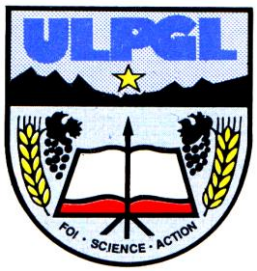
UNIVERSITÉ LIBRE DES PAYS DES GRANDS LACS

FACULTÉ DE SCIENCES ET TECHNOLOGIES APPLIQUÉES

DEPARTEMENT DE GENIE ELECTRIQUE ET INFORMATIQUE



BP. 368 GOMA

www.ulpgl.net

SYSTÈME DÈCENTRELISÉ BASÉ SUR LA BLOCKCHAIN POUR LA CERTIFICATION DES DIPLOMÉS D'UNE UNIVERSITÉ : CAS ULPGL

Par **KATEMBO KANIKI Joseph**

Travail présenté en vue de l'obtention du Diplôme d'Ingénieur Civil en Génie Électrique et Informatique

Option : Génie Informatique

Directeur : Prof. BARAKA MUSHAGE Olivier

Encadreur: Msc. KAMBALE WAMUHINDO Abednego

ANNÉE ACADÉMIQUE 2022 - 2023

Epigraphe

« *La blockchain est une technologie révolutionnaire qui a le potentiel de transformer de nombreux secteurs, y compris l'éducation.* »

Marc Andreessen

Dédicace

*À mon père **MUHINDO KANIKI Damien***

*À ma mère **KAVIRA MUSAVULI Jeannette***

À mes frères et sœurs

À tous mes amis et camarades

Pour leur soutien.

Remerciements

Nous adressons nos sincères remerciements :

A notre Dieu, Le Père tout puissant, Seigneur et Sauveur JESUS CHRIST. L'auteur et maitre de notre existence ainsi que de notre destinée.

Au Professeur Dr. Ir. BARAKA MUSHAGE Olivier, le directeur de ce travail, qui a accepté de prendre de son temps pour nous conduire et nous guider tout au long de l'élaboration et la rédaction de la présente œuvre scientifique. Ces mêmes sentiments de gratitude s'adressent à la personne du Master Ir. KAMBALE WAMUHINDO Abednego l'encadreur de ce travail, pour les judicieuses remarques et nobles conseils.

A toutes les autorités académiques de l'ULPGL/ GOMA pour l'encadrement et l'instruction durant ces années passées sous leur surveillance.

À nos parents MUHINDO KANIKI Damien et KAVIRA MUSAVULI Jeannette pour leur considérable soutient sous toutes les formes durant notre vie estudiantine.

À nos proches frères, sœurs et tantes avec qui nous partageons toutes les circonstances de la vie en famille ; K. ROGER, K. JEAN-LOUIS, K. EVARISTE, K. MOISE, K. PASCAL, K. JACQUES, S. MICHELLE.

À nos compagnons de lutte : B. SERAPHIN, K. DENISE, H. BONHEUR, B. DORCAS, R. JULIEN, K. MYRIAM, K. BENJAMIN, M. MUHESI,

Aux amis et connaissances entre autres : N. JACQUES, R. ANTHONY, N. AZARIA, K. PRINCE, M. MULOLWA, B. KAKOZI, K. MAVETYA, K. USHINDI, M. MWANABUTE

Enfin, que tout ceux-là ayant contribués de près ou de loin à la réalisation de notre travail et qui n'ont pas été cités trouvent ici l'expression de notre profonde gratitude.

Résumé

La vérification de l'authenticité des diplômes universitaires revêt une importance cruciale pour les employeurs et les établissements d'enseignement. Cependant, la falsification de diplômes est devenue une pratique courante, ce qui complique la tâche de vérification et entraîne des retards et des coûts importants dans le traitement des dossiers. Face à ces défis, il est impératif de développer des solutions novatrices visant à garantir l'intégrité des documents. Dans cette perspective, ce travail propose la mise en place d'un système décentralisé basé sur la technologie blockchain pour certifier les diplômés universitaires. Le processus de conception s'appuie sur l'utilisation de diagrammes UML et repose sur des mécanismes de cryptographie avancée afin d'assurer la préservation de l'intégrité des données. Ce système permet la conservation sécurisée d'informations relatives aux diplômés et offre un mécanisme de vérification reposant sur l'utilisation de hachages cryptographiques. En empêchant toute altération des documents, ce système améliore considérablement le processus de vérification des diplômes et contribue à renforcer la réputation des institutions en garantissant l'authenticité de leurs titres délivrés. Ce système constitue une solution de base et adaptable aux besoins évolutifs du domaine de la certification des diplômés.

Mots-clés : Système décentralisé, Blockchain, Certification des diplômes, Sécurité Informatique

Abstract

The verification of the authenticity of university diplomas holds crucial importance for employers and educational institutions. However, the falsification of diplomas has become a common practice, complicating the verification task and resulting in delays and significant costs in document processing. Faced with these challenges, it is imperative to develop innovative solutions aimed at ensuring document integrity. In this context, this study proposes the implementation of a decentralized system based on blockchain technology for certifying university graduates. The design process relies on the use of UML diagrams and incorporates advanced cryptographic mechanisms to ensure the preservation of data integrity. This system enables the secure storage of information pertaining to graduates and provides a verification mechanism based on the use of cryptographic hashes. By preventing any alteration of documents, this system significantly enhances the diploma verification process and contributes to bolstering the reputation of institutions by guaranteeing the authenticity of the conferred diplomas. The designed system represents a fundamental and adaptable solution to the evolving needs of the field of diploma certification.

Keywords: Decentralized System, Blockchain, Diploma Certification, Computer Security

Table des matières

Liste des abréviations

AC	Autorité de Certification
AES	Advanced Encryption Standard
API	Application Programming Interface
CRL	Liste de révocation de certificats
CU	Cas d'utilisation
CV	Curriculum vitae
DApps	Decentralized applications
DAO	Decentralized Autonomous Organization
DeFi	Decentralized finance
DES	Data Encryption Standard
ECC	Elliptic Curve Cryptography
ESU	Enseignement Supérieur et Universitaire
ETH	Ether
EVM	Ethereum Virtual Machine
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
ID	Identifiant
IPFS	InterPlanetary File System
IDE	Integrated Development Environment
JWT	JSON Web Token
KE	Encryption Key
KD	Decryption Key
MIT	Massachusetts Institute of Technology
NIST	National Institute of Standards and Technology
NFT	Non-fungible tokens

NoSQL	Not Only SQL
PDF	Portable Document Format
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PoW	Proof-of-Work
PoS	Proof-of-Stake
PoA	Proof-of-Authority
P2P	Peer-to-Peer
RDC	République démocratique du Congo
RC4	Rivest Cipher 4
RSA	Rivest–Shamir–Adleman
RPoW	Reusable Proofs of Work
SHA-3	Secure Hash Algorithm 3
SHA-256	Secure Hash Algorithm 256
SMR	State Machine Replication
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
THTD	Trust Hierarchy of Tiers Devices
TLS	Transport Layer Security
TCP/IP	Transmission Control Protocol/Internet Protocol
UI	User Interface
ULPGL	Université libre des pays de grands lacs
UML	Unified Modeling Language

Liste des tableaux

Liste des figures

0. Introduction générale

0.1. Contexte

La vérification et la validation des documents sont des tâches courantes dans de nombreux contextes, tels que l'éducation, l'emploi, la finance et le gouvernement [1]. Dans le domaine de l'éducation, les documents les plus courants à vérifier sont les diplômes et les certificats. Ces documents attestent des compétences et des connaissances acquises par les étudiants au cours de leurs études. Les diplômes et les certificats sont importants pour les étudiants car ils leur permettent d'accéder à des emplois, de poursuivre leurs études ou d'obtenir des bourses.
Les institutions d'enseignement et les employeurs ont besoin de pouvoir vérifier l'authenticité des diplômes et des certificats pour s'assurer que les étudiants ont bien les compétences et les connaissances qu'ils prétendent avoir [2].

Plusieurs pistes de solutions sont envisagées, tant au niveau international qu'au niveau national, pour aider des institutions universitaires à bien concevoir l'authenticité et la valeur des documents scolaires et académiques, de ce fait même faciliter la reconnaissance des documents contrefaits [3].
Les possibilités aujourd'hui offertes par les progrès technologiques pour la validation des documents sont loin d'être négligeables ; on peut citer par exemple la blockchain, qui permet de créer de la confiance sans recourir à un tiers tout en permettant d'inscrire de manière indélébile et infalsifiable des transactions, ou toutes autres informations, sur un registre public que chacun peut consulter librement. Il s'agit donc en quelque sorte d'une immense base de données mais avec l'originalité d'être ouverte, distribuée et infalsifiable. Les systèmes décentralisés basés sur la blockchain offrent un nouveau potentiel pour la vérification et la validation des documents [4].

0.2. Identification et formulation du problème

Falsifier un diplôme ou mentir sur son CV est une pratique courante, motivée par divers facteurs, notamment le chômage de masse, la concurrence accrue, les formations onéreuses ou la malhonnêteté intellectuelle.

De nos jours, avec l'évolution de l'informatique, il est plus facile de reproduire un diplôme avec des outils informatiques sophistiqués. Etant données les facilités avec lesquels les documents peuvent être falsifiés cela crée un climat de méfiance. Les employeurs et les institutions d'enseignement doivent donc prendre des précautions pour s'assurer de l'authenticité des diplômes. La difficulté à vérifier l'authenticité des diplômes présentés par les candidats oblige les institutions d'enseignement et les employeurs à faire valider les informations contenues dans ces diplômes par les universités. Cela alourdit le processus de traitement des dossiers, ce qui entraîne des coûts importants et une perte de temps.

L'absence d'un système pour résoudre le problème de falsification de diplôme dans le milieu universitaire en RDC s'avère un handicap pour les institutions d'enseignements et les entreprises. C'est dans ce cadre que nous nous intéressons à la mise en place d'un système décentralisé basé sur la blockchain pour la certification des diplômés d'une université, qui constitue une solution de plus, parmi tant d'autres dans le but de répondre aux besoins de « la vérification de l'authenticité de diplôme », pour ainsi atténuer l'ampleur de la falsification de diplôme dans le milieu universitaire en RDC.

Ce qui précède, nous amène à nous poser les questions suivantes :

- Comment peut-on mettre en place une solution informatique pour prévenir de façon optimale le problème d'usurpation des diplômes ?
- Comment le système proposé servirait-il dans le processus de traitement des dossiers de candidature dans une entreprise ou institution d'enseignement ?
- Pourquoi le système proposé aurait-il un effet positif sur les institutions qui octroient les diplômes ?

Ces questions nous serviront de guide tout au long de notre recherche.

0.3. Formulation des hypothèses

Les hypothèses sont des réponses anticipatives aux questions de recherche posées. Nous référant aux questions posées ci-haut, nous nous sommes proposé les hypothèses ci-dessous :

- Il serait possible d'éviter l'usurpation des diplômes en mettant en place un système informatique décentralisé basé sur la blockchain car ce dernier rendrait impossible la modification des documents et des données
- La mise en place de ce système décentralisé basé sur la blockchain pour la certification des diplômés pourrait réduire le temps de traitement de dossiers de candidature dans une entreprise ou institutions d'enseignement vu que toutes les informations pertinentes sur un candidat pourraient être stockées sur la blockchain, accessible à tout moment par les utilisateurs autorisés
- L'utilisation de la blockchain pourrait rendre crédible les diplômes octroyés par l'université aux yeux des employeurs et assainirait ainsi l'image de plusieurs institutions scolaires et académiques dans la société car elle donnerait une assurance sur l'authenticité des diplômes.

0.4. Justification du choix du sujet et motivations

La motivation pour ce sujet de recherche vient de notre curiosité et le besoin de repousser les limites de la connaissance en apprenant de nouvelles technologies, mais aussi d'un souci profond d'apporter aux universités de la RDC un système décentralisé et sécurisé de vérification de l'intégrité des diplômes. Ce système permettra aux employeurs de faire confiance aux diplômes présentés par des candidats aux embauches, et de gagner en temps dans la vérification de ces diplômes. Enfin, il renforcera la crédibilité des diplômes octroyés par les universités de la RDC. Le présent travail pourrait ouvrir la voie à la création d'une entreprise de gestion d'authenticité des documents numériques.

0.5. Énoncé des objectifs de recherche

0.5.1. L'objectif général

L'objectif global de ce présent travail est de proposer une solution informatique en mettant en place un système décentralisé et sécurisé de vérification de l'intégrité d'un diplôme d'une université.

0.5.2. Les objectifs opérationnels/spécifiques

En se basant sur l'objectif général, nous avons formulés les objectifs spécifiques comme suit :

- Établir une distinction entre les systèmes centralisés et décentralisés.
- Présenter les techniques utilisées en cryptographie.
- Présenter la technologie blockchain.
- Présenter les systèmes de stockage décentralisés.
- Étudier l'écosystème de développement de la blockchain.
- Choisir une plateforme de développement de *smart contract* (Un contrat intelligent est un programme informatique qui automatise l'exécution d'un contrat. Il est généralement stocké sur une blockchain et est exécuté lorsque certaines conditions prédéfinies sont remplies [5].).
- Concevoir le système décentralisé basé sur la blockchain pour la certification des diplômés d'une université.
- Créer un smart contrat pour la vérification et la certification des diplômés d'une université.
- Réaliser le système décentralisé basé sur la blockchain pour la certification des diplômés d'une université.

0.6. Méthodologie de recherche et délimitation du travail

0.6.1. Méthodologie de recherche

Pour atteindre nos objectifs, nous nous serviront de quelques méthodes et techniques notamment :

- La méthode expérimentale qui nous permettra de concevoir et de tester le système enfin s'assurer de son bon fonctionnement et voir aussi s'il répond aux critères fixés dans les objectifs de ce travail ;
- La technique documentaire nous permettra d'appréhender les différentes notions relatives à la réalisation de ce présent travail au travers la consultation des ouvrages et articles relatifs à notre sujet de recherche ;
- La technique d'interview nous permettra d'avoir des informations nécessaires à la contextualisation du présent travail ;
- La méthode comparative pour justifier notre choix de système décentralisé vis-à-vis de système centralisé.

0.6.2. Délimitation du travail

Dans notre travail, il est question de mettre en place un système décentralisé basé sur la blockchain pour la certification des diplômés d'une université. Les limites suivantes ont été appliquées à la portée de cette étude :

- Nous allons nous limiter à la certification des diplômés d'une seule université, nous prendrons le cas de l'ULPGL.
- La comparaison de la liste de finalistes de l'université et du ministère de l'enseignement supérieur et universitaire ne sera pas implémentée dans cette première version du système.
- La création manuelle d'un Template pour chaque type de diplôme (licence ou master) est une étape obligatoire avant la génération des diplômes.

0.7. Subdivision du travail

Hormis l'introduction générale et la conclusion générale, ce travail est subdivisé en 4 chapitres :

- Le premier introduit le concept de la cryptographie : ici on présentera le type de chiffrement, de fonctions hachages, la signature numérique, le certificat numérique et les infrastructures des systèmes à clé publique.
- Le deuxième introduit le concept de la Blockchain, les caractéristiques de la blockchain, et différentes définitions des concepts de base. On présentera le réseau pair à pair, ainsi que l'architecture de la Blockchain.
- Le troisième concerne la conception et l'analyse du système : ici on fera l'étude de l'existant tout en présentant les caractéristiques de diplôme à authentifier et on fera la conception fonctionnelle du système proposé à l'aide du langage UML.
- Le quatrième qui est le dernier concerne l'implémentation du système.

Chapitre 1 La cryptographie

1.1 Introduction

La blockchain est une technologie émergente qui a le potentiel de révolutionner de nombreux secteurs, tels que la finance, la santé, la supply-chain et la gouvernance [5]. La blockchain est une base de données distribuée, ce qui signifie qu'elle est partagée par un réseau d'ordinateurs. Chaque ordinateur du réseau stocke une copie complète de la base de données, ce qui rend très difficile sa modification ou sa falsification [6]. Les données de la blockchain sont également sécurisées par l'utilisation de la cryptographie.

