – E-MAIL ENVOYER PAR MAGIC AUTH
SOURCE : CHAIN IT

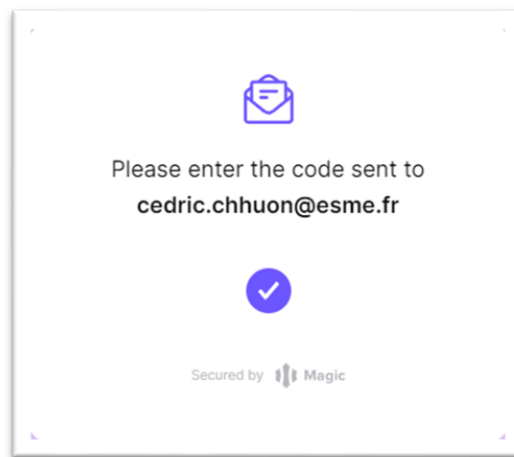


FIGURE 4.1.3.4 – CONFIRMATION DE CONNEXION
SOURCE : CHAIN IT

4.2. Back End – Serveur

Le back end est la partie d'un logiciel que les utilisateurs ne peuvent pas voir ou avec laquelle ils ne peuvent pas interagir et qui contient toutes les fonctionnalités. Le back end de notre application est codé en Node JS et contient les éléments suivants :

- Création du serveur Node JS
- Connexion aux API (Infura, Magic Auth, Fleek)
- Import des Fonctions de Fleek Storage et gestion de l'ajout des diplômés

Pour faciliter la création du serveur, nous avons utilisé la librairie express.js.

Dans un premier temps nous avons reliés le serveur aux différents dossiers du projet en créant des routes statiques :

```
app.use(express.static(path.join(__dirname, 'assets')));
app.use(express.static(path.join(__dirname, 'vendor')));
app.use(express.static(path.join(__dirname, 'eth')));
app.use(express.static(path.join(__dirname, 'back')));
app.use(express.static(path.join(__dirname, 'fleek')));
app.use(express.static(path.join(__dirname, 'authentication')));
app.use(express.static(path.join(__dirname, 'page')));
```

FIGURE 4.2.1.1 – ROUTES STATIQUES

SOURCE : CHAIN IT

Nous avons par la suite rajouté deux requêtes Web :

La première requête GET permet d'effectuer une redirection vers le fichier index.html

```
app.get('/', function (req, res) {
  res.sendFile(__dirname + '/page/index.html');
});
```

FIGURE 4.2.1.2 – REDIRECTION DE PAGE

SOURCE : CHAIN IT

La deuxième requête permet de lire le diplôme envoyé par la requête POST en utilisant la librairie fs. La fonction « FunctionUpload » est la fonction qui permet d'upload les fichiers dans Fleek Storage et dans la blockchain, elle sera expliquée dans la partie Fleek. Enfin la dernière ligne permet d'effectuer une redirection.

```
app.post('/manage_diploma.html', uploadMiddleware,(req,res) => {

  dataPDF = fs.readFileSync(req.files.diplome_create.path);

  FunctionUpload(req.body.nom_create,req.body.prenom_create,req.body.annee_create,req.body.filiere_create,req.body.jury_create,req.files.diplome_create.path);

  res.sendFile(__dirname + '/page/manage_diploma.html');
});
```

FIGURE 4.2.1.3 – AJOUT D'UN DIPLOME

SOURCE : CHAIN IT

4.3. La base de données Fleek

4.3.1. Architecture de la base

Nous avons décidé d'une architecture pour répertorier les fichiers des diplômes (json et pdf) de façon à retrouver facilement les différents diplômes. Ils seront répertoriés comme ci-dessous :

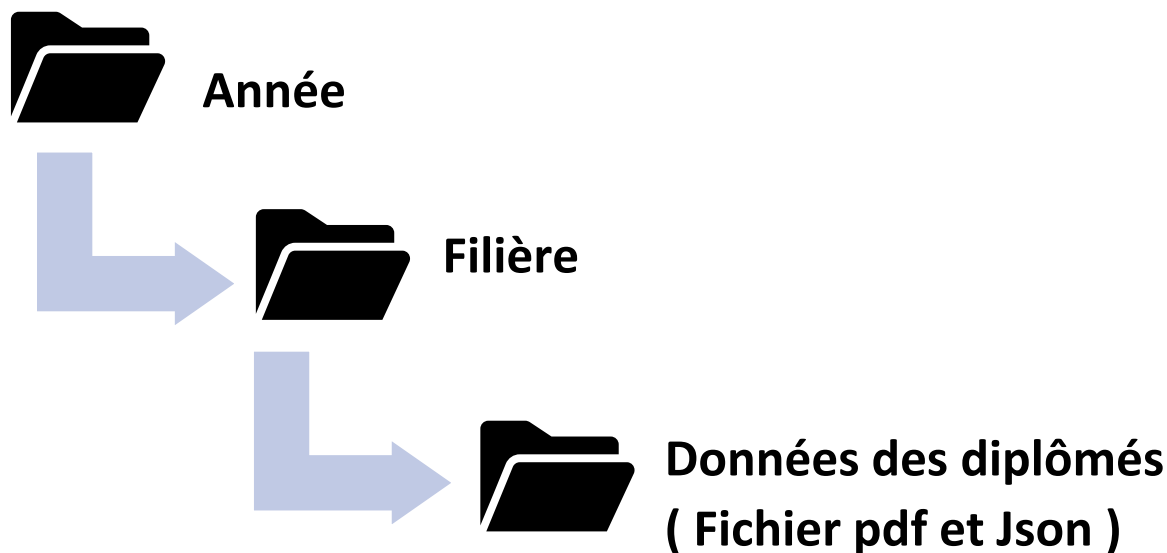


FIGURE 4.3.1.1 — ARCHITECTURE DE LA BASE DE DONNEES
SOURCE : CHAIN IT

Cette disposition est assez simpliste mais permet un repérage aisé sans se perdre.