Vu que la cryptographie est une technologie essentielle pour la blockchain, dans ce chapitre nous présenterons les notions de base de la cryptographie.

1.2 Définition

La cryptographie est la science qui utilise les mathématiques pour crypter et décrypter des données. À la croisée des mathématiques, de l'informatique et parfois de la physique, elle permet ce dont les civilisations ont besoin depuis qu'elles existent : « maintenir le secret ». Pour éviter la guerre, pour protéger un peuple, il faut parfois cacher certaines choses [7].

1.3 L'utilisation de la cryptographie

La cryptographie est traditionnellement utilisée pour dissimuler les messages de certains utilisateurs. Cet usage est d'autant plus préoccupant aujourd'hui que pour la communication sur internet, les données transitent dans des infrastructures dont la fiabilité et la confidentialité ne peuvent être garanties. Désormais, la cryptographie sert non seulement à protéger la confidentialité des données, mais également à garantir leur intégrité et leur

authenticité, ainsi la cryptographie ne prend en charge que les 4 premiers sur les 5 services de sécurité fondamentaux [6]:

- **La confidentialité** : Elle consiste à rendre l'information inintelligible à d'autres personnes que les acteurs de la transaction.
- **L'intégrité** : Elle consiste à déterminer si les données n'ont pas été altérées durant la communication.
- **L'authentification :** Elle consiste à assurer l'identité d'un utilisateur, c-à-d de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.
- **La non répudiation** : La non répudiation de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction.
- **La Disponibilité** : La disponibilité des informations signifie qu'elles doivent être accessibles à tous les utilisateurs autorisés, lorsqu'ils en ont besoin.

1.4 Le Chiffrement

La confidentialité est la préoccupation numéro un du cryptage. Ce problème est résolu par le concept de chiffrement. Il existe deux grandes familles d'algorithmes de chiffrement par clé : L'algorithme à clé secrète ou l'algorithme symétrique et l'algorithme à clé publique ou l'algorithme asymétrique [5].

a. Chiffrement Symétrique

En cryptographie classique, la clé de chiffrement et la clé de déchiffrement sont identiques : c'est une clé secrète que les tiers communicants doivent connaître et eux seuls connaissent. Ce processus de cryptage est dit symétrique.

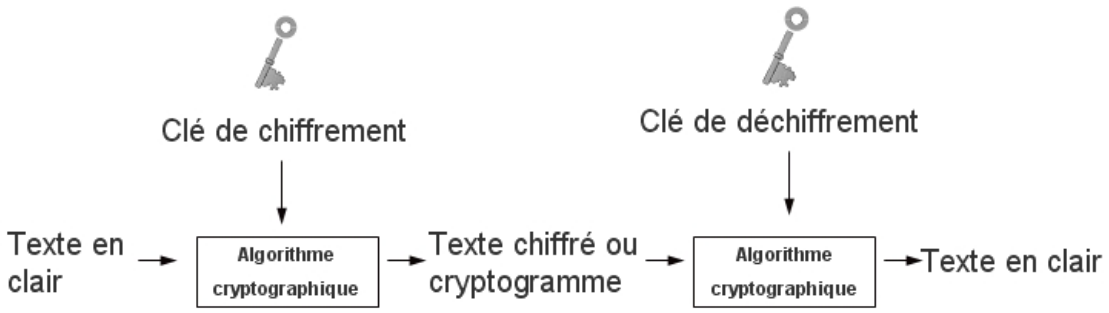


Figure 1-1 chiffrement symétrique [9]

Caractéristiques [9]:

- Les clés sont identiques : KE = KD = K, avec KE : clé de chiffrement et KD : clé de déchiffrement
- La clé doit rester secrète,
- Les algorithmes les plus répandus sont le DES (*Data Encryption Standard*), AES (*Advanced Encryption Standard*), 3DES (*Triple Data Encryption Standard*),
- Au niveau de la génération des clés, elle est choisie aléatoirement dans l'espace des clés,
- Ces algorithmes sont basés sur des opérations de transposition et de substitution des bits du texte clair en fonction de la clé,
- La taille des clés est souvent de l'ordre de 128 bits. Le DES en utilise 56, mais l'AES peut aller jusqu'à 256,
- L'avantage principal de ce mode de chiffrement est sa rapidité,
- Le principal désavantage réside dans la distribution des clés : pour une meilleure sécurité, on pratiquera à l'échange de manière manuelle. Malheureusement, pour de grands systèmes, le nombre de clés peut devenir conséquent. C'est pourquoi on utilisera souvent des échanges sécurisés pour transmettre les clés. En effet, pour un système à N utilisateurs, il y aura $N\,(N\,-\,1)\,/\,2$ paires de clés.

Les principaux algorithmes de chiffrement symétrique sont [10] :

- DES (Data Encryption Standard) : un algorithme de chiffrement par bloc à 56 bits, qui est désormais considéré comme obsolète.

- Triple DES (3DES) : une variante de DES qui utilise trois clés différentes pour chiffrer les données, ce qui augmente la sécurité.

- AES (Advanced Encryption Standard) : un algorithme de chiffrement par bloc à 128, 192 ou 256 bits, qui est considéré comme le plus sûr des algorithmes de chiffrement symétrique.

RC4 : un algorithme de chiffrement par flux, qui est utilisé dans de nombreux protocoles de communication, tels que SSL (Secure Sockets Layer) et TLS (Transport Layer Security).
Par la suite nous détaillerons deux de ces algorithmes notamment le DES et le AES, qui sont deux exemples de l'évolution du chiffrement symétrique : DES est un algorithme plus ancien qui est aujourd'hui considéré comme étant moins sûr. AES est un algorithme plus récent qui est considéré comme étant beaucoup plus sûr.

- **Algorithme de chiffrement du D.E.S**

D.E.S. est un crypto-système basé sur des blocs. Cela signifie que le D.E.S. n'encode pas les données immédiatement à l'arrivée des caractères, mais il divise principalement le texte brut en blocs de 64 bits, qu'il encode séparément puis réassemble[9].

L'algorithme repose principalement sur 3 étapes, en plus de la gestion spécifique de la clé :

- Permutation initiale
- Calcul médian (16 fois) : application d'un algorithme complexe appliqué en fonction de la clé
- Permutation finale

- **Advanced Encryption Standard (AES)**

En 1997, le NIST a lancé un appel d'offres pour un algorithme de chiffrement symétrique. Parmi les 15 algorithmes soumis, Rijndael a été choisi pour devenir la norme AES. AES est un algorithme à bloc de 128 bits qui utilise une variété de techniques de chiffrement, telles que la substitution, la permutation et XOR. Il est disponible en trois longueurs de clé : 128,

192 et 256 bits. AES est un algorithme de chiffrement sécurisé et performant, largement utilisé par les organisations gouvernementales et les appareils mobiles [11].

b. Chiffrement asymétrique ou à clé publique

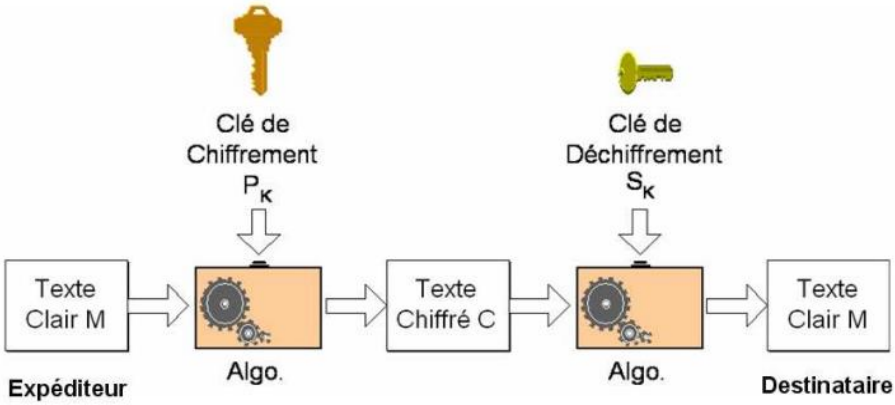


Figure 1-2 Chiffrement asymétrique [9]

La cryptographie asymétrique utilise deux clés, une clé publique et une clé privée. La clé publique est partagée avec tout le monde, tandis que la clé privée est gardée secrète.
La clé publique permet de chiffrer un message, mais elle ne permet pas de le déchiffrer. Seul le destinataire du message, qui possède la clé privée, peut le déchiffrer.
La recherche des clés par force brute est théoriquement possible, mais les clés sont trop grandes pour être crackées dans un temps raisonnable. En conséquence, la cryptographie asymétrique est plus lente que la cryptographie symétrique, qui utilise une seule clé pour chiffrer et déchiffrer les messages [12].
Par la suite nous essayons de comprendre le chiffrement RSA et ECC qui sont utilisés à plusieurs étapes du processus de blockchain pour sécuriser les transactions, les données et la confidentialité des utilisateurs.

- **Le chiffrement RSA**

Le cryptage RSA est un algorithme de chiffrement asymétrique, qui repose sur la théorie des nombres premiers. Il a été découvert en 1977 par Ron Rivest, Adi Shamir et Leonard Adleman, trois chercheurs du Massachusetts Institute of Technology. RSA est basé sur le fait qu'il est très facile de multiplier deux nombres premiers, mais très difficile de les factoriser, c'est-à-dire de trouver ces deux nombres premiers si l'on connaît seulement leur produit. Cette propriété rend le cryptage RSA très difficile à casser, car pour déchiffrer un message chiffré avec RSA, il faut trouver la clé privée, qui est une paire de nombres premiers. [13]

Génération des clés

Le RSA fonctionne à partir de deux nombres premiers, que l'on appellera p et q. Ces deux nombres doivent être très grands, car ils sont la clé de voûte de notre cryptage. Aujourd'hui, on utilise des clés de 128 à 1024 bits, ce qui représente des nombres décimaux allant de 38 à 308 chiffres !

Une fois ces deux nombres déterminés, multiplions-les.

On note n le produit :

$$\mathbf{n = p \times q,} \text{ et } \mathbf{z = (p - 1) \times (q - 1).}$$

Équation 1-1

Cherchons maintenant un nombre e (inférieur à z), qui doit nécessairement être premier avec z. Calculons ensuite l'inverse de e modulo z, que nous noterons d :

$$\mathbf{d \equiv e^{-1} \bmod((p - 1)(q - 1))}$$

Équation 1-2

Le couple (e, n) est la clé publique, et (d, n) est la clé privée. [14]

Cryptage et décryptage

Pour crypter un nombre par exemple **t**, il suffit de le mettre à la puissance **e**. Le reste modulo n représente le nombre une fois crypté.

$$\mathbf{c = t^e \bmod n}$$

Équation 1-3

Avec **c** le nombre crypté.

Pour décrypter, on utilise la même opération, mais en mettant à la puissance **d** :

$$\mathbf{t = c^d \bmod n}$$

Équation 1-4

Avec **t** le nombre à décrypté

Une fois e, d et n calculés, on peut détruire p, q et z, qui ne sont pas nécessaires pour crypter et décrypter. Pire encore, on peut calculer très rapidement la clé privée d à partir de p et q, il ne faut donc pas conserver ces nombres [14].

L'utilisation des courbes elliptiques

La cryptographie sur les courbes elliptiques (ECC) est un type de cryptographie asymétrique qui utilise les propriétés mathématiques des courbes elliptiques pour garantir la sécurité. Elle a été proposée indépendamment par Koblitz et Miller dans les années 1980. ECC présente plusieurs avantages par rapport aux autres algorithmes de cryptographie asymétrique, comme RSA. Tout d'abord, ECC est plus efficace, car il utilise des clés plus courtes. Cela permet de réduire la consommation de ressources. Deuxièmement, ECC est plus sûr que RSA pour les mêmes longueurs de clés. Cela est dû au fait que le problème de la multiplication de points sur les courbes elliptiques est considéré comme plus difficile que le problème de factorisation des nombres premiers. En raison de ces avantages, ECC est de plus en plus utilisé dans divers domaines, notamment la sécurité des communications, la signature numérique et la cryptographie à clé publique [15].

D'une manière générale, sur R, les courbes elliptiques seront considérées comme l'ensemble des couples (x, y) tels que :

$$y^2 = x^3 + ax + b$$

Équation 1-5

dont le discriminant $-(4a^3 + 27b^2)$ est non nul.

Pour la dessiner, pour a et b fixés, on calcule y tel que :

$$y = \sqrt{x^3 + ax + b}$$

Équation 1-6

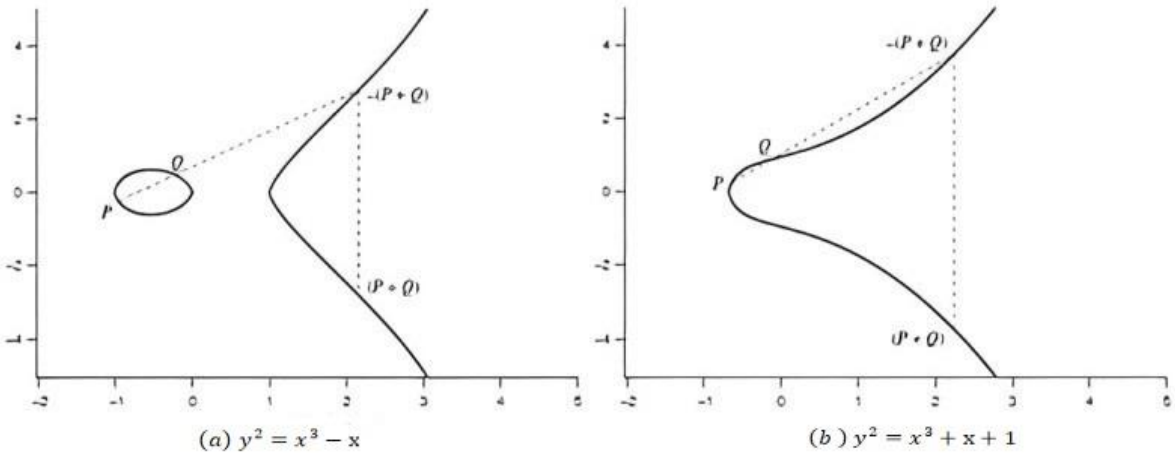


Figure 1-3 Exemple de la courbe elliptique [9]

Étant donné une courbe E, définie par une équation dans un corps fini (telle que E : $y^2 = x^3 + ax + b$), la multiplication de points est définie comme l'ajout répété d'un point le long de cette courbe. Notons kP = P + P + P + … + P pour un scalaire (entier) k et un point P = (x, y) qui se trouve sur la courbe E. La sécurité de l'ECC moderne dépend de la difficulté de déterminer k à partir de Q = kP étant donné les valeurs connues de Q et P si k est grand. Il s'agit du problème du logarithme discret pour les courbes elliptiques : $\log_P(Q)$. En effet, l'ajout de deux points sur une courbe elliptique (ou l'ajout d'un point à lui-même) donne un troisième point sur la courbe elliptique dont l'emplacement n'a pas de relation immédiatement évidente avec les emplacements des deux premiers, et en répétant cela plusieurs fois on a un point kP cela peut être essentiellement n'importe où. Inverser ce processus, c'est-à-dire étant donné Q=kP et P, et déterminer k, ne peut être fait qu'en essayant tous les k possibles, ce qui est impossible de les essayer tous dans un temps raisonnable [16].

1.5 Fonctions de hachage

Les fonctions de hachage ont de nombreuses applications pratiques allant de la simple vérification de l'intégrité des fichiers et du stockage des mots de passe à l'utilisation dans les protocoles et algorithmes cryptographiques. Ils sont utilisés dans les réseaux pair à pair (P2P), le partage de fichiers P2P, les filtres Bloom, les arbres Merkle et les tables de hachage distribuées. Les fonctions de hachage sont utilisées pour créer des résumés de longueur fixe de chaînes d'entrée de n'importe quelle longueur. Les fonctions de hachage sont sans clé et assurent l'intégrité des données.

Il existe trois propriétés de sécurité qui doivent être respectées, en fonction du niveau d'intégrité [17] :

- **Résistance pré-image** : cette propriété indique que si on lui donne une valeur y, il est informatiquement impossible (presque impossible) de trouver une valeur x telle que $h(x) = y$. Ici, h est la fonction de hachage, x est l'entrée et y est le hachage. La première propriété de sécurité exige que y ne puisse pas être calculé de manière inverse en x. x est considéré comme une pré-image de y, d'où le nom de résistance pré-image. C'est ce qu'on appelle également une propriété à sens unique.
- **Deuxième résistance pré-image** : La deuxième propriété de résistance pré-image indique qu'étant donné x, il est impossible par calcul de trouver une autre valeur x' telle que $x' \neq x$ et $h(x') = h(x)$. Cette propriété est également connue sous le nom de faible résistance aux collisions.
- **Résistance aux collisions** : la propriété de résistance aux collisions indique qu'il est impossible par calcul de trouver deux valeurs distinctes x' et x telles que $h(x') = h(x)$. En d'autres termes, deux messages d'entrée différents ne doivent pas être hachés vers la même sortie. Cette propriété est également connue sous le nom de forte résistance aux collisions.

Ces propriétés de sécurité sont décrites dans le diagramme suivant :

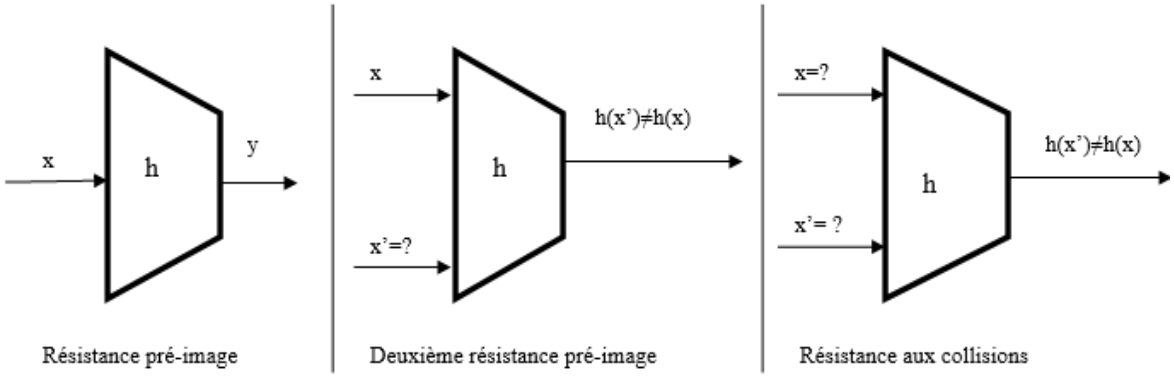


Figure 1-4 Propriétés de sécurité pour une fonction de hachage [17]

a. Algorithmes de hachage sécurisés

- **SHA-256**

SHA-256 a une limite de taille de message d'entrée de 2^{64}-1 bits. La taille du bloc est de 512 bits et la taille des mots est de 32 bits. Le résultat est un résumé de 256 bits. La fonction de compression traite un bloc de message de 512 bits et une valeur de hachage intermédiaire de 256 bits. Il y a deux composants principaux de cette fonction : la fonction de compression et une planification des messages.

À un niveau élevé, SHA-256 peut être visualisé dans le diagramme suivant :

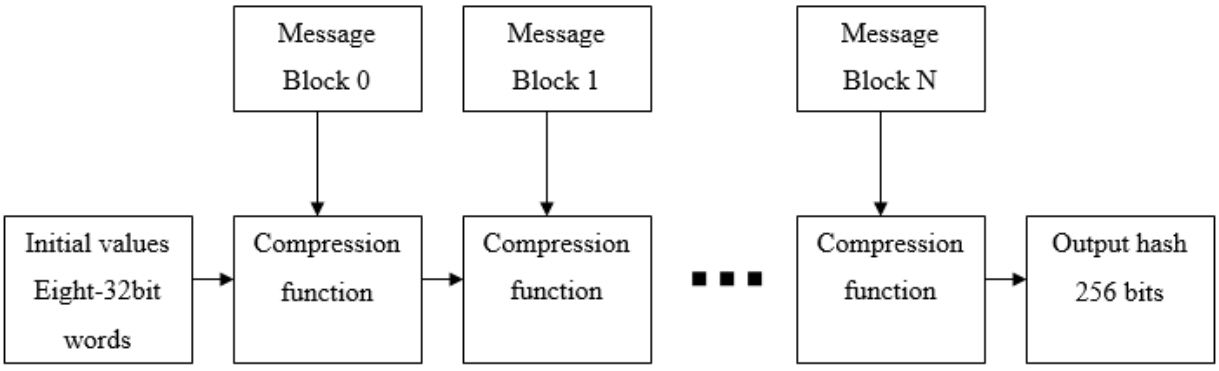


Figure 1-5 Principe de fonctionnement de SHA-256 [17]

- **SHA-3 (Keccak)**

SHA-3 (Keccak) est une fonction de hachage cryptographique qui fonctionne en absorbant les données d'entrée dans un état interne, puis en générant une sortie de hachage en fonction de cet état. L'état interne est une série de 1600 bits, qui est initialement initialisé à une valeur aléatoire. Les données d'entrée sont divisées en blocs de 1024 bits, qui sont ensuite ajoutés à l'état interne. L'état interne est ensuite transformé en utilisant une série d'opérations mathématiques, qui sont conçues pour rendre difficile la prédiction de la sortie de hachage à partir de l'entrée. La sortie de hachage est une chaîne de 256 bits, qui est unique pour chaque ensemble d'entrées. SHA-3 est considéré comme une fonction de hachage très sécurisée, et il est utilisé dans de nombreux protocoles et applications cryptographiques [18].

b. Arbre de Merkle

Un type particulier de structure de stockage de données basé sur des fonctions de hachage s'appelle Arbre de Merkle (Merkle tree) :

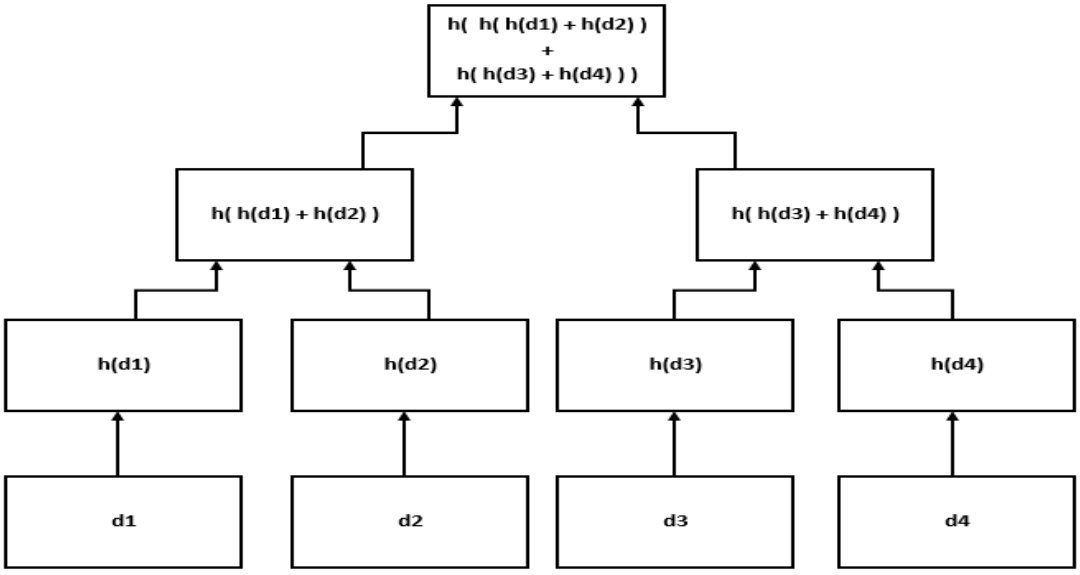


Figure 1-6 Arbre de merkle [19]

Pour le construire on commence par découper le fichier en plusieurs blocs, ici d1, d2, d3 et d4 dans ci-dessous. On obtient ainsi h(d1), h(d2), h(d3) et h(d4). On concatène ensuite ces empreintes deux à deux (par exemple h(h(d1) + h(d2))), et on re-hache l'ensemble obtenu. Ces opérations sont réitérées jusqu'à obtenir une unique empreinte, la racine de notre arbre. Cette racine dépend de toutes les opérations faites précédemment, et donc directement des données initiales. Les propriétés des fonctions de hachage, telles qu'expliquées précédemment, garantissent de pouvoir vérifier que des données correspondent bien à une racine de Merkle, car modifier une donnée revient à modifier son empreinte. Dans notre exemple, pour vérifier l'intégrité de d3, nous avons besoin de h(d4), h(h(d1) + h(d2)) et de la racine. Il n'est donc pas nécessaire de télécharger toutes les données [19].

1.6 La signature numérique

La signature numérique a pour but de garantir les propriétés d'authenticité de l'émetteur d'un message, l'intégrité des données et la non-répudiation.

C'est une composante de la cryptographie à clé publique / clé privée, également appelée cryptographie asymétrique. Chaque utilisateur possède un couple de clés, constitué d'une clé publique (connue par l'ensemble du réseau), notée **« e »** et d'une clé privée (connue uniquement par son possesseur), notée « d ». Pour les fonctionnalités de chiffrement, la clé publique sert à chiffrer et la clé privée à déchiffrer. Pour la signature, ce sera l'inverse.

Pour signer un message m, on calcule tout d'abord son empreinte avec une fonction de hachage h(m). Une fonction de signature S publique associe une clé privée d et un texte en clair m. On calcule la signature $s_m = S(d, h(m))$.

On envoie le message original m et la signature associée s_m.. Une fonction de vérification V associe une clé publique « e » et une signature s_m. Le destinataire du message reçoit un texte en clair, que l'on notera m' et une signature s_m.. En appliquant la fonction de vérification de la signature avec la clé publique du destinataire « e », $V\ (e, s_m, m')$, on vérifie à la fois que l'empreinte du message reçu correspond à celle de la signature (intégrité), et que la signature a bien été faite par la bonne personne (authenticité). Tout le monde connaît e, donc peut s'assurer que c'est le bon émetteur, car lui seul connaît le « d » associé. Combiné au fait que

l’on respecte les propriétés d’intégrité et d’authenticité, on obtient également la propriété de non répudiation [19]. Une signature numérique se compose donc de deux phases [20] :

- Phase de signature : L'expéditeur hache les données et les signe avec une signature numérique générée à l'aide de sa clé privée. Le hachage signé est envoyé au destinataire avec les données originales.
- Phase de vérification Les données signées sont déchiffrées avec la clé publique de l'expéditeur et comparées à la valeur de hachage des données originales.

Le fonctionnement d'une fonction de signature numérique générique est illustré dans le schéma suivant :

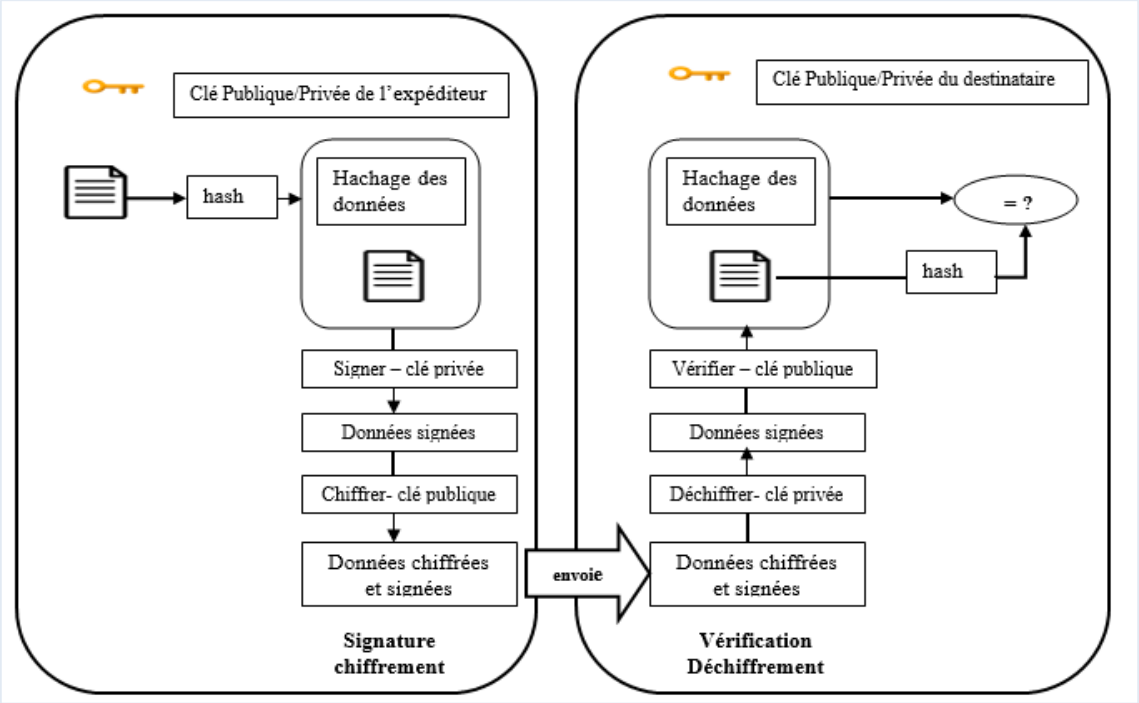


Figure 1-7 Phases de signature-chiffrement et de vérification-déchiffrement [20]

1.7 Certificat numérique

Jusque-là, nous avions toujours supposé que les clés publiques étaient distribuées de manière sécurisée. Si cette hypothèse n'est pas vérifiée, les schémas asymétriques peuvent conduire à des attaques de « Man in the Middle » qui est une attaque informatique dans laquelle un attaquant s'interpose entre deux parties communicantes afin d'intercepter, modifier ou supprimer les données transmises [21]. Le scénario suivant illustre une telle attaque :

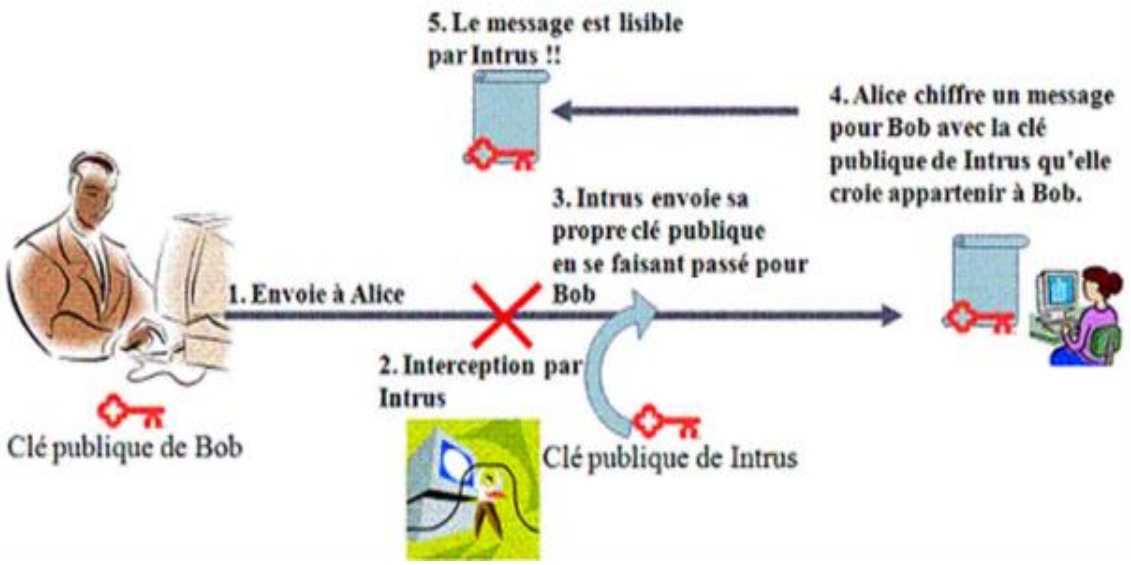


Figure 1-8 Man in the middle [21]

La solution au problème dit de « Man in the Middle » consiste à utiliser un certificat numérique qui assure un lien entre l'identité et la clé publique correspondante dans un document numérique signé par un tiers de confiance appelé autorité de certification. Il existe plusieurs modèles de réseau avec Tiers de Confiance, avec deux modèles extrêmes, le modèle hiérarchique et le modèle distribué. Le système tiers de confiance hiérarchique largement utilisé est le modèle X.509 [21].

a. Chaîne de certificats

Pour obtenir un certificat numérique, les clients doivent soumettre une demande à un organisme accrédité. Il transmet avec sa requête sa clé publique. L'organisme construit un

certificat incorporant la clé publique du client, il signe le certificat à l'aide de sa clé privée. L'autorité de certification délivre un certificat signé contenant la clé publique et l'identité exacte du propriétaire. Toute personne faisant référence à ce certificat fait confiance à l'autorité qui l'a délivré, et peut être sûre de l'authenticité de la clé publique qu'il contient [21].