On se retrouve alors dans fleek avec cette interface :

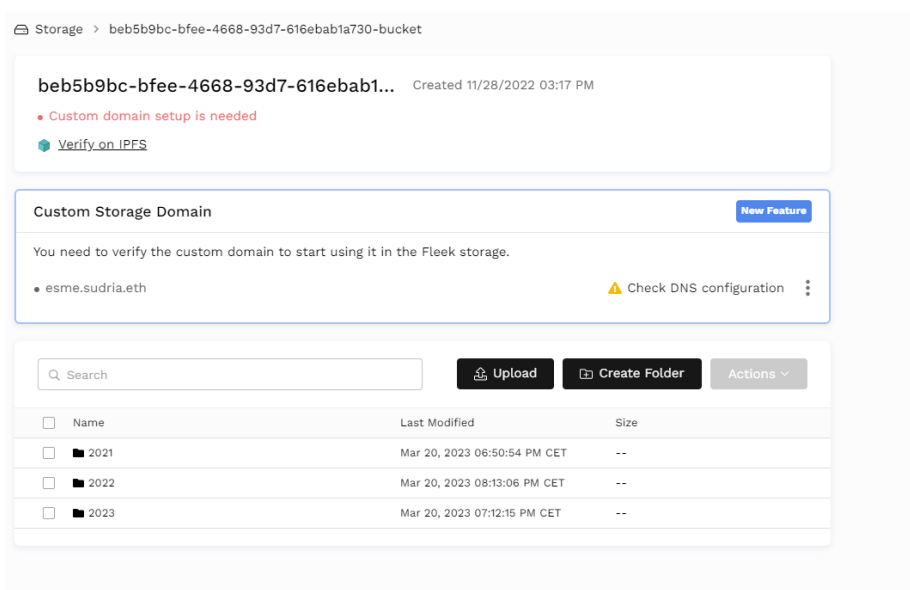


FIGURE 4.3.1.2 — PAGE FLEEK DE LA BASE DE DONNEES
SOURCE : CHAIN IT

On y retrouve les différents fichiers avec les différentes années qu'on a pu créer au préalable. Si on ouvre un de ces documents, on se retrouve alors ici :

Search

Upload

Create Folder

Actions

<div><div></div></div>	Name	Last Modified	Size
<div><div></div></div>	<div><div></div>A3M</div>	Mar 19, 2023 05:59:56 PM CET	--
<div><div></div></div>	<div><div></div>A3S</div>	Mar 20, 2023 07:14:48 PM CET	--

FIGURE 4.3.1.3 – PAGE FLEEK DE LA BASE DE DONNEES

SOURCE : CHAIN IT

On peut voir ici les filières qu'on a également créé au préalable, si filière n'existant pas dans fleek lors de l'insertion du diplôme dans l'application web, cette filière sera créée automatiquement. En ouvrant une filière, on tombe alors sur les diplômes des étudiants (les fichiers json et pdf).

Q

Search

📁

Upload

📁

Create Folder

Actions

▼

<input type="checkbox"/>	Name	Last Modified	Size
<input type="checkbox"/>	📄 Alain_Terrieur.json	Mar 20, 2023 07:12:23 PM CET	204.00 Bytes
<input type="checkbox"/>	📄 Alain_Terrieur.pdf	Mar 20, 2023 07:12:15 PM CET	20.98 KB
<input type="checkbox"/>	📄 Alex_Terrieur.json	Mar 20, 2023 06:27:15 PM CET	204.00 Bytes
<input type="checkbox"/>	📄 Alex_Terrieur.pdf	Mar 20, 2023 06:27:11 PM CET	20.98 KB
<input type="checkbox"/>	📄 Alex_Terrieur.json	Mar 20, 2023 06:57:30 PM CET	203.00 Bytes
<input type="checkbox"/>	📄 Alex_Terrieur.pdf	Mar 20, 2023 06:57:24 PM CET	20.98 KB
<input type="checkbox"/>	📄 Cédric_Chhuon.json	Mar 20, 2023 07:49:38 PM CET	209.00 Bytes
<input type="checkbox"/>	📄 Cédric_Chhuon.pdf	Mar 20, 2023 07:49:32 PM CET	37.38 KB
<input type="checkbox"/>	📄 Romain_Pierre.json	Mar 19, 2023 06:07:49 PM CET	204.00 Bytes
<input type="checkbox"/>	📄 Romain_Pierre.pdf	Mar 19, 2023 06:07:41 PM CET	253.64 KB
<input type="checkbox"/>	📄 Romgle_Faris.json	Mar 19, 2023 06:00:45 PM CET	203.00 Bytes
<input type="checkbox"/>	📄 Romgle_Faris.pdf	Mar 19, 2023 06:00:38 PM CET	253.64 KB
<input type="checkbox"/>	📄 Terrieur_Alain.json	Mar 19, 2023 06:00:01 PM CET	205.00 Bytes
<input type="checkbox"/>	📄 Terrieur_Alain.pdf	Mar 19, 2023 05:59:56 PM CET	253.64 KB

FIGURE 4.3.1.4 – PAGE FLEEK DE LA BASE DE DONNEES

SOURCE : CHAIN IT

4.3.2. Code « *upload.js* » - Ajout d'un diplômé dans Fleek Storage

Nous avons élaboré un script nommé « *upload.js* » qui nous a permis de télécharger les fichiers json et pdf du diplôme afin de les mettre dans la base de données Fleek avant d'ajouter l'URI dans la blockchain. Ce script se compose d'une fonction « *FunctionUpload* » prenant en paramètre le nom, le prenom, l'année, la filière, le jury ainsi que le diplôme afin d'insérer le tout dans le bon dossier dans fleek. On utilise la bibliothèque fleek pour utiliser les méthodes fleek, la bibliothèque fs pour lire les différents fichiers, et web3 permet d'accéder à la blockchain ethereum.