Il existe la certification par le système hiérarchique de Tiers de Confiance (THTD) qui est un processus qui permet de vérifier l'identité et l'authenticité d'un objet numérique, tel qu'un certificat électronique, un document ou un fichier. Ce système repose sur une hiérarchie d'autorités de certification (AC), qui sont des organismes indépendants et réputés pour leur fiabilité [22]. On présentera par la suite la décentralisation de la confiance vue que la blockchain est basée sur ce principe.

b. Décentralisation de la confiance (Web-of-trust PGP)

La toile de confiance est une décentralisation de la confiance, ce qui signifie qu'elle n'est pas centralisée par une autorité de certification. Au lieu de cela, la confiance est distribuée entre les utilisateurs de PGP

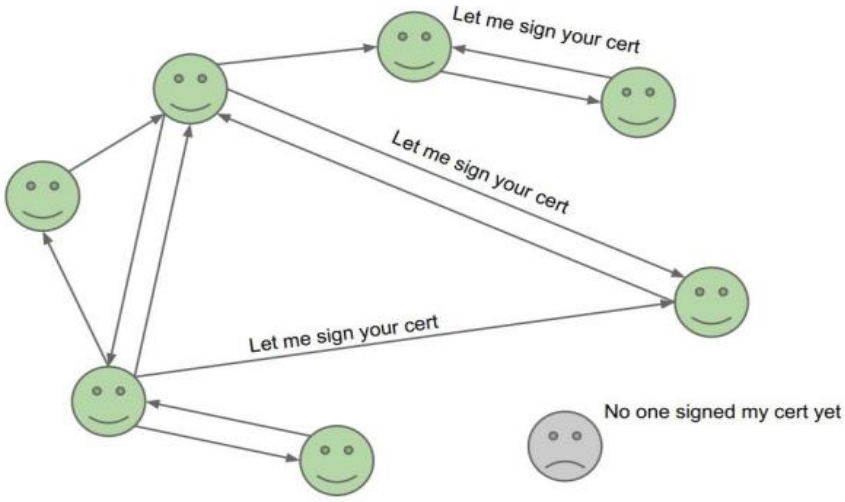


Figure 1-9 Système décentralisé de Tiers de Confiance [22]

Chaque utilisateur de PGP est responsable de la vérification des clés publiques des autres utilisateurs. Pour ce faire, l'utilisateur peut utiliser sa propre clé publique pour signer la clé publique d'un autre utilisateur. Cette signature électronique atteste que l'utilisateur a rencontré l'autre utilisateur en personne et qu'il est convaincu de l'authenticité de sa clé publique.

Plus une clé publique est signée par d'autres utilisateurs, plus elle est considérée comme fiable. Cela est dû au fait que la probabilité que toutes les signatures soient frauduleuses est très faible.

1.8 Infrastructure des systèmes à clé publique.

L'Infrastructure à Clé Publique ou PKI (Public Key Infrastructure) regroupe toutes les dispositions techniques et organisationnelles nécessaires à la gestion d'un système cryptographie à clé publique.

Les fonctions principales d'un PKI sont [21] :

- Emission de paires de clés (clé publique, clé privée)
- Attribution d'identifiants uniques aux participants au réseau
- Conservation des clés
    1. Dépôt de clés, utile quand il s'agit de garantir la continuité d'un service en cas de changement de titulaire
    2. Archivage des clés, utile pour pouvoir relire de vieux messages
    3. Sauvegarde des clés, utile en cas de perte de clé
    4. Récupération de clé perdues
- Émettre et publier des certificats
- Révoquer les certificats en mettant à jour la liste de révocation de certificats (CRL)
- Fixer leur durée de validité
- Archiver les certificats expirés (important pour les problèmes de non-répudiation)
- Donner une datation sécurisée

1.9 Séquence de vérification de certificats

Pour valider un certificat numérique, il faut suivre les étapes suivantes :

1. La signature valide-t-elle le document ? On fait la vérification de hachage
2. La clé publique est-elle celle du certificat ?
3. Le certificat est-il celui du sujet ?
4. Le certificat est-il validé par le CA ? Valider toute la chaîne de certification, jusqu'à la racine
5. Le certificat racine est-il approuvé ? Doit déjà être en possession du certificat racine
6. Le certificat figure-t-il dans une CRL ? On vérifie dans la CRL si le certificat n'est pas révoqué [22]

1.10 Conclusion Partielle

Dans ce chapitre nous avons abordé de manière succincte la cryptographie. Nous avons compris que la cryptographie ne suffit pas à garantir l'intégrité et la traçabilité des données. Ainsi par la suite nous parlerons de la blockchain, qui est un registre distribué qui enregistre les transactions de manière sécurisée et transparente. Elle utilise la cryptographie pour sécuriser les données et les rendre inviolables.

Chapitre 2 La blockchain

2.1 Introduction

La protection des données sur Internet est un enjeu majeur ; c'est pourquoi les chercheurs se concentrent désormais sur les technologies de cryptage et de sécurité des données comme la Blockchain. La blockchain fonctionne sur un réseau pair à pair d'ordinateurs, tous exécutant le protocole et conservant une copie identique des blocs de transactions, qui sont transmis sans intermédiaires et sans aucune autorité centrale à un mécanisme appelé consensus. La Blockchain elle-même est un grand livre public qui enregistre toutes les transactions depuis le bloc genèse (premier bloc) jusqu'à aujourd'hui [23].

Dans ce chapitre, nous présentons la nouvelle technologie « Blockchain », qui promet d'assurer cette confiance numérique sans avoir besoin d'une autorité centrale. Nous montrons comment l'utiliser pour partager et contrôler en toute sécurité des informations entre des parties qui ne se font pas nécessairement confiance. Ce chapitre comprend : Historique de la Blockchain, définition, caractéristiques, fonctionnement, composants et rôle de l'algorithme de consensus.

2.2 Quelques définitions

a. Transactions

Le registre d'une blockchain est constitué de transactions. La transaction la plus simple est l'échange d'actifs numériques d'un nœud à l'autre. Elle comprend au minimum une adresse de destinataire, une adresse d'expéditeur et une valeur. Une transaction peut également contenir une ou plusieurs entrées, qui sont des références aux sorties de transactions précédentes [24].

b. Bloc

Un bloc est une structure de données qui regroupe des ensembles de transactions et est distribuée à tous les nœuds du réseau. Le bloc contient un en-tête, une structure d'empreintes cryptographiques garantissant l'intégrité de son contenu, les *smart contracts* sous forme compilée, les transactions, et les données, ainsi que son maillage avec le bloc précédent. Les blocs sont créés par les mineurs à l'aide d'un algorithme de consensus et sont ajoutés à la blockchain, qui enregistre l'ensemble des transactions effectuées sur le réseau. La difficulté et le taux de hachage sont des mécanismes clés de la création de nouveaux blocs [25].

c. Consensus

Les mécanismes de consensus sont des protocoles qui garantissent que tous les nœuds (les périphériques de la chaîne qui gèrent la chaîne et (parfois) traitent les transactions) sont synchronisés les uns avec les autres et conviennent des transactions légitimes à ajouter à la chaîne. Ces mécanismes de consensus sont cruciaux pour le bon fonctionnement de la Blockchain. Ils garantissent que tout le monde utilise la même Blockchain. N'importe qui peut soumettre des éléments à ajouter à la Blockchain. Par conséquent, il est essentiel que toutes les transactions soient vérifiées en permanence et que la Blockchain soit vérifiée en permanence par tous les nœuds. Sans un bon mécanisme de consensus, la Blockchain peut faire face à diverses attaques [26].

d. Minage

Le minage de blockchain est un processus qui vérifie et enregistre les transactions sur une blockchain. Le minage c'est l'utilisation de la puissance de calcul informatique afin de résoudre un problème cryptographique simple dans son principe mais demandant une puissance de calcul importante. Un mineur est un particulier ou une société qui connecte sur le réseau une ou plusieurs machines équipées pour effectuer du minage. Un mineur est rémunéré lorsqu'il est le premier à trouver la solution du problème cryptographique posé par le minage [24].

2.3 Historique de Blockchain

La technologie Blockchain est l'une des plus grandes innovations du 21e siècle en raison de son impact considérable dans divers domaines. L'histoire de la blockchain remonte au début des années 1990 :

En 1991, les chercheurs Stuart Haber et W. Scott Stonetta ont jeté les bases de la technologie blockchain en développant une blockchain cryptographiquement sécurisée pour horodater des documents, empêchant ainsi toute modification ultérieure. En 1992, cette technologie a été améliorée avec l'intégration d'arbres Merkle, ce qui a permis de regrouper plusieurs documents en un seul bloc [27].

En 2004, Hal Finney, un informaticien et activiste de la crypto-monnaie, a développé un système appelé RPoW (Reusable Proofs of Work) qui a résolu le problème de la double dépense, offrant une vérification en temps réel de l'exactitude et de l'intégrité des transactions. En 2008, la première blockchain véritable a vu le jour avec la création de Bitcoin par un individu anonyme sous le pseudonyme de Satoshi Nakamoto. Bitcoin est basé sur un algorithme de preuve de travail (HashCash) et repose sur un réseau pair à pair décentralisé pour suivre et vérifier les transactions, éliminant ainsi la nécessité d'une autorité centrale [28].

En 2013, Vitalik Buterin a fondé Ethereum, une blockchain conçue pour être plus flexible que Bitcoin, capable d'enregistrer des contrats et des transactions plus complexes. Ethereum a introduit les contrats intelligents, des programmes autonomes exécutés sur sa blockchain grâce à la machine virtuelle Ethereum (EVM) [29].

Ces étapes historiques démontrent l'évolution de la technologie blockchain, de ses débuts en tant que système de sécurité pour l'horodatage des documents à sa transformation en une plateforme polyvalente pour la gestion de contrats et de transactions décentralisées.

2.4 Caractéristiques principales de la technologie Blockchain

En général, la technologie Blockchain est caractérisée par [5]:

- **Éliminer les intermédiaires** : La technologie blockchain permet d'échanger des données de manière sécurisée et transparente, sans intervention d'un tiers de confiance. Les transactions sont validées par un consensus entre les utilisateurs du réseau, qui vérifient mutuellement le travail de validation, d'où la Blockchain est décentralisée, donc la confiance est distribuée.
- **La transparence** : La transparence de la blockchain est due au fait que toutes les informations sont accessibles à tous les utilisateurs du réseau. Cela permet à chacun de vérifier la validité des données et de détecter d'éventuelles anomalies. Cependant, la transparence de la blockchain peut être remise en cause par l'anonymat des utilisateurs. En effet, les utilisateurs de la blockchain peuvent choisir de masquer leur identité, ce qui rend plus difficile l'identification des auteurs d'éventuelles activités frauduleuses.
- **La sécurité** : La blockchain est une technologie qui offre un haut niveau de sécurité. Cela est dû à son architecture décentralisée, qui rend difficile la modification ou la suppression des données. Les données sont stockées sur une multitude de serveurs répartis dans le monde entier. Cela rend impossible la destruction de toutes les copies des données, même en cas de cyberattaque ou de contrôle de l'État. En effet, pour modifier ou supprimer une donnée, il faudrait modifier ou supprimer toutes les copies de la donnée sur tous les serveurs du réseau. Cela est un processus extrêmement complexe et coûteux, qui est rarement réalisable.
- **L'autonomie** : La puissance de calcul et l'espace d'hébergement sont fournis par les nœuds du réseau, c'est-à-dire les utilisateurs eux-mêmes. Il n'y a donc pas besoin d'infrastructures centrales. Au sein d'une blockchain, l'infrastructure n'est plus concentrée dans les mains d'une organisation mais est, au contraire, éclatée dans l'ensemble des points du réseau.

2.5 Les modèles de la Blockchain

Il existe trois types de Blockchain : La blockchain publique ouverte à tous, la blockchain privée à accès et usage limités à un certain nombre d'acteurs, et la blockchain consortiums (hybrides).

a. Blockchain publique

Une blockchain publique est une blockchain que tout utilisateur peut rejoindre sans condition. En particulier, sur une blockchain comprenant des mineurs, n'importe quel utilisateur peut devenir mineur. Bitcoin et Ethereum sont des blockchains publiques. Les blockchains publiques disposent généralement d'un niveau de décentralisation plus élevé que les blockchains privées [24].

b. Blockchain privée

Ce type de Blockchain est considéré comme un réseau centralisé car entièrement contrôlé par une organisation. Dans une Blockchain privée, le régulateur confirme l'introduction de nouveaux membres et accorde des autorisations de lecture et d'écriture. L'agence peut être dirigée seule ou dirigée conjointement par différents participants. Son accès et son utilisation sont donc limités à certains acteurs. Nul ne peut y participer sans autorisation, mais chacun peut s'y référer [26].

c. Blockchain consortium

Une blockchain de consortium est un type de blockchain qui combine des éléments de blockchain privée et publique. Elle est contrôlée par un groupe restreint d'organisations qui ont convenu de partager des données sur un réseau décentralisé. Dans une blockchain de consortium, les procédures de consensus sont contrôlées par les nœuds prédéfinis du consortium [30].

2.6 Architecture de la blockchain

Nous commencerons par examiner comment la blockchain fonctionne en tant que couche dans un réseau pair à pair distribué. La couche la plus basse est la couche réseau, généralement Internet, et constitue la couche de communication de base pour toute blockchain. La blockchain peut être considérée comme une couche d'un réseau pair à pair distribué fonctionnant sur Internet, comme le montre le diagramme suivant. Ceci est similaire à SMTP, HTTP ou FTP fonctionnant sur TCP/IP [17]:

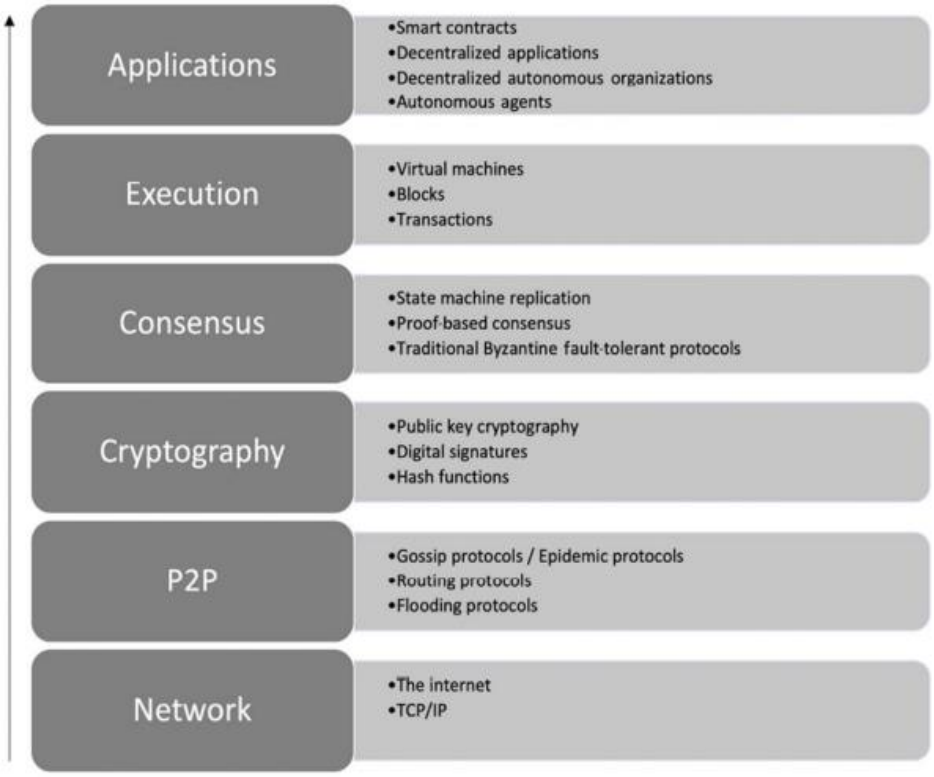


Figure 2-1 Architecture de la blockchain [17]

Nous allons maintenant aborder tous ces éléments un par un : La couche la plus basse est la couche réseau, qui est généralement Internet et fournit une couche de communication de base pour toute blockchain.

- Un réseau P2P (peer-to-peer) s'exécute au-dessus de la couche réseau, où chaque entité est à la fois client et serveur. Cela signifie que chaque nœud du réseau peut fournir des ressources aux autres nœuds, et peut également demander des ressources aux autres nœuds.
- La couche Cryptographie est essentielle à la sécurité de la blockchain. Elle utilise des protocoles cryptographiques pour garantir l'intégrité des processus, la diffusion sécurisée des informations et les mécanismes de consensus. Elle comprend la cryptographie à clé publique, les signatures numériques et les fonctions de hachage cryptographique.
- Vient ensuite la couche Consensus, qui implique l'utilisation de différents mécanismes de consensus pour garantir un consensus entre les différents participants à la blockchain. Il s'agit d'un autre élément important de l'architecture blockchain, qui comprend diverses techniques telles que le SMR (State Machine Replication), les mécanismes de consensus basés sur des preuves ou les protocoles de consensus byzantins traditionnels (dérivés de la recherche traditionnelle sur les systèmes distribués) tolérants aux pannes.
- La couche d'exécution de la blockchain fournit des services d'exécution, tels que le transfert de valeur, l'exécution de contrats intelligents et la génération de blocs. Elle est composée de machines virtuelles, de blocs, de transactions et de contrats intelligents. Les machines virtuelles telles que l'EVM, ewasm et Zinc VM permettent aux contrats intelligents d'être exécutés sur la blockchain.
- Enfin, la couche Applications de la blockchain est composée de contrats intelligents, d'applications décentralisées, de DAO (Decentralized Autonomous Organization) et d'agents autonomes. Elle permet aux utilisateurs d'interagir avec la blockchain via des applications décentralisées [17].

2.7 Fonctionnement

La blockchain est une base de données distribuée, partagée par un réseau d'ordinateurs. Les participants du réseau, appelés nœuds, stockent une copie identique de la base de données et valident les transactions. Pour ajouter une transaction à la blockchain, les nœuds la votent. Si la transaction est validée par une majorité des nœuds, elle est ajoutée à la base de données.
Les transactions sont regroupées dans des blocs, qui sont liés les uns aux autres par un hachage cryptographique. Cela permet de garantir que la blockchain est sécurisée et non modifiable. Les données associées aux transactions peuvent être de n'importe quel type. Le format des transactions est défini par le réseau sous-jacent, tandis que les données présentes dans celles-ci sont définies par les participants [26]. Les données de la blockchain peuvent être cryptées et signées numériquement pour garantir leur authenticité, leur intégrité et leur non-répudiation.

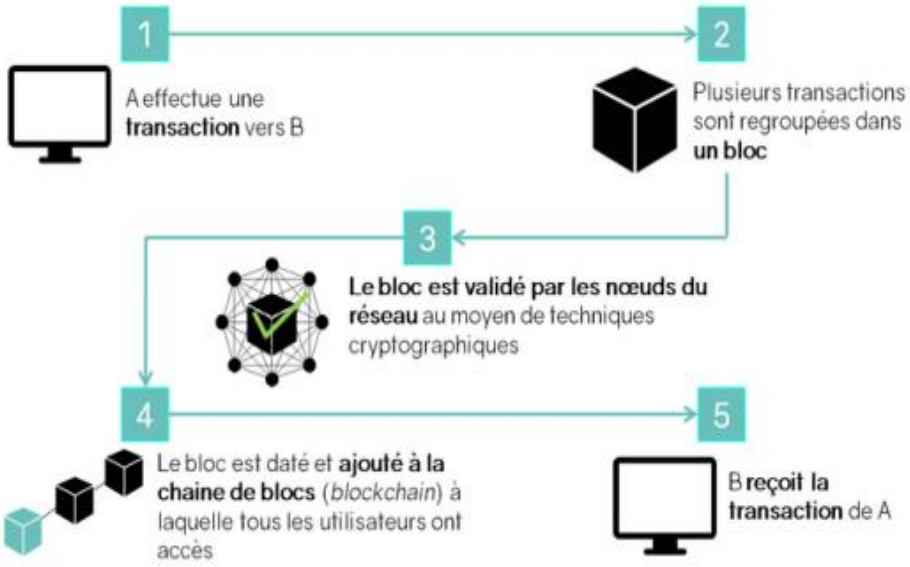


Figure 2-2 Fonctionnement de la Blockchain [31]

Le fonctionnement d'une transaction peut schématiquement être décrit en 5 étapes :

1. A effectue une transaction vers B.
2. Plusieurs transactions sont regroupées dans un bloc.

3. Le bloc est validé par les nœuds du réseau au moyen de techniques cryptographiques.
4. Quand le bloc est validé, il est daté et ajouté à la chaîne de blocs à laquelle tous les utilisateurs ont accès.
5. . B reçoit la transaction de A

2.8 L'échange pair à pair (p2p)

Un réseau pair à pair (P2P) est un réseau informatique dans lequel les ordinateurs sont connectés entre eux de manière égale. Il n'y a pas d'ordinateur central qui contrôle le réseau. Les ordinateurs du réseau partagent des ressources entre eux, telles que des fichiers, des données ou des services [7].

a. Rôle du P2P dans la blockchain

L`architecture paire à paire de la blockchain permet à toutes les crypto-monnaies d`être transférées dans le monde entier, sans avoir besoin d'intermédiaire ou de serveur central (rappelons ici que la blockchain est une technologie qui peut être utilisée pour d'autres applications que les crypto-monnaies). La blockchain permet le suivi décentralisé d'un ou plusieurs actifs numériques sur un réseau pair à pair. Lorsque nous disons un réseau pair à pair, cela signifie un réseau pair à pair décentralisé où tous les ordinateurs sont connectés d'une manière ou d'une autre, et où chacun conserve une copie complète du registre et le compare à d`autres appareils pour garantir que les données sont exactes. Ceci est différent d'une banque, où les transactions sont stockées de manière privée et ne sont gérées que par la banque [7].

b. La diffusion des blocs sur un réseau pair à pair

Chaque bloc est validé par un certain nombre d'utilisateurs appelés « mineurs » et transmis aux « nœuds » du réseau, c'est-à-dire aux détenteurs d'un registre, qui est la chaîne de blocs. Ceci est constamment mis à jour. Dans les blockchains dites ouvertes (*permissionless*), comme le bitcoin, tout internaute peut ainsi devenir un nœud de réseau en téléchargeant le

registre depuis un nœud existant. Chaque nœud est connecté à plusieurs autres, appelés pairs, eux-mêmes ayant leurs propres pairs, ce qui forme un réseau pair à pair [32].

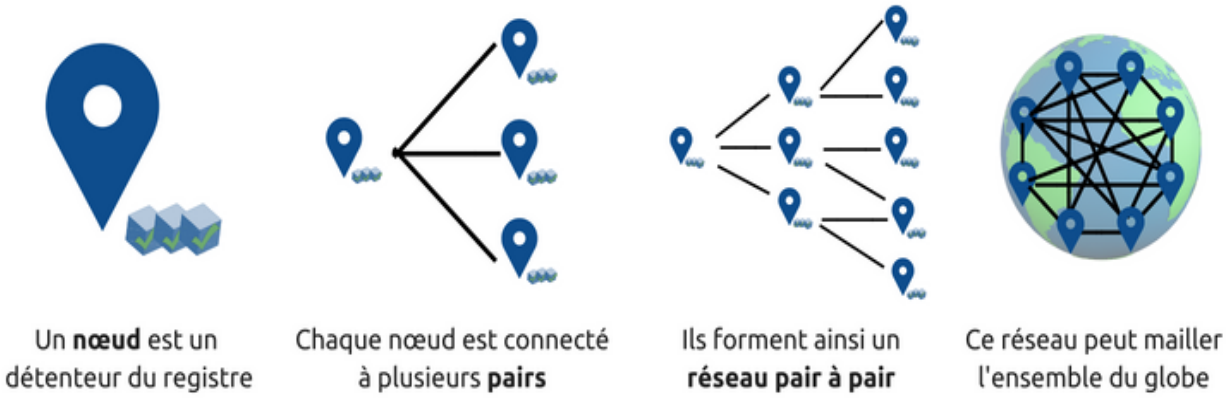


Figure 2-3 – Réseau pair à pair [32]

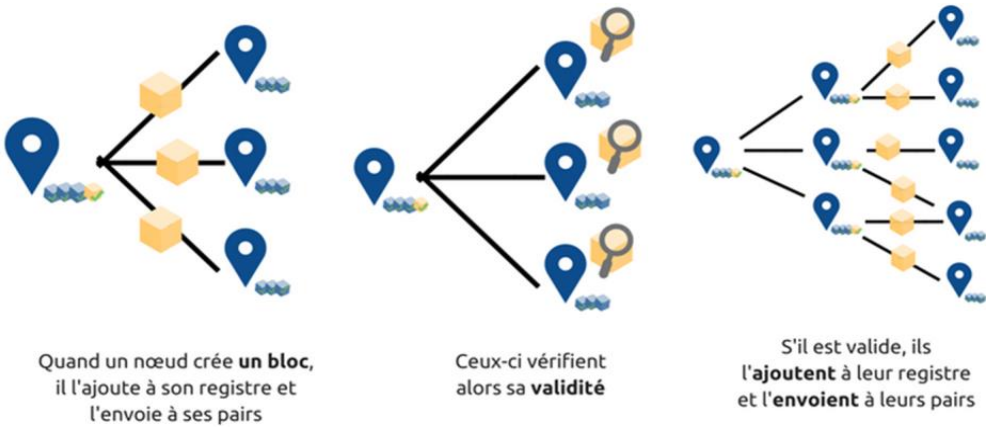


Figure 2-4 Diffusion d'un bloc dans le réseau [32]

Un nœud qui tente d'introduire un bloc invalide dans le réseau aura peu de chances de réussir. Les autres nœuds vérifient l'authenticité des blocs avant de les ajouter à leur registre. Par conséquent, un bloc invalide est peu susceptible d'être transmis à d'autres nœuds.

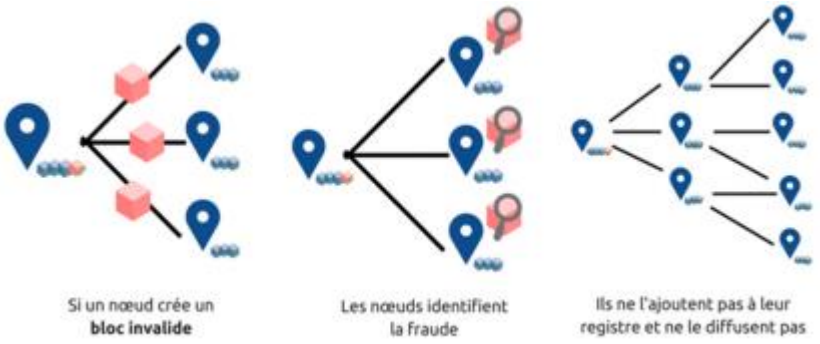


Figure 2-5 Introduction d'un bloc invalide [32]

La validation des blocs est un processus décentralisé qui garantit l'intégrité de la blockchain. Les utilisateurs du réseau participent à la validation en vérifiant l'authenticité des blocs. Cette sécurité, source de confiance, est l'un des aspects essentiels de la blockchain. La distribution des copies de la blockchain et la vérification par les utilisateurs rendent la modification de la blockchain très difficile, voire impossible.

c. Mécanismes de consensus

Dans les paiements traditionnels, les transactions sont centralisées et contrôlées par des tiers, tels que les banques. Dans la blockchain, les transactions sont décentralisées et vérifiées par tous les participants au réseau. Il est important que tout le monde ait la même copie. Ce système est plus transparent et sécurisé, car il n'y a pas de tiers qui peuvent contrôler ou modifier les données. Pour parvenir à ce consensus cohérent à l'échelle du système, un consensus mécanisme, soit une preuve de travail ou une preuve de participation est nécessaire [7].

Preuve de travail (PoW)

La Preuve de travail est le premier algorithme de consensus créé pour faire respecter le protocole crypto de la blockchain. Utilisé par Bitcoin, il nécessite une grande puissance de calcul, car il repose sur une fonction de hachage. Elle va calculer une empreinte unique nécessaire à la validation du bloc à partir des données fournies. Ce travail de hachage récompense les mineurs qui fournissent la puissance de calcul [33].

Preuves d'enjeu (PoS)

La preuve d'enjeu est un mécanisme de consensus alternatif à la preuve de travail. Au lieu de se fonder sur la puissance de calcul, la preuve d'enjeu se fonde sur la possession de la cryptomonnaie du réseau. Dans un système preuve d'enjeu, les nœuds qui valident les blocs sont sélectionnés au hasard en fonction de la quantité de cryptomonnaie qu'ils possèdent. Plus un nœud possède de cryptomonnaie, plus il a de chances d'être sélectionné.

La preuve d'enjeu présente plusieurs avantages par rapport à la Preuve de travail. Elle est plus économe en énergie, car elle ne nécessite pas de puissance de calcul importante. Elle est également plus rapide, car les blocs peuvent être validés plus rapidement [34].

Preuve d'autorité (PoA)

La preuve d'autorité est un algorithme de consensus qui permet de valider les transactions sur un réseau blockchain. Dans un système preuve d'autorité, un nombre restreint de validateurs sont désignés pour valider les blocs. Ces validateurs sont généralement des entités de confiance, telles que des entreprises ou des institutions publiques. Les blocs sont validés sans vérification, à l'unanimité ou à la majorité. Cela signifie que tous les validateurs doivent approuver le bloc, ou qu'une majorité d'entre eux doit l'approuver [35].

2.9 Le principe de la décentralisation

On ne pouvait pas finir ce chapitre sans parler de la décentralisation qui est une caractéristique essentielle de la blockchain.

On parle d'un réseau centralisé lorsque celui-ci est contrôlé par une autorité centralisée. Dans un réseau centralisé, l'architecture est construite autour d'un serveur unique où tous les processus importants sont exécutés. [36].

Plutôt que de s'appuyer sur un seul serveur central géré par une autorité centralisée, un réseau décentralisé répartit les tâches de traitement de l'information sur plusieurs appareils souvent appelés "nœuds". Ainsi, même si l'un des nœuds tombe en panne ou est attaqué, les serveurs restants peuvent continuer à accorder l'accès aux données aux utilisateurs [36]

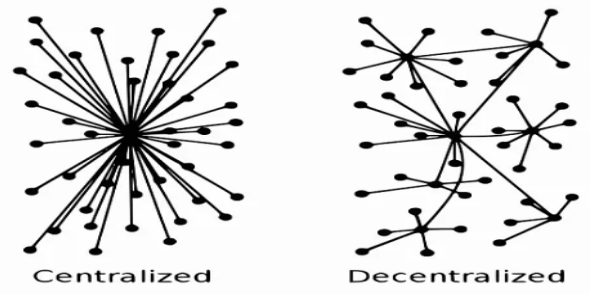


Figure 2-6 Système centralisée et décentralisé [37]

2.10 Conclusion Partielle

Ce chapitre introductif nous a permis d'acquérir une compréhension des concepts fondamentaux de notre sujet. Nous avons saisi que la blockchain offre la possibilité d'améliorer la transparence et la fiabilité des transactions. Dans le chapitre suivant, nous nous lancerons dans la phase de conception et de modélisation de notre système, où nous identifierons les différentes fonctionnalités de notre application.