On y entre également notre clé d'API et clé d'API secrète de fleek afin d'accéder à sa base de données.

```
const fs = require('fs');
const fleek = require('@fleekhq/fleek-storage-js');
const Web3 = require('web3');

const apiKey = 'tEgyrr1CPoVt4culi4Q1xw==';
const apiSecret = 'xbcWexeLICXxIns6zMXic9PNNbEoBXBi13L1C14zvLI=';

async function FunctionUpload(nom,prenom,annee,filiere,jury,diplome){
```

FIGURE 4.3.2.1 – INCLUSION DES MODULES DE UPLOAD.JS

SOURCE : CHAIN IT

Nous nous sommes aidés de la bibliothèque fleek qui fournit une méthode « *fleek.upload(inputPDF)* » prenant en paramètre la clé et la clé secrète qui nous permet de renvoyer le hash d'un fichier, sa publicURL, sa clé et son bucket.

```
fs.readFile(filePath, async (error, fileData) => {
  const uploadedFile = await fleekStorage.upload({
    apiKey: 'my-key',
    apiSecret: 'my-secret',
    key: 'my-file-key',
    data: fileData,
  });
})
```

FIGURE 4.3.2.1 – EXEMPLE UTILISATION DE FLEEK.UPLOAD

SOURCE : FLEEK

Ce code va ainsi nous permettre de récupérer le fichier pdf du diplôme, de récupérer son URI afin de la rajouter au fichier json, et d'ensuite tout télécharger dans la base de données. Cette fonction va également créer un dossier si la filière, ou l'année n'existe pas.

```
14 dataPDF = fs.readFileSync(diplome);
15
16
17 const inputPDF = {
18   apiKey,
19   apiSecret,
20   key: "/" + annee + "/" + filiere + "/" + nom + "_" + prenom + ".pdf",
21   data : dataPDF,
22 };
23
24 const resultPDF = await fleek.upload(inputPDF);
25
26 dataJSON = `{
27   "nom" : "${nom}",
28   "prenom" : "${prenom}",
29   "annee" : "${annee}",
30   "filiere" : "${filiere}",
31   "jury" : "${jury}",
32   "diplome" : "${resultPDF["hash"]}"
33 }`;
34
35
36 const inputJSON = {
37   apiKey,
38   apiSecret,
39   key: "/" + annee + "/" + filiere + "/" + nom + "_" + prenom + ".json",
40   data : dataJSON,
41 };
42
43 const resultJSON = await fleek.upload(inputJSON);
44
```

FIGURE 4.3.2.3 – FONCTIONNALITE DE LA FONCTION DE UPLOAD.JS

SOURCE : CHAIN IT

La constante *inputPDF* correspond aux clés de l'API et indique le chemin où le fichier doivent être téléchargés dans fleek. On y indique également le type du fichier.

La constante *inputJSON* fonctionne avec le même principe. On remarquera donc l'utilisation de `fleek.upload`.

Lors de l'ajout des fichiers, cela peut prendre quelques secondes.

Le script permet également la connexion avec ethereum.

4.4. Authentification avec Magic Auth

4.4.1. Présentation de Magic Auth

Magic Auth est une extension de Magic Link qui permet aux utilisateurs d'accéder à des services en ligne ou à des applications sans avoir à saisir un nom d'utilisateur et un mot de passe chaque fois qu'ils se connectent.

L'utilisation de Magic Auth rend l'authentification plus sécurisée et plus pratique pour les utilisateurs, car elle élimine le besoin de saisir des informations de connexion.

Sur l'interface de notre projet, on peut observer l'historique des connexions à droite, les clés publiques et privées en bas.

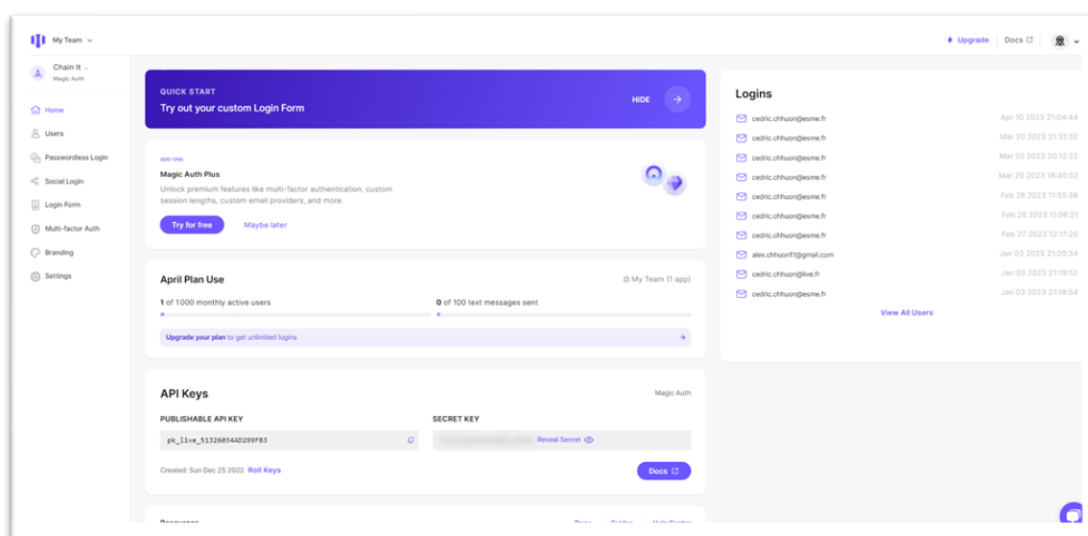


FIGURE 4.4.1.1 – PAGE D'ACCUEIL MAGIC AUTH

SOURCE : CHAIN IT

Magic Auth, possède une panoplie de configuration qui permet la personnalisation du type de connexion et la gestion des authentifications. La gestion des utilisateurs s'effectue par une liste d'accès et une liste de bannissement qui permettent d'autoriser ou de bannir un utilisateur.