Chapitre 3 Conception du Système

3.1 Introduction

Après avoir présenté la blockchain dans le chapitre précédent, nous allons à présent nous plonger dans la conception de notre application qui repose sur cette technologie. Nous commencerons par présenter un projet similaire, puis nous exposerons l'architecture générale de notre système. Enfin, nous définirons les besoins fonctionnels de notre système.

3.2 Projet Similaire

Le *Massachusetts Institute of Technology* (MIT) utilise la blockchain pour certifier les diplômes obtenus par les étudiants. C'est une université privée ancrée dans la recherche et située à Cambridge, dans le Massachusetts, aux États-Unis. Elle a été fondée en 1861 et est considérée comme l'une des meilleures universités du monde [38].

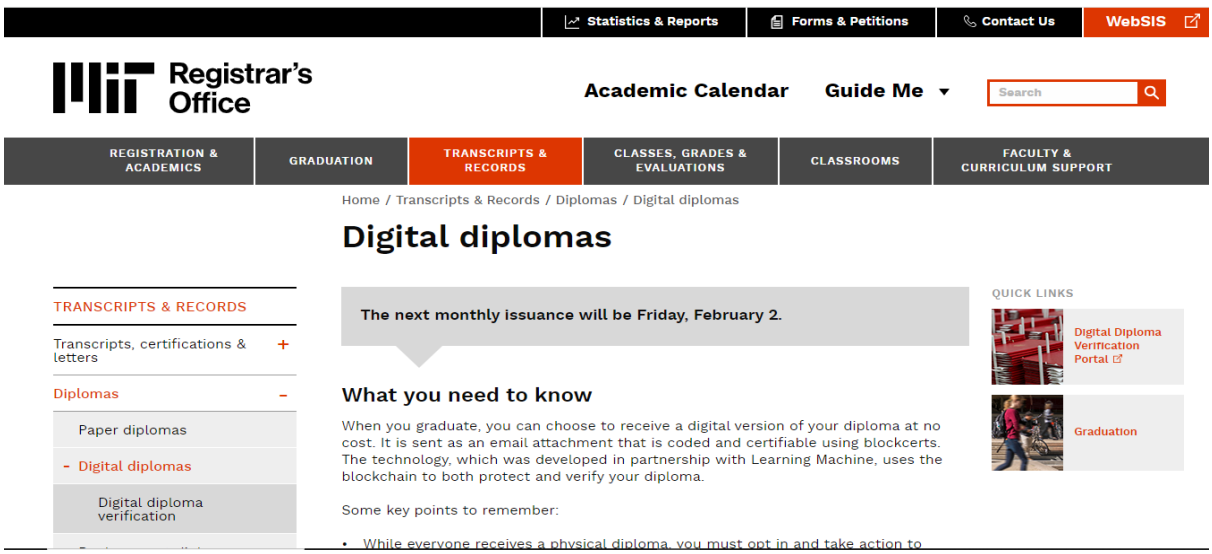


Figure 3-1 Page du site de l'MIT expliquant l'accès au diplôme numérique [38]

Le MIT utilise la blockchain pour délivrer des diplômes numériques afin de garantir l'authenticité et la sécurité de ces diplômes. La Figure 3-1 donne une vue de la page d'accueil au service des diplômes numériques. La blockchain est une technologie de stockage et de transmission d'informations qui est décentralisée, ce qui signifie qu'elle n'est pas contrôlée par un organe central. Cela signifie que les diplômes numériques du MIT sont enregistrés sur un réseau des serveurs répartis dans le monde entier. Lorsque le MIT délivre un diplôme numérique, il crée un jeton numérique qui contient les informations du diplôme, telles que le nom du diplômé, le nom du diplôme, la date de remise du diplôme, etc. Ce jeton numérique est ensuite enregistré sur la blockchain. Chaque fois qu'un diplôme numérique est consulté en utilisant l'interface de la Figure 3-2, le jeton numérique est vérifié sur la blockchain. Cela permet de s'assurer que le diplôme est authentique et qu'il n'a pas été falsifié [39].

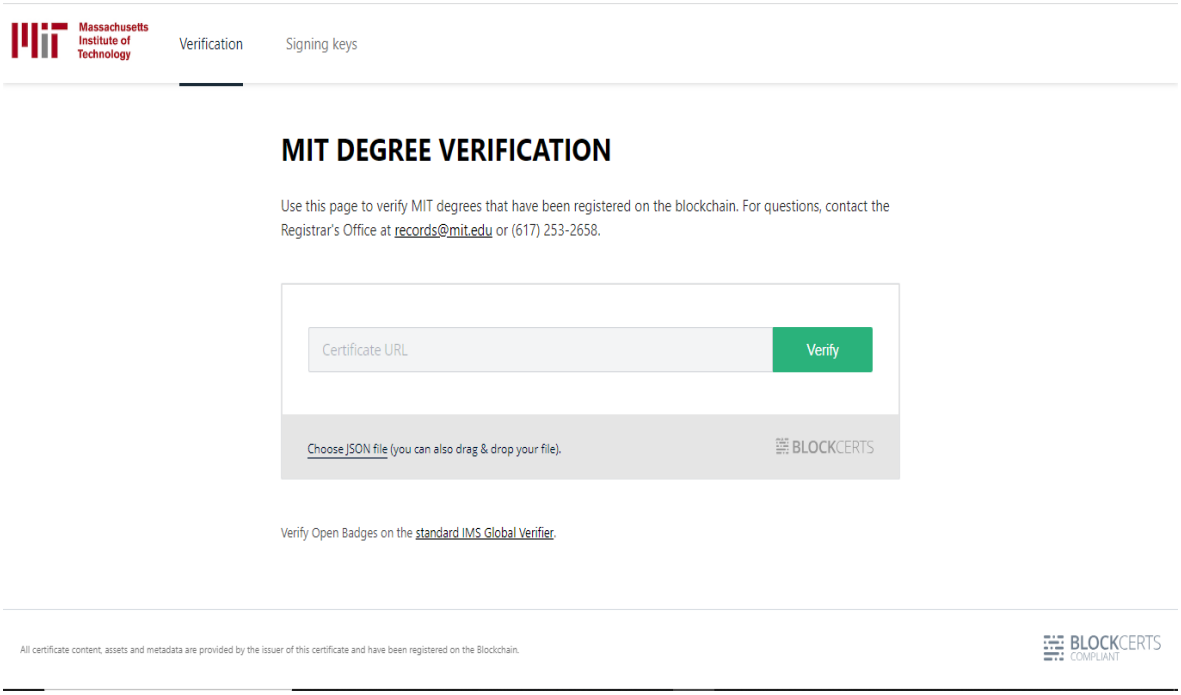


Figure 3-2 Page du site de l'MIT pour la vérification de l'url de diplôme [39]

Dans les sections à venir, nous illustrerons comment concevoir et mettre en œuvre un tel système pour son utilisation au sein d'une université en République Démocratique du Congo (RDC).

3.3 Analyse détaillée des besoins

3.3.1 Architecture globale

a. Clarification de l'idée

La définition de la portée et du but d'une application Blockchain est une étape essentielle de son développement. Les étapes de développement d'une application Blockchain sont les suivantes :

- Avant de développer une application Blockchain, il est important de comprendre pourquoi on va utiliser cette technologie :

  La blockchain offre plusieurs avantages, notamment :

  - Sécurité : La blockchain est une technologie très sécurisée, car il est très difficile de falsifier ou de modifier les informations qui y sont enregistrées. Cela est important pour la gestion des diplômes, car ces derniers sont une preuve de qualification et de compétence.

  - Transparence : La blockchain offre une transparence totale, car les informations qui y sont enregistrées sont accessibles à tous. Cela permet aux étudiants, aux employeurs et aux autres parties prenantes de vérifier l'authenticité des diplômes.

  - Efficacité : La blockchain peut améliorer l'efficacité des processus, car elle automatise les transactions et les opérations. Cela peut simplifier et accélérer le processus d'obtention d'un diplôme.

- La deuxième question est la suivante : La blockchain est-elle la solution appropriée pour ce projet ? L'arbre de décision de la Figure 3-3 répond à toutes les questions sur la nécessité d'une blockchain [26].

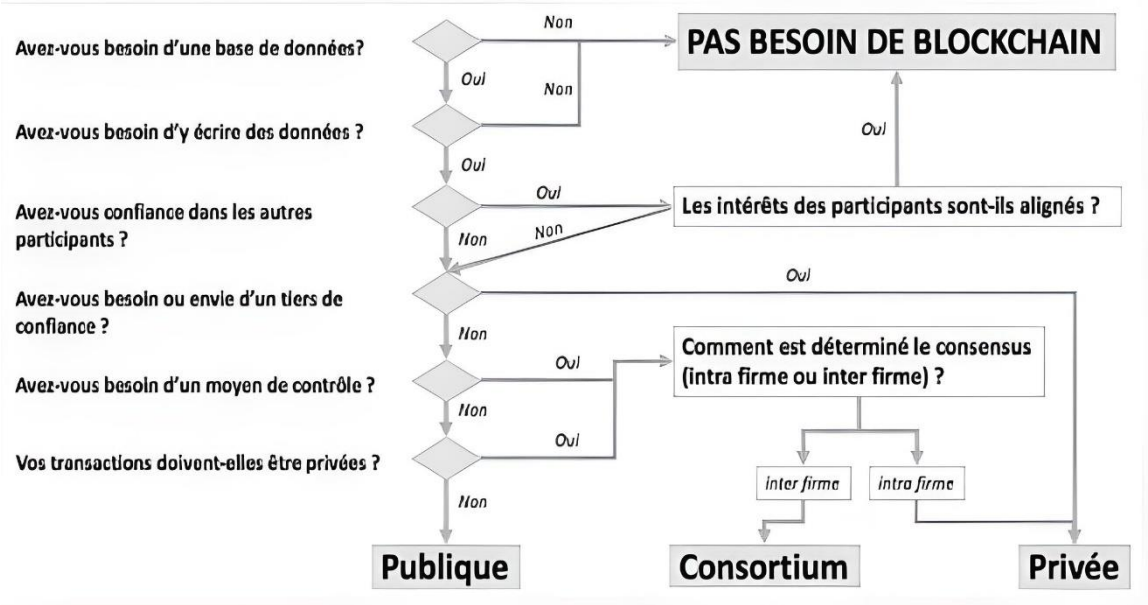


Figure 3-3 Arbre de décision [26]

L'arbre de décision permet de déterminer si la blockchain est une solution appropriée et si quel type de blockchain est adapté pour un cas d'utilisation donné. Pour ce faire, nous allons poser six questions sur les besoins du cas d'utilisation :

- Avons-nous besoin d'une base de données ?
- Avons-nous besoin d'y écrire des données ?
- Avons-nous confiance dans les autres participants ?
- Avons-nous besoin ou envie d'un tiers de confiance ?
- Avons-nous besoin d'un moyen de contrôle ?
- Nos transactions doivent-elle être privées ?

En répondant à toutes ces questions, pour notre cas, nous avons besoin d'une base de données où nous stockerons les informations des universités et des diplômes. Ensuite, nous avons besoin de la blockchain car les informations doivent être partagées entre plusieurs participants, dont certains ne sont pas dignes de confiance. Finalement, on n'a pas besoin d'un moyen de contrôle centralisé, et nous allons nous orienter vers une Blockchain publique.

b. Identification des acteurs du système

Apres analyse du système, nous sommes arrivés à la conclusion qu'il serait nécessaire de distinguer 4 différents acteurs qui sont :

- **Vérificateur** : Ce sont les entreprises, les institutions d'enseignements et toutes autres personnes disposant d'un hash de certificat à vérifier.
- **Université** : C'est l'entité qui va générer les diplômes numériques à partir du système et les envoyer aux finalistes. Aussi cette entité peut révoquer les diplômes.
- **Admin (Authentic)** : L'entité chargée de concevoir les modèles de diplômes pour les universités et de valider leur conformité avec les palmarès du ministère de l'Enseignement supérieur.
- **Etudiant** : C'est l'entité qui doit recevoir le diplôme et le faire accepter par l'employeur.

c. Architecture globale

La figure suivante présente une vue d'ensemble de l'architecture de notre système :

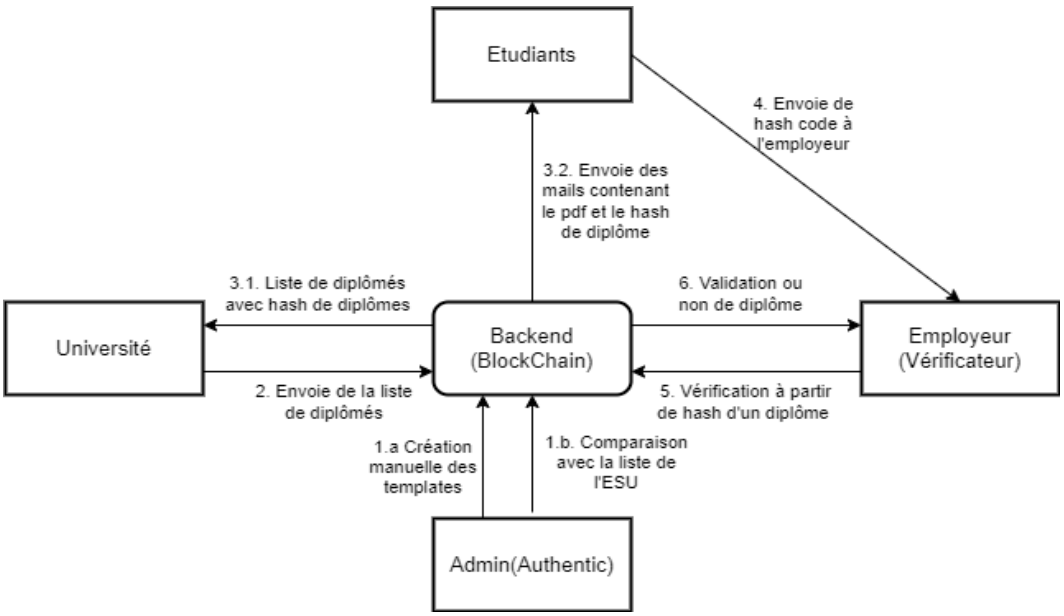


Figure 3-4 Architecture globale du système

Interprétation de l'architecture globale

1.a. *Création manuelle des Templates* : L'administration du système doit, à partir des exemplaires et des informations fournies par l'université, implémenter, moyennant les codes, un Template qui servira de certificat numérique.

1.b. *Comparaison avec la liste de l'ESU* : L'admin doit insérer la liste de finalistes reconnues par l'ESU dans le système, et la comparer avec la liste introduite par l'université. On doit s'assurer que les deux listes soient les mêmes. Rappelons que cette fonctionnalité ne sera pas implémentée dans le système développé dans le cadre de ce travail.

2. *Envoie de la liste de diplômés* : L'université doit insérer la liste de diplômés dans le système afin que le système enregistre les informations des diplômes sur la blockchain.

3.1. Après enregistrement des informations sur blockchain, on affichera chaque diplôme auquel est associé un hash.

3.2. On enverra à chaque étudiants finaliste un fichier PDF de diplôme et un hash associé à ce diplôme

4. Lorsque l'étudiant veut postuler à un emploie, il va tout simplement envoyer le hash à l'employeur pour qu'il vérifie l'authenticité du diplôme.

5. L'employeur va entrer le hash code sur la page d'accueil du système.

6. Le système va vérifier la validité de diplôme, à partir du hash : Il va d'abord vérifier si le hash existe dans la blockchain et ensuite vérifie si le diplôme n'est pas dans la liste des diplômes révoqués.