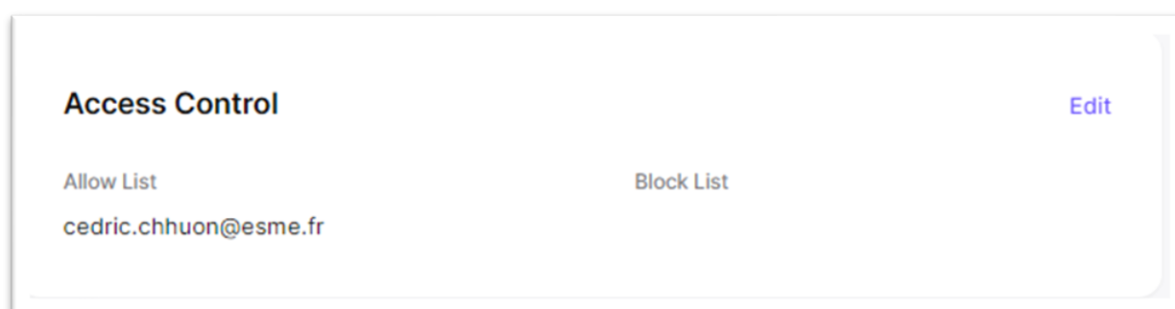


FIGURE 4.4.1.2 – GESTION DES UTILISATEURS

SOURCE : CHAIN IT

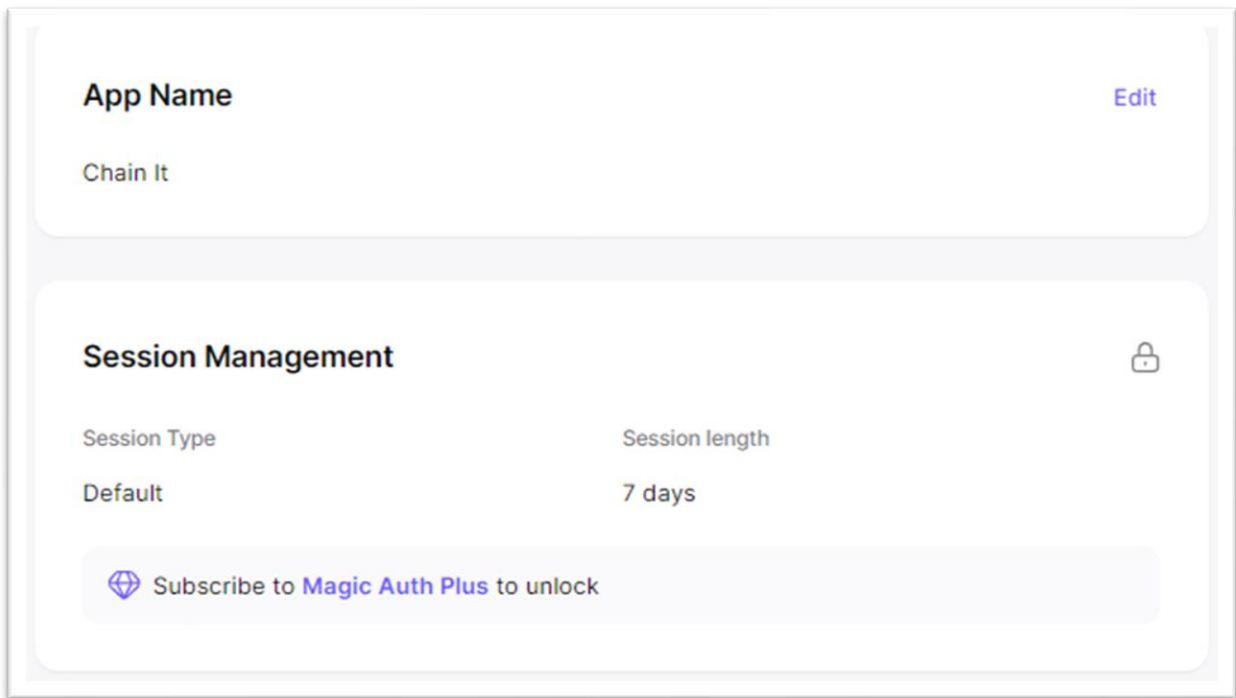


FIGURE 4.4.1.3 – GESTION DES SESSIONS

SOURCE : CHAIN IT

Magic Auth renforce d'autant plus la sécurité des comptes en instaurant différents moyens d'authentification sans mot de passe et une solution d'authentification à plusieurs facteurs (MFA).

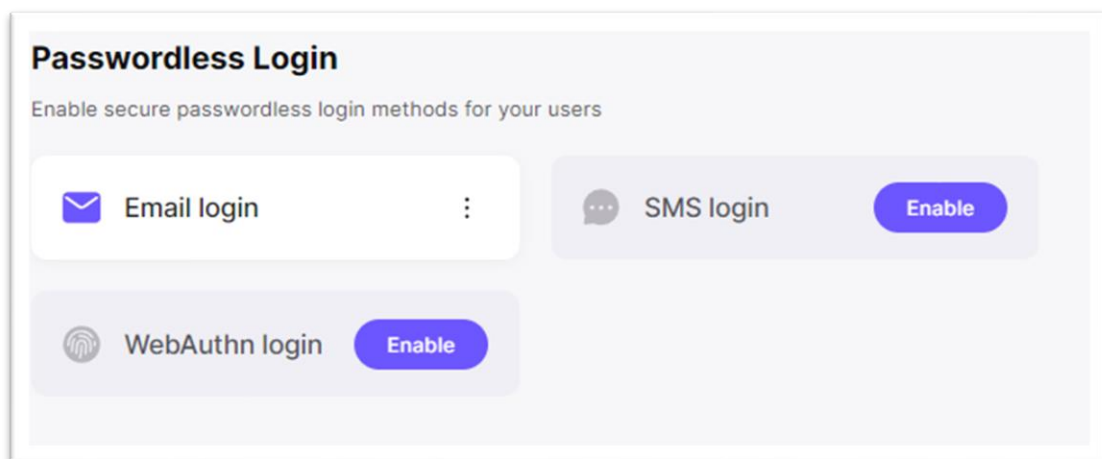


FIGURE 4.4.1.4 – AUTHENTIFICATION AVEC MAGIC AUTH

SOURCE : CHAIN IT

4.4.2. Code d'implémentation de Magic Auth

```
<!-- 1 Install Magic SDK -->
<script src="https://auth.magic.link/sdk"></script>
<script>
  /* 2 Initialize Magic Instance */
  let magic = new Magic('pk_live_51326034AD299FB3');

  /* 3 Implement Render Function */
  const render = async () => {
    let html = '';
    ...}
  /* 4 Implement Login Handler */
  const handleLogin = async (e) => {
    e.preventDefault();
    const email = new FormData(e.target).get('email');
    if (email) {
      /* One-liner login with email OTP 🧙 */
      await magic.auth.loginWithEmailOTP({ email });
      render();
    }
  };
  /* 5 Implement Logout Handler */
  const handleLogout = async () => {
    await magic.user.logout();
    render();
  };
};
```

FIGURE 4.4.2.1 – CODE SOURCE MAGIC AUTH
SOURCE : CHAIN IT

Au niveau du code de développement, la solution Magic Auth se déroule en 5 parties :

1. Installation et connexion à l'API Magic Auth
2. Connexion à notre projet en entrant la clé publique
3. Création d'un formulaire d'authentification et attente de la validation de l'e-mail
4. Gestion des erreurs
5. Gestion des déconnexions

4.5. Ajout des diplômés dans la blockchain

4.5.1. Développement du Smart contract

Le contrat intelligent ou smart contract nous permet d'ancrer toutes les informations des diplômés via un NFT. Ce NFT est créé grâce à l'interface ERC721. Cette interface, qui est également un autre smart contract, doit être importée afin d'être utilisée (Ci-dessous).

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "@openzeppelin/contracts/utils/Counters.sol";
import "./Whitelist.sol";
```

FIGURE 4.5.1.1 – IMPORTATION DES CONTRATS INTELLIGENTS

SOURCE : CHAIN IT

Pour notre contrat, nous avons 4 interfaces différentes dont 2 provenant de openzeppelin.

- ERC721.sol permettant de créer les NFTs
- ERC721URIStorage.sol permettant d'appeler des fonctions plus spécifiques (pour la modification)
- Counters.sol permettant d'incrémenter le nombre de NFTs
- Whitelist.sol permettant de gérer l'accès des wallets

Un smart contract se doit d'être sécurisé afin que tout le monde ne puisse pas interagir avec. Ci-dessous, le code du smart contrat Whitelist.sol.

Nom du contrat

Tableau intelligent stockant des adresses

Fonction nous disant si l'adresse est bien valide ou non

Fonction permettant d'ajouter un utilisateur pouvant être appelé uniquement par le propriétaire du SC

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/access/Ownable.sol";

contract Whitelist {

    address owner;

    mapping(address => bool) whitelistedAddresses;

    constructor() {
        owner = msg.sender;
    }

    modifier onlyOwner() {
        require(msg.sender == owner, "Ownable: caller is not the owner");
        _;
    }

    function isWhitelisted(address _address) public view returns(bool) {
        require(whitelistedAddresses[_address], "Whitelist: You need to be whitelisted");
        return true;
    }

    function addUser(address _addressToWhitelist) public onlyOwner {
        whitelistedAddresses[_addressToWhitelist] = true;
    }

    function verifyUser(address _whitelistedAddress) public view returns(bool) {
        bool userIsWhitelisted = whitelistedAddresses[_whitelistedAddress];
        return userIsWhitelisted;
    }
}
```

FIGURE 4.5.1.2 – EXPLICATION DU CODE SOURCE DU CONTRAT INTELLIGENT

SOURCE : CHAIN IT

Et enfin nous avons le smart contract permettant de faire le backend de notre application.

```
contract ESMEDIPLOME is ERC721, ERC721URIStorage, Whitelist {
    using Counters for Counters.Counter;
    Counters.Counter public _tokenIdCounter;
    constructor() ERC721("EsmeDiplome", "ESME") {}
    struct URI {
        string url;
    }
    URI[] private Uri_list;

    event URIAdded(string url);

    function create(string memory ipfsURLs) public {
        require(isWhitelisted(msg.sender)); // require the msg.sender to be whitelisted
        uint256 tokenId = _tokenIdCounter.current();
        _tokenIdCounter.increment();
        _mint(msg.sender, tokenId);
        _setTokenURI(tokenId, ipfsURLs);
        Uri_list.push(URI(ipfsURLs));
        emit URIAdded(ipfsURLs);
    }

    function tokenURI(uint256 tokenId)
        public
        view
        override(ERC721, ERC721URIStorage)
        returns (string memory)
    {
        require(isWhitelisted(msg.sender));
        return super.tokenURI(tokenId);
    }

    function getAll() public view returns(URI[] memory){
        require(isWhitelisted(msg.sender));
        return Uri_list;
    }
}
```

FIGURE 4.5.1.3 – CODE SOURCE DU CONTRAT INTELLIGENT
SOURCE : CHAIN IT

Tout d'abord, notre contrat intelligent s'appelle ESMEDIPLOME et hérite des fonctions de ERC721, ERC721URIStorage ainsi que de Whitelist. Il est donc capable d'appeler toutes ses fonctions. On initialise le tokenIdCounter qui sera notre compteur de NFTs.

Le constructeur donne le nom de la collection (EsmeDiplome) ainsi que son logo (ESME).

La structure suivante stockera les métadonnées des NFTs.

L'évènement permettra de remplir la structure.

La fonction Create prend en paramètres les metadonnées de notre NFTs. Dans notre cas, il s'agit du lien URI vers la base IPFS. La première ligne de la fonction est très importante car elle permet de filtrer les personnes appelant la fonction. Il faut que la personne soit enregistrée afin de pouvoir continuer. On incrémente de 1 notre nombre de nft. La fonction **_mint** donne un token et un propriétaire au NFT. La fonction **setTokenURI** ajoute les métadonnées au NFT. Et pour finir on stocke dans la stucture le lien URI.