3.3.2 Conception et Analyse

Parmi les treize diagrammes officiels de UML 2.0, nous présenterons trois d'entre eux pour permettre une meilleure compréhension du fonctionnement du système : le diagramme de cas d'utilisation, le diagramme de séquence (tous deux faisant partie des diagrammes de comportement) et le diagramme de classe (qui appartient aux diagrammes structurels). Ces diagrammes nous permettront de visualiser les interactions entre les acteurs et le système, ainsi que de comprendre la structure globale du système en mettant en évidence les composants clés qui le constituent. Avant de présenter ces diagrammes, commençons tout

d'abord par présenter les spécifications du système qui reprennent de manière structurée et claire les besoins fonctionnels du système.

a. Modèle de Spécifications

Le modèle de spécifications est un ensemble de phrases bien formées (respectant une certaine formulation) et numérotées où chaque phrase est appelée spécification [40]. Le tableau 3-1 reprend les principales spécifications retenues.

Tableau 3-1 Modèle de spécification du système

Id	Spécification	Etat	Criticisme	Effort	Stabilité	Cible
1	Le système doit permettre à l'université de s'authentifier	Approuvé	Important	4 jours	Stable	1
2	Le système doit permettre à l'université de générer des diplômes numériques à partir de liste de diplômés	Approuvé	Critique	12 jours	Stable	1
4	Le système doit permettre à université d'envoyer par mail le hash et le PDF de diplôme au finaliste	Approuvé	Critique	7 jours	Stable	1
5	Le système doit permettre aux entreprises et autres universités de vérifier l'authenticité de diplôme	Approuvé	Critique	5 jours	Stable	1
6	Le système doit permettre de comparer la liste de diplômé de l'ESU et l'université avant de générer le diplôme	Proposé	Utile	7 jours	Instable	2
7	Le système doit permettre à l'admin du système de créer manuellement de Template de diplôme	Approuvé	Critique	10 jours	Stable	1

b. Diagramme de cas d'utilisation

Un cas d'utilisation représente un ensemble de séquences d'actions à réaliser par le système et produisant un résultat observable intéressant pour un acteur particulier représenté par des ellipses et limité par un rectangle pour représenter le système [41]. C'est avec ces différents diagrammes des cas d'utilisation que nous représentons les besoins fonctionnels de l'application.

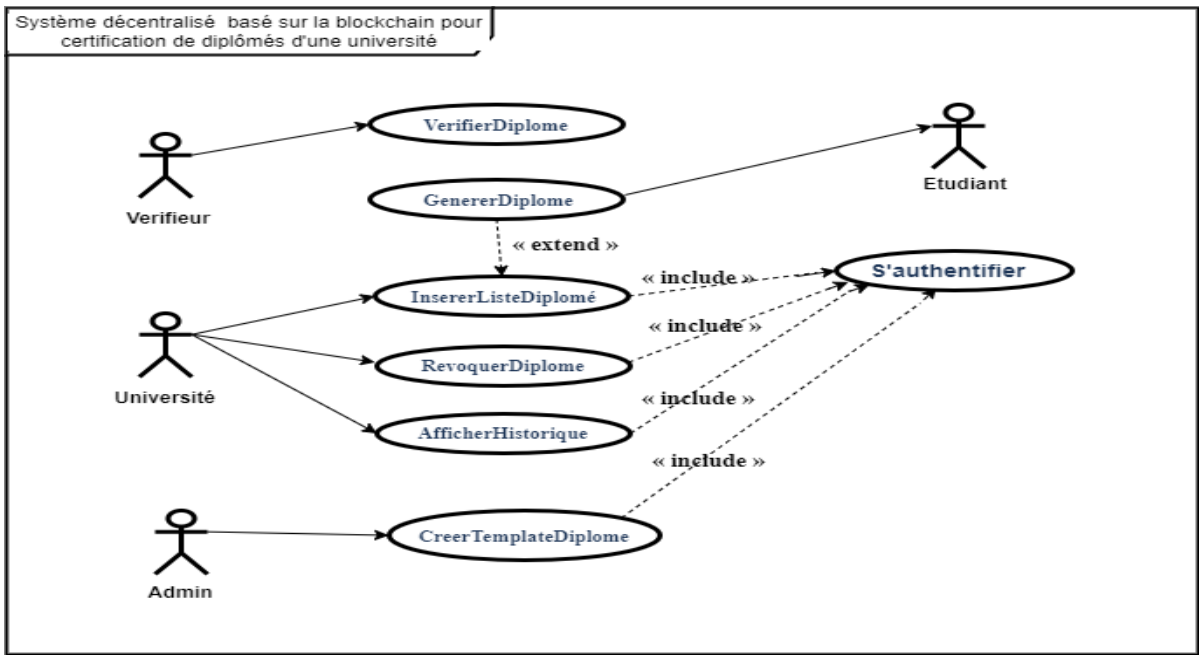


Figure 3-5 Diagramme des cas d'utilisations du système

Documentation des cas d'utilisations

Tableau 3-2 Vérification de diplôme à partir de hash

CU : VerifierDiplôme
ID : 1
Description brève : Vérification de diplôme à partir de hash
Acteurs primaires : Vérificateur
Acteurs secondaires : Aucun
Préconditions :
Enchaînement principal : 1. Le CU commence lorsque le vérifieur arrive sur la page d'accueil 2. Lorsqu'il clique sur « Vérifier diplôme » 3. Le vérifieur entre le hash code dans un popup 4. Le système vérifie si le hash existe et code n'est pas dans la liste de diplôme révoquer 5. Le système le dirige vers sa page de résultat de vérification 5.1. Le système affiche le résultat négatif si le diplôme n'est pas authentique

5.2. Le système affiche le résultat positif si le diplôme est authentique 6. Le Vérifieur revient à l'étape 1 de l'enchaînement principal
Postconditions : aucune
Enchaînements alternatifs :

Tableau 3-3 Authentification de l'université

CU : S'authentifier
ID : 2
Description brève : Authentification de l'université
Acteurs primaires : Université, Admin
Acteurs secondaires : Aucun
Préconditions :
Enchaînement principal : 1. Le CU commence lorsque l'utilisateur clique sur « se connecter » 2. L'utilisateur entre son nom de compte et son mot de passe 3. Le système valide le nom de compte et le mot de passe 4. Le système le dirige vers sa page par défaut
Postconditions : Utilisateur connecté »
Enchaînements alternatifs : CompteInvalide 1. L'utilisateur sera redirigé vers le point 2 de l'enchaînement principale MotdePasseInvalide 1**.** L'utilisateur sera redirigé vers le point 2 de l'enchaînement principale

Tableau 3-4 Insertion la liste de diplômé

CU : InsererListeDiplôme
ID : 3
Description brève : Insertion la liste de diplômé
Acteurs primaires : Université

Acteurs secondaires : Aucun
Préconditions : L'université doit être authentifier
Enchaînement principal : 1. Le CU commence lorsque l'université clique sur « InsererListe » 2. L'université récupère la liste Excel 3. L'université charge la liste dans le système
Postconditions : Fichier chargé
Enchaînements alternatifs : FichierInvalide 1. L'utilisateur sera redirigé vers le point 2 de l'enchaînement principale Annulation 1. L'utilisateur sera redirigé vers le point 1 de l'enchaînement principale

Tableau 3-5 L'université génère les diplômes

CU : GénérerDiplôme
ID : 4
Description brève : L'université génère les diplômes
Acteurs primaires : Université
Acteurs secondaires : Etudiant
Préconditions : L'université doit être authentifier ; La liste de diplômé doit être insérer
Enchaînement principal : 1. Le CU commence lorsque l'université clique sur « Générer diplôme » 2. Le système produit un hash de chaque diplôme 3. Le système envoie à chaque diplômé un mail contenant un hash et le PDF de diplôme
Postconditions : Diplômes générés »
Enchaînements alternatifs : ErreurTransaction 1. L'utilisateur sera redirigé vers le point 2 de l'enchaînement principale

Tableau 3-6 Révocation d'un diplôme

CU : RévoquerDiplôme
ID : 5
Description brève : Révocation d'un diplôme
Acteurs primaires : Université
Acteurs secondaires : Aucun
Préconditions : L'université doit être authentifier ; Le diplôme doit avoir été générer
Enchaînement principal : 1. Le CU commence lorsque l'utilisateur clique sur « RevoquerDiplome » 3. L'université sélectionne le diplôme à révoquer 4. L'université clique sur « Révoquer » 5. Le système ajoute le hash dans la liste de révoqué
Postconditions : Diplôme révoquer »
Enchaînements alternatifs :

Tableau 3-7 Afficher l'historique des diplômes déjà générer et déjà révoquer

CU : AfficherHistorique
ID : 6
Description brève : Afficher l'historique des diplômes déjà générer et déjà révoquer
Acteurs primaires : Université
Acteurs secondaires : Aucun
Préconditions : L'université doit être authentifier
Enchaînement principal : 1. Le CU commence lorsque l'utilisateur clique sur « afficherHistorique » 2. Il peut sélectionner soit : 2.1 Voir Diplôme Générer 2.2 Voir Diplôme Révoquer
Postconditions :

Enchaînements alternatifs :

Tableau 3-8 Consultation de diplôme

CU : ConsulterDiplôme
ID : 7
Description brève : Consultation de diplôme
Acteurs primaires : Utilisateur
Acteurs secondaires : Aucun
Préconditions : Le diplôme doit avoir été généré ; Recevoir un mail contenant le diplôme
Enchaînement principal : 1. Le CU commence lorsque l'utilisateur clique sur « ConsulterDiplome »
Postconditions : L'étudiant voit les détails de diplôme »
Enchaînements alternatifs :

Tableau 3-9 Création de Template d'un diplôme

CU : CréerTemplatediplôme
ID : 8
Description brève : Création de Template d'un diplôme
Acteurs primaires : Admin
Acteurs secondaires : Aucun
Préconditions :
Enchaînement principal : 1. Le CU commence lorsque l'université envoie par mail les détails de leurs diplômes 2. L'admin dessine à partir de code le diplôme en question
Postconditions : Template créer »
Enchaînements alternatifs :

c. Diagramme de classe

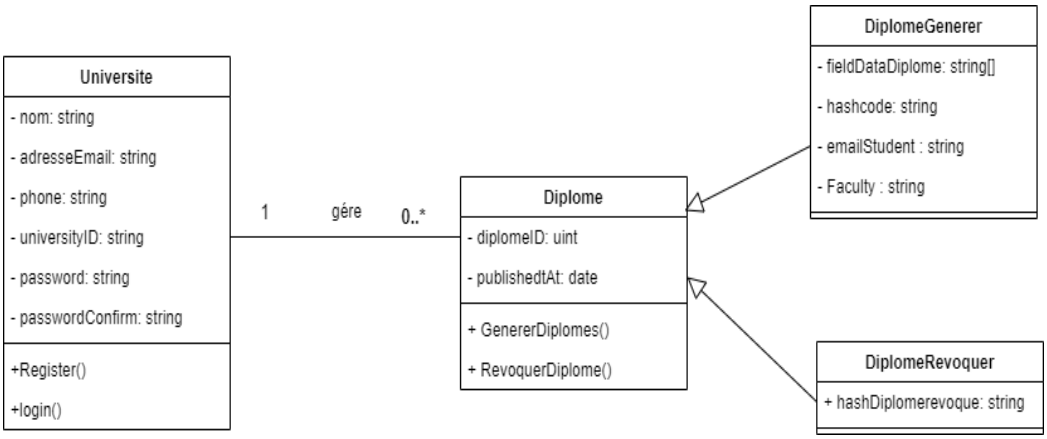


Figure 3-6 Diagramme de classe du système

Interprétation du diagramme des classes

Le diagramme des classes du système étudié est basé sur les règles suivantes :

- Une université peut gère zéro ou plusieurs diplômes
- Un diplôme peut être gérer par une seule université
- Les classes « DiplomeGenerer » et « DiplomeRevoquer » hérite de la classe « Diplôme »

d. Diagramme de séquences

Un diagramme de séquence est un diagramme UML qui représente la séquence des interactions entre les objets d'un système. Il permet de visualiser la façon dont les objets communiquent entre eux pour réaliser une tâche donnée [41]. Par la suite nous présenterons les diagrammes de séquences de quelques cas d'utilisations de notre système. Notamment le cas d'utilisation « S'authentifier » qui correspond au diagramme de séquence de la figure 3-7, « VérifierDiplôme » qui correspond au diagramme de séquence de la figure 3-8 , « AfficherHistorique» qui correspond au diagramme de séquence de la figure 3-9, « RévoquerDiplôme » qui correspond au diagramme de séquence de la figure 3-10 et « GénérerDiplôme » qui correspond au diagramme de séquence de la figure 3-11.