La fonction tokenURI prend en paramètre un tokenId (par exemple : 1) et nous renvoie les informations sur le NFT avec ses métadonnées. On verifie toujours si le portefeuille appelant la fonction est autorisé.

Pour finir, la fonction getAll utilise la structure afin de renvoyer toutes les informations des NFTs. On vérifie également si le wallet est bien autorisé. Cette fonction est un besoin du front-end, elle n'était pas nécessaire pour le smart Contract.

4.5.2. Interaction avec le contrat intelligent

Pour interagir avec le contrat intelligent et donc la blockchain, il faut que notre application possède un nœud RPC dans la blockchain Ethereum. Pour simplifier la création du nœud, nous avons utilisé les infrastructures Web3 proposées par Infura.

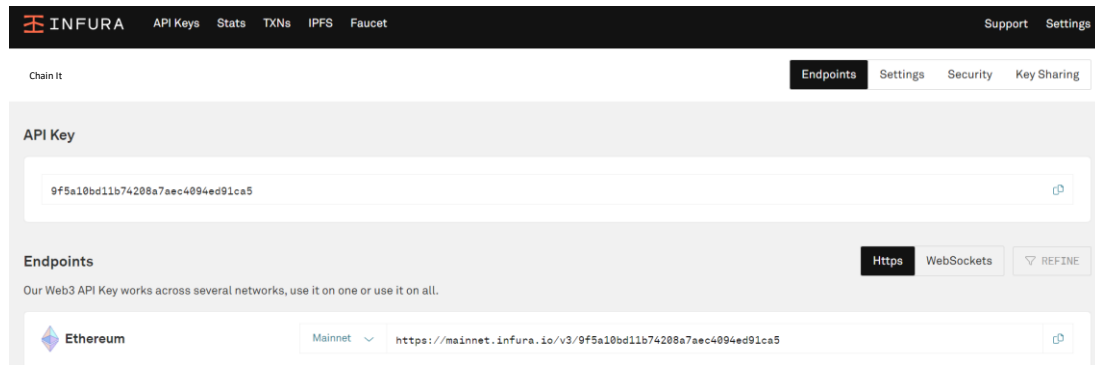


FIGURE 4.5.2.1 – INTERFACE INFURA
SOURCE : CHAIN IT

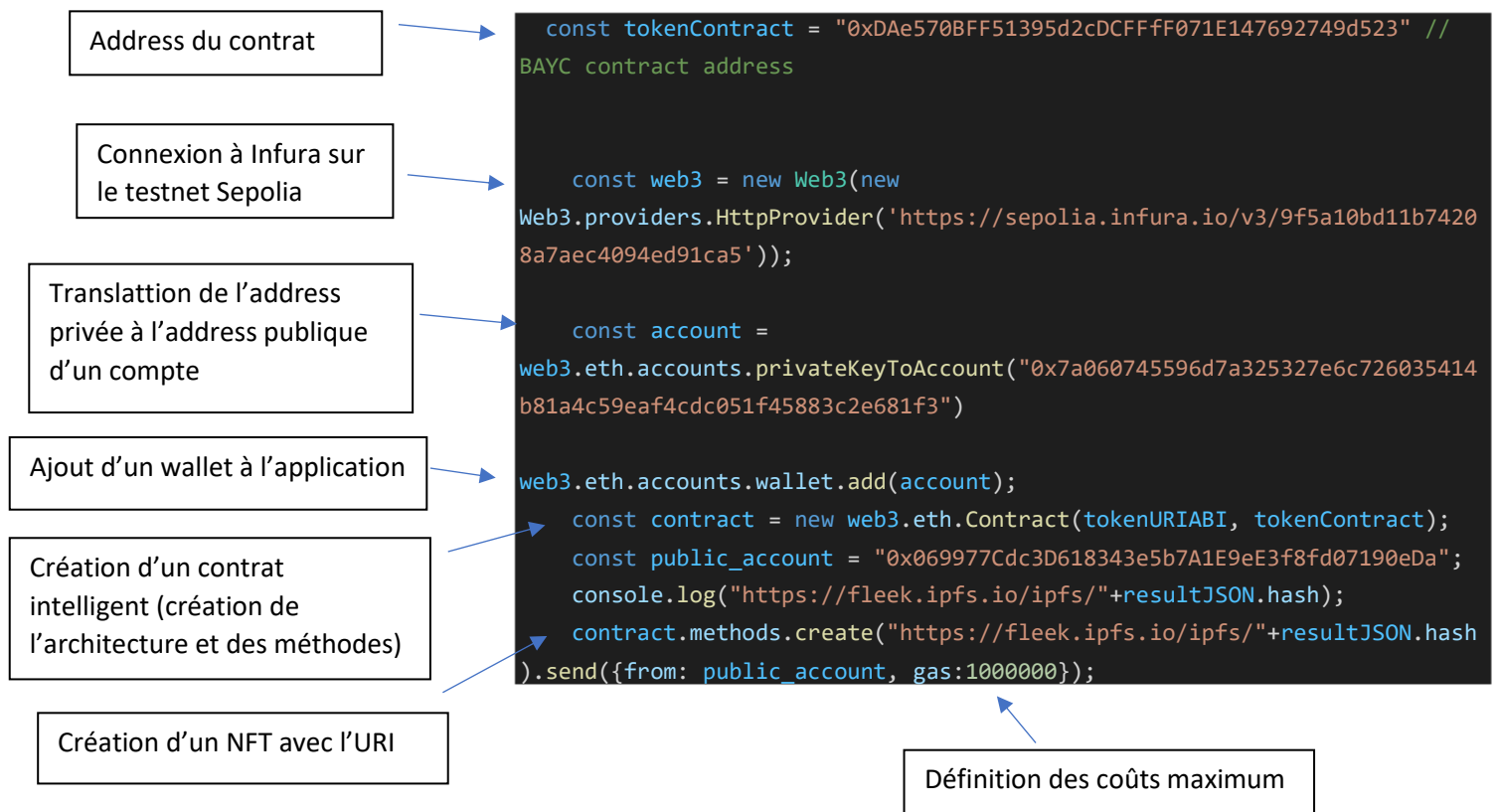


FIGURE 4.5.2.2 – CREATION D'UN DIPLOME DANS LA BLOCKCHAIN
SOURCE : CHAIN IT

4.6. L'organisation et la gestion du projet

Nous avons réalisé ce projet tout au long de l'année avec un calendrier qui est le suivant :

Le choix du projet se fera en octobre, au tout début de l'année scolaire ; nous avons consacré notre premier semestre (durant la période école) à étudier le contexte et à se poser différentes questions : Pourquoi choisir ce projet, quels sont les enjeux, est-ce nécessaire... C'est également durant cette période qu'une étude de marché a été réalisée. À la suite de cela, on a pu se demander quels outils à utiliser pour mener à bien notre projet, et pourquoi.

Les différents outils utilisés ont été séparé en deux catégories : outils de communication et outils techniques. On peut retrouver dans la première catégorie Teams, Whatsapp mais encore Github. Ce dernier a été très utile quant à son utilisation pour se partager nos différents scripts et code de notre application web, que ce soit entre nous ou notre encadrant.

Pour les outils techniques, on a choisi html, php, node.js et Magic Auth en ce qui concerne l'application web, Fleek storage pour la base de données et Ethereum pour la blockchain.

Durant l'année, nous avons des réunions hebdomadaires avec notre enseignant encadreur afin de mener ce projet à bien tout du long. Ce sont ces réunions, réalisées d'abord sur Teams puis sur Google Meet, qui nous ont permis d'avancer et d'éviter de se diriger vers la mauvaise direction grâce à l'encadrant qui nous prodiguait des précieux conseils.

Durant la période école, nous avons également des créneaux spéciaux pour pouvoir étudier notre projet, ce qui nous laissait du temps afin d'avancer, sans être ensevelis de cours à côté. Cette organisation de la part de l'école nous a facilité la réalisation du projet et de son bon déroulement.

Nous avons également des réunions journalières, sans l'encadrant, afin de faire le point avec l'équipe et de savoir comment avance chacun.

L'équipe a été divisé en trois parties, comme on pourrait le faire avec ce projet : Nicolas s'occupait de la partie Blockchain, en réalisant les contrats intelligents et manipulant Ethereum, Cédric s'est occupé de la partie application web (partie front end avec les langages html, css, javascript, et back end avec nodeJS) et de la gestion des authentifications avec Magic Auth. Williams, avec l'aide de Cédric, s'est occupé de la partie base de données en maniant Fleek storage.

4.7. Axes d'améliorations

Notre application décentralisée comporte quelques fonctionnalités qui peuvent évoluer ou être amélioré. Il faut tout pouvoir importer en masse une classe voire une promotion d'un seul coup. En mettant un fichier en paramètres de l'importation, avec les diplômes numérisés correspondant, nous pourrions générer les NFTs des étudiants en une seule fois.

D'autre part, le retour de l'expérience utilisateur sur le front-end de notre application nous permettra de l'améliorer afin d'être le plus intuitif et efficace possible. Nous devrons probablement faire une refonte totale ou simplement alléger ou durcir quelques points clés pour l'utilisateur.

Pour finir, la sécurisation de notre site est primordiale. Sécuriser et chiffrer les tokens en les personnalisant pour chaque étudiant est un point clé de la sécurisation que nous pouvons améliorer.

4.8. Nos difficultés

Dans la partie précédente, on énonçait l'importation en masse d'étudiant. Cette fonctionnalité n'a pas pu être mis en place dans cette première version à cause d'une mauvaise gestion entre le fichier csv (contenant les données de tous les élèves) et des diplômes numérisés en PDF. Nous avons essayé de gérer les deux ensembles sans succès.

Ensuite, nous avons eu des problèmes d'interaction multithread. En effet, notre application utilise le framework NodeJS car nous étions dépendant de fleek qui est compatible uniquement avec NodeJS. Ce framework fait difficilement de bonnes interactions entre le front-end et le back-end. Lorsque nous voulions lancer des fonctions asynchrones (des fonctions s'exécutant en parallèle du code) pour interroger le contrat intelligent, nous ne pouvions récupérer la variable au même timing que l'exécution de notre script. Nous avons dû faire une interaction depuis l'HTML afin de pouvoir récupérer la variable et la stocker.

De plus, trois parties indépendantes ont dû être fait en parallèle à savoir, la partie Fleek, la partie Blockchain, et la partie Front-end. Lors de la fusion des trois ensembles, nous avons découvert des problèmes d'interaction qui ont juste demandé du temps afin d'être corrigé.

Pour finir, le contrat intelligent est lisible de tous. La sécurisation est essentielle sinon n'importe qui pourrait se servir de notre contrat. Il a fallu écrire le code tout en pensant à sa sécurisation. Nous avons donc implémenté que dans chaque fonctions, l'utilisateur ou le portefeuille appelant la fonction soit contrôlé. Si l'adresse est autorisée, alors la fonction s'exécute, sinon la fonction s'arrête et aucune exécution n'est faite.

4.9. Certification d'un diplôme par un recruteur

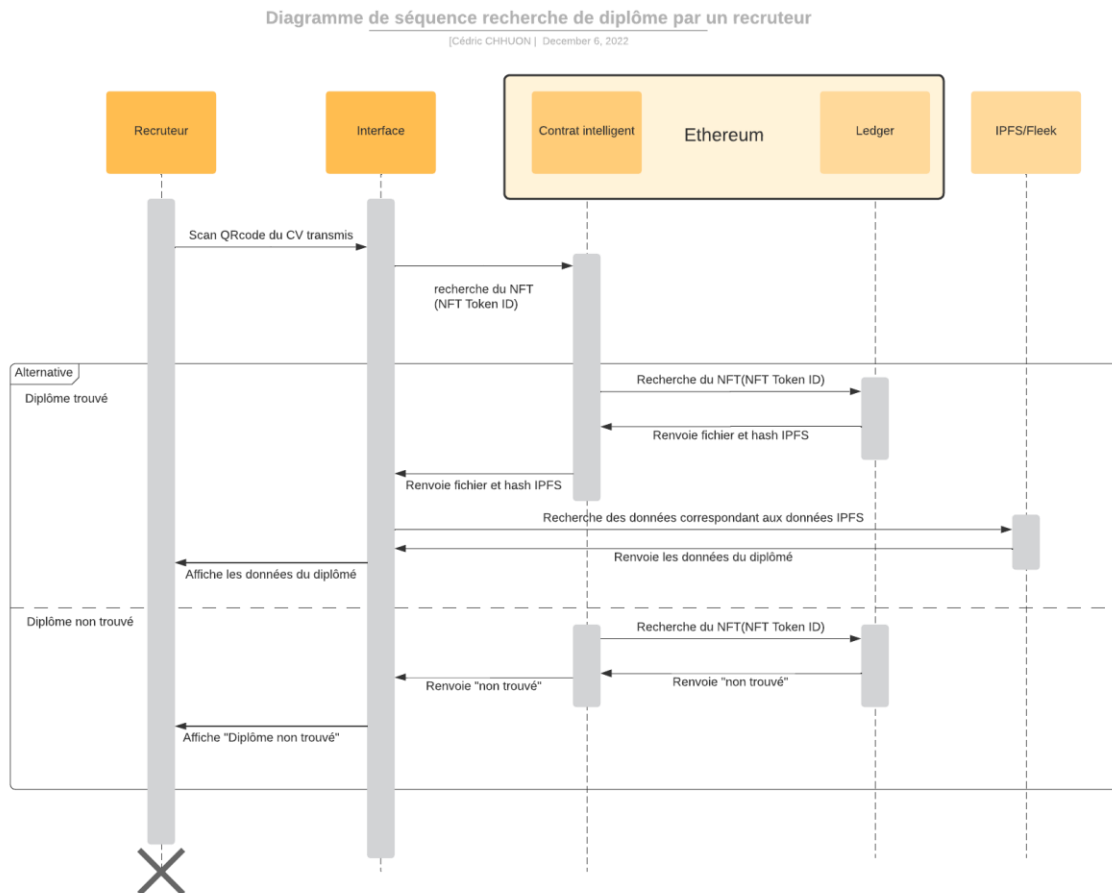


FIGURE 3.3.1 – DIGRAMME DE SEQUENCE DE LA VERIFICATION D'UN DIPLOME PAR LE RECRUTEUR

SOURCE : CHAIN IT

5. Conclusion

Actuellement, le monde du recrutement fait face à une vague de faux diplômes qui engendre une perte de confiance dans les CV. Les établissements scolaires et les entreprises y sont fortement impactés et mettent en place des moyens de vérification. Notre groupe Chain It propose une solution de certification des diplômes pour transformer les CV en s'appuyant sur la propriété d'immuabilité de la blockchain. Durant cette année, notre groupe a pu travailler, malgré les difficultés, afin de mener à bien le projet et de proposer une solution au problème. L'application web permet d'y mettre des diplômes, certifiés par la blockchain, et sont consultables à tout moment, en un clic, par les recruteurs.

6. Annexe

6.1. Bibliographie

6.1.1. Documentation de l'étude du sujet

<https://verifcv.com/comment-les-recruteurs-detectent-les-faux-diplomes-2/>

https://www.youtube.com/watch?v=boIKpua8jP0&ab_channel=GroupdiplomaVerifdiploma

https://fr.wikipedia.org/wiki/Contrat_intelligent

<https://www.economie.gouv.fr/entreprises/blockchain-definition-avantage-utilisation-application#>

<https://academy.binance.com/fr/articles/the-evolution-of-the-internet-web-3-0-explained>

https://www.youtube.com/watch?v=on5VXHlzTYA&ab_channel=Everycheck

https://www.youtube.com/watch?v=iSttHhBspkk&ab_channel=LeParisien

<https://www.afip-detective.com/la-verification-de-curriculum-vitae/#:~:text=Le%20Code%20du%20travail%20stipule,pr%C3%A9sentes%20sur%20un%20curriculum%20vitae%E2%80%A6>

<https://certificate.bcdiploma.com/check/D1AE259E5D4C8FF00D641E95CA553A9140545AF2DDAB4C83C87A712BCBAE5E2DN0FKQXAxL1VTeks0RXIjOWIBWDNYT05zRURnQVp5RVEzd25tQnJKOVdFbXhh eG0w>

<https://verifdiploma.com/>

<https://www.everycheck.com/>

6.1.2. Documentation sur le développement de la solution au niveau de la blockchain

<https://www.infura.io/>

<https://www.quicknode.com/guides/smart-contract-development/how-to-create-and-deploy-an-erc-721-nft>

6.1.3. Documentation sur le développement de la base de données Fleek Storage

<https://docs.fleek.co/>

<https://cryptoast.fr/interplanetary-file-system-ipfs-reseau-partage-fichiers-web-3/>

6.1.4. Documentation sur le développement de l'authentification avec Magic Auth

<https://magic.link/>

6.1.1. Documentation sur le développement du QR Code

<https://developers.cloudflare.com/workers/tutorials/build-a-qr-code-generator/>

6.2. Documents annexes

Technologies des chaînes de blocs et technologies de registre
distribué — Vocabulaire — ISO 22739:2020(F)

Rapport du Groupe de Travail : Jeton Non Fongible (JNF) ou Non-Fungible Token (NFT) – Christophe Ozcan – CEO, Crypto4All