Diagramme de séquence pour l'Authentification

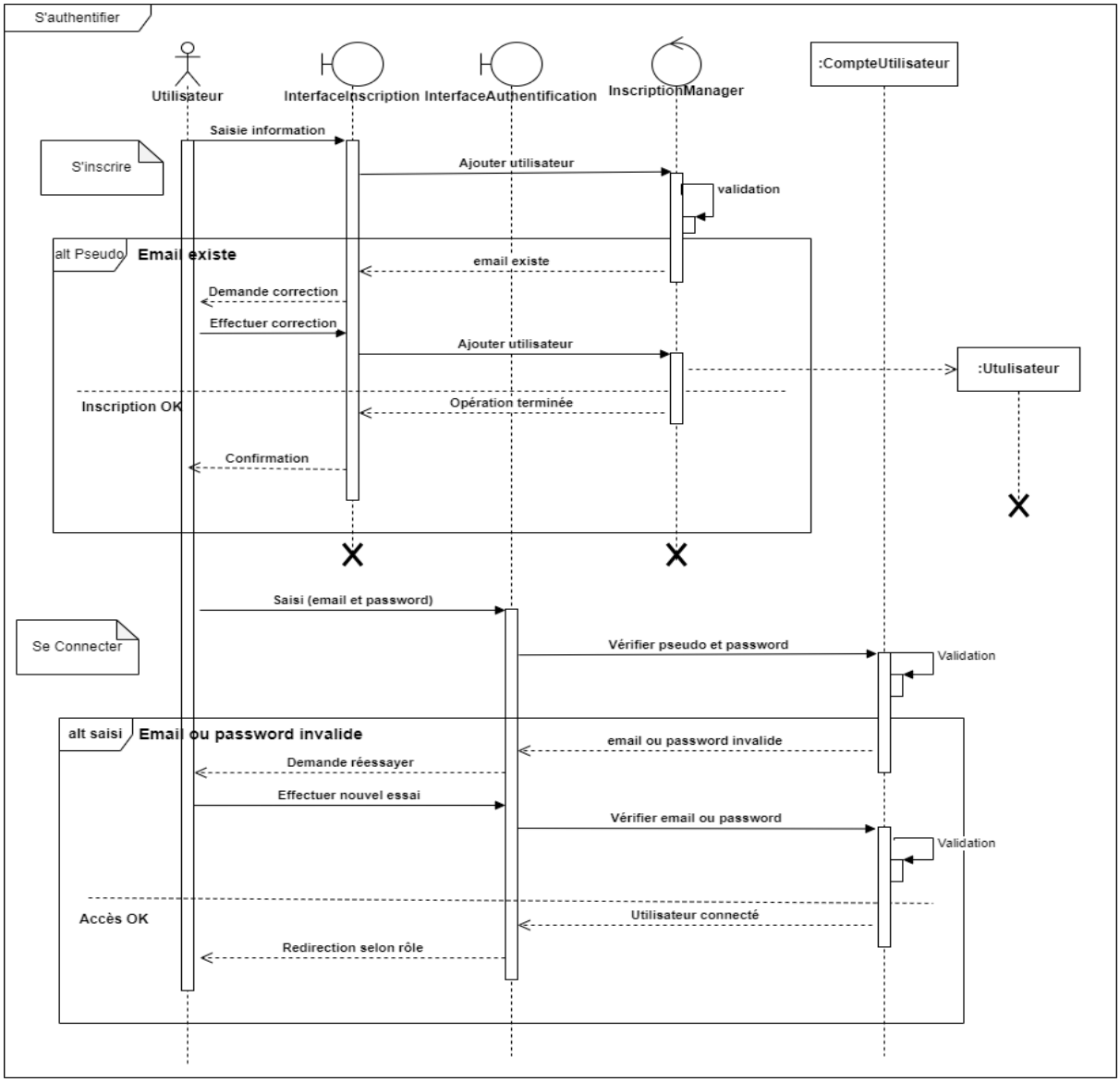


Figure 3-7 Diagramme de séquence pour l'authentification

Diagramme de séquence pour la Vérification de Diplôme

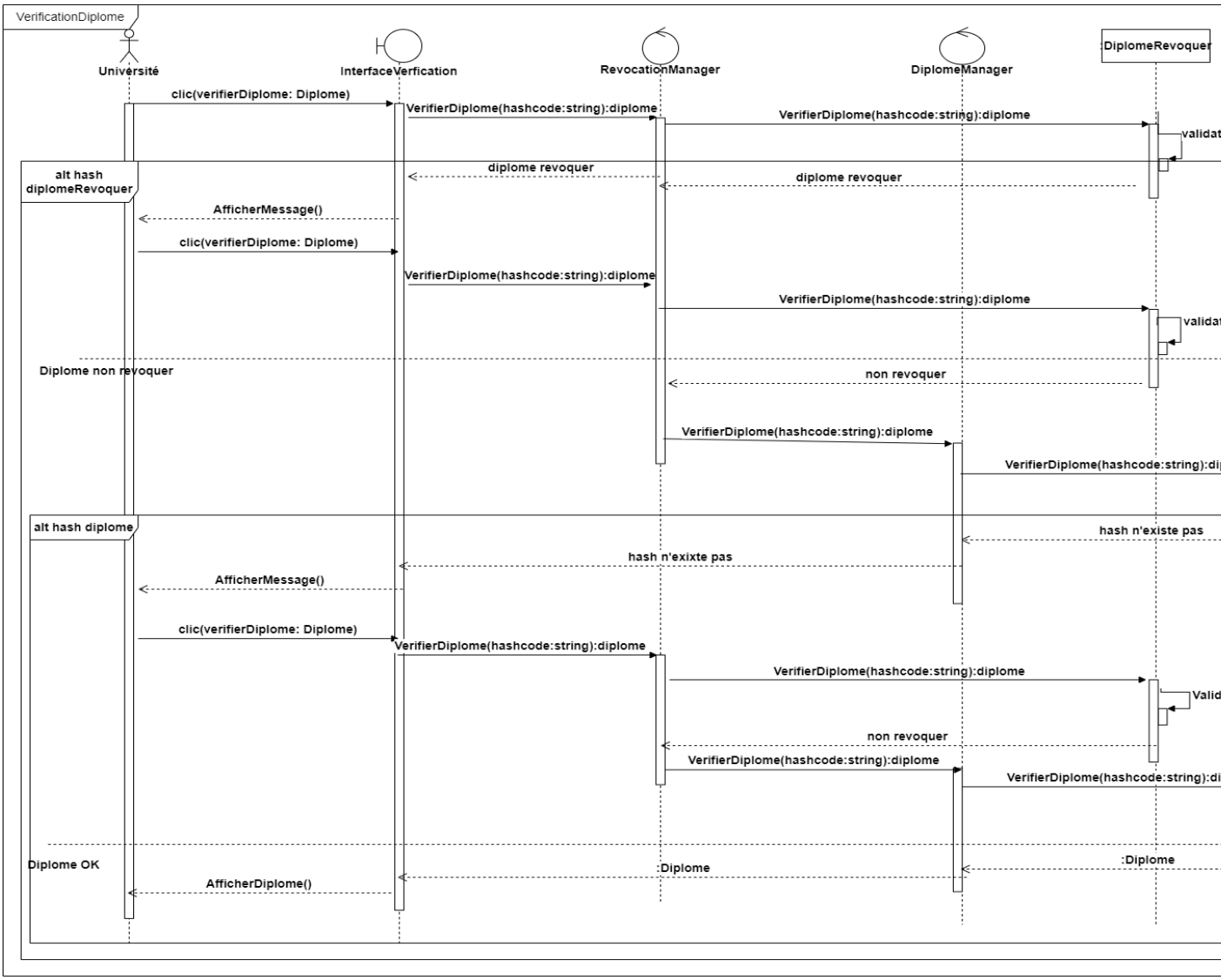


Figure 3-8 Diagramme de séquence pour la vérification de diplôme

Diagramme de séquence pour l'affichage de l'historique

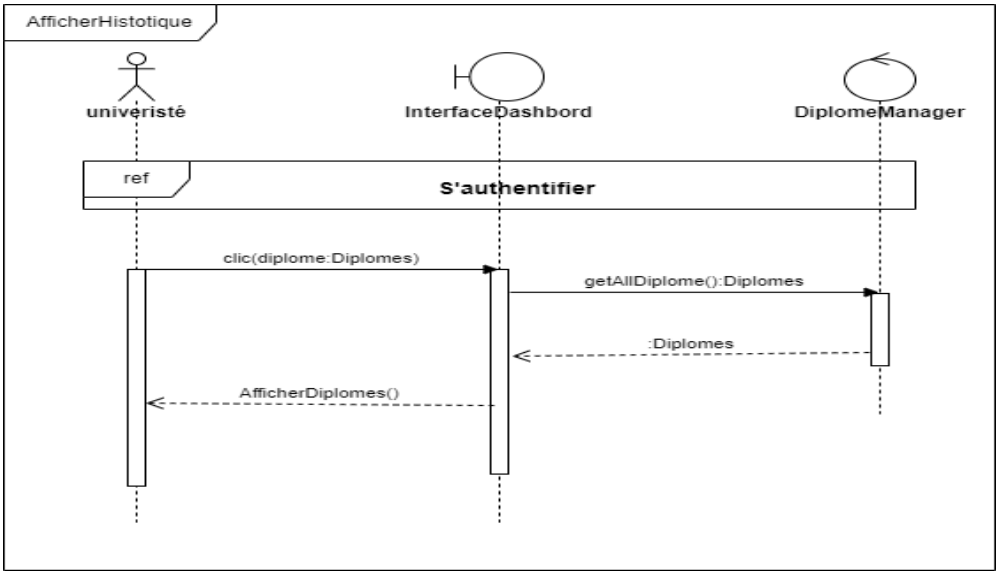


Figure 3-9 Diagramme de séquence pour l'affichage des diplômes

Diagramme de séquence pour la révocation d'un diplôme

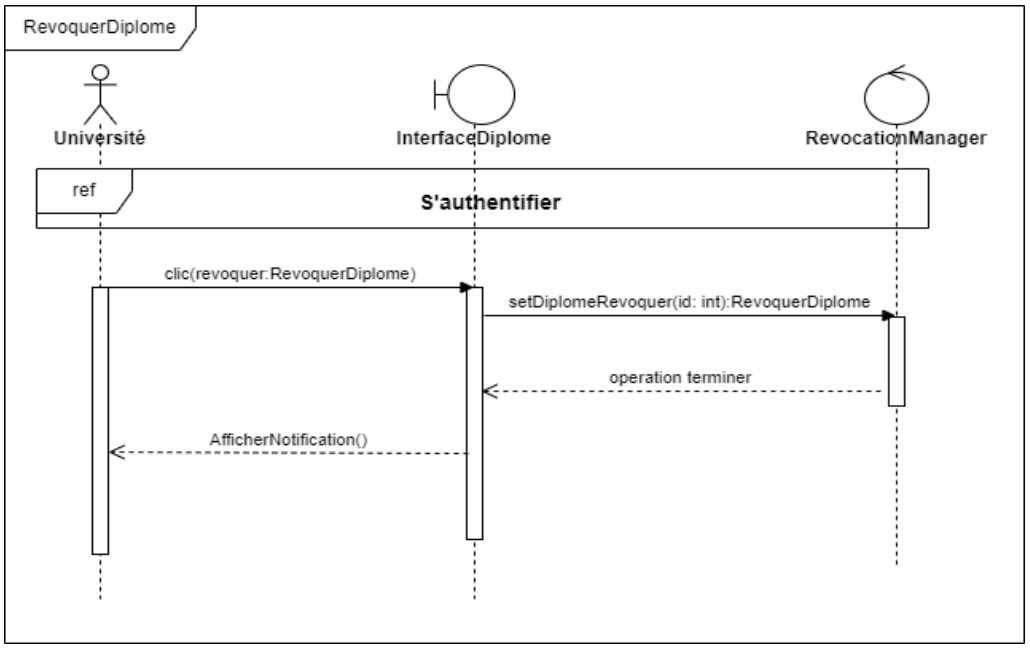


Figure 3-10 Diagramme de séquence pour la révocation d'un diplôme

Diagramme de séquence pour générer un diplôme

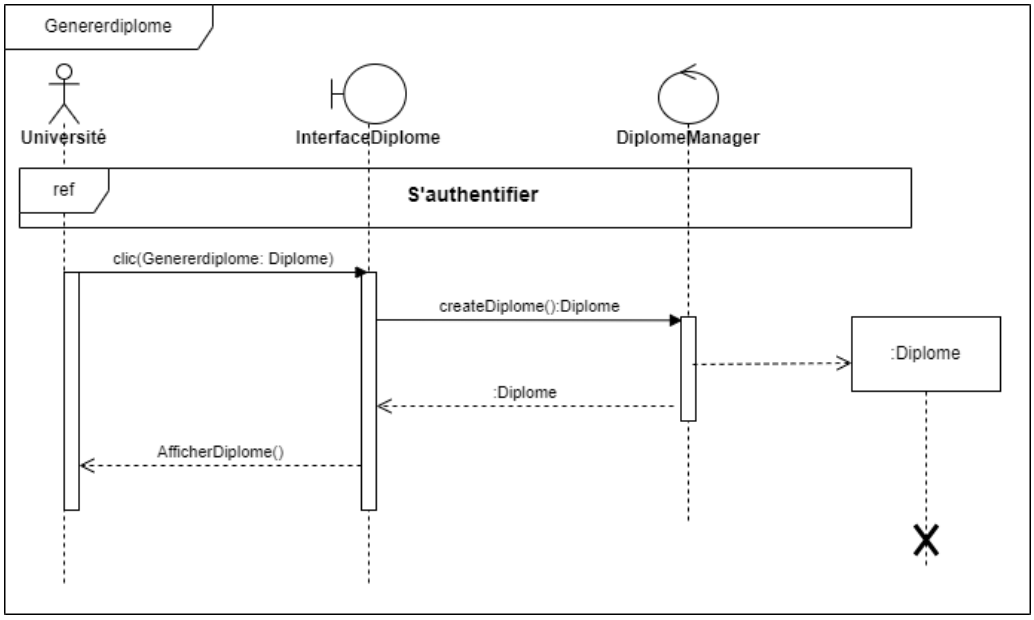


Figure 3-11 Diagramme de séquence pour générer un diplôme

3.4 Conclusion partielle

Dans ce chapitre dédié à la conception, nous avons analysé les besoins de notre système en adoptant une approche Blockchain. La conception a été faite grâce aux différents diagrammes de modélisation UML. Ainsi nous avons essayé d'exprimer le fonctionnement de notre système en nous basant principalement sur les diagrammes de cas d'utilisation et le diagramme de classe. Dans le chapitre suivant nous présenterons l'implémentation des fonctionnalités détaillées au cours de la conception faite jusqu'ici.

Chapitre 4 Implémentation du système

4.1 Introduction

Dans le chapitre précédent, nous avons présenté la conception de notre système de certification des diplômés d'une université basé sur la blockchain. Dans ce chapitre, nous allons présenter l'implémentation de ce système. Nous commencerons par décrire les outils et langages que nous avons utilisés, ainsi que les différentes fonctionnalités.

4.2 Outils et Langages de programmation

4.2.1 Les outils utilisés

a. Ethereum :

Ethereum est une plateforme blockchain décentralisée et open source qui permet de créer des contrats intelligents (*smart contracts*). Les contrats intelligents sont des programmes informatiques stockés sur la blockchain qui exécutent automatiquement les conditions de leur contrat une fois qu'elles sont remplies. Ils sont sécurisés par la cryptographie et ne peuvent pas être modifiés ou falsifiés une fois qu'ils ont été déployés. Ethereum est la plateforme de contrats intelligents la plus populaire, et elle est utilisée pour créer une grande variété d'applications décentralisées (DApps), telles que des échanges décentralisés (DEX), des plateformes de finance décentralisée (DeFi), des jeux blockchain et des jetons non fongibles (NFT) [42]. Nous l'utiliserons pour nos tests, Ganache qui simule un réseau Ethereum complet, avec ses propres nœuds, son propre minage et sa propre blockchain.

b. Truffle :

Truffle est un framework qui va permettre la création de dApp sur la blockchain. Il se compose de nombreux types d'outils nécessaires pour la création de dApp comme l'écriture des contrats intelligents avec Solidity. Cela va permettre également le test de ces contrats et leur déploiement sur la blockchain. Il va donner également un endroit pour développer l'application côté client. Il permet de démarrer facilement notre projet en utilisant des boîtes à Truffle. Une boîte est essentiellement un modèle standard permettant aux développeurs de créer rapidement des dApp robustes et adaptables [43].

c. Ganache :

Ganache est un outil qui permet de tester des contrats intelligents Ethereum sans avoir à les déployer sur le réseau principal. Ganache va nous fournir dix comptes Ethereum avec une balance de 100 ether (de faux ether) pour chaque compte, ainsi qu'une interface graphique qui nous permet d'examiner tout ce qui se passe dans cette blockchain. Cela permet de gagner du temps et de l'argent, car les transactions sur les blockchains de test sont gratuites et instantanées [44].

d. Meta-mask :

Metamask est une extension de navigateur web utilisée pour gérer les transactions de cryptomonnaie. Le Metamask connecte le navigateur web normal à Ethereum. Chaque utilisateur reçoit une clé privée et une clé publique. La clé publique est utilisée comme l'adresse du compte où la clé privée est cachée. Avec Metamask, nous pourrons nous connecter à notre blockchain Ethereum locale avec un compte personnel et interagir avec notre contrat intelligent. Metamask se chargera également de la gestion de nos fonds Ether dont nous aurons besoin pour payer les transactions [45].

e. IPFS

L'IPFS est une solution au problème de limitation du stockage de données dans la blockchain. Il permet de stocker les données décentralisées et immuables dans l'IPFS, tandis que la blockchain conserve l'adresse de hachage de l'emplacement des données. Les contrats

intelligents fonctionnent comme une sorte de pont qui relie l'IPFS et la blockchain [46]. Nous l'utiliserons dans notre système pour stocker les diplômes générer et récupérer les hash pour être enregistrer dans la blockchain.

f. Mongodb

MongoDB est une base de données NoSQL open source. Puisqu'il s'agit d'une base de données non relationnelle, elle peut traiter des données structurées, semi-structurées et non structurées. Elle utilise un modèle de données non relationnel, orienté document, et un langage de requête non structuré [47]. Nous l'utiliserons pour enregistrer les informations concernant les universités.

g. Vs Code

Visual studio code ou VS Code est un éditeur de code développé par Microsoft en 2015. Contrairement à d'autres produits Microsoft, il est l'un de ces premiers produits open source et gratuit, et surtout disponible sur les systèmes d'exploitation Windows, Linux et Mac [48].

h. Remix

Remix est un IDE en ligne qui permet de développer, déployer et administrer un *smart contract* pour les blockchains de type Ethereum. Il embarque un compilateur de script Solidity et un réseau de test afin de déployer le contrat. Il permet aussi d'avoir accès à une interface exposant les fonctions du contrat afin de tester ces fonctions [49]. Nous l'utiliserons pour tester notre *smart contract*.

i. Postman

Postman est un outil puissant utilisé pour tester les services Web et les API. Il permet de créer une requête avec la méthode HTTP et les paramètres requis, de soumettre la demande et d'inspecter les résultats [50].

j. Nodemailer

Nodemailer est un module Node.js qui permet d'envoyer des e-mails facilement à partir d'une application Node.js. Il prend en charge une variété de fournisseurs d'e-mail, notamment Gmail, Yahoo Mail, Outlook, etc. [51]. Nous l'utiliserons pour envoyer les emails.

k. Mailtrap

Mailtrap est un service pour le test sécurisé des e-mails envoyés depuis les environnements de développement. Mailtrap récupère les e-mails dans une boîte de réception virtuelle afin de tester et optimiser les campagnes d'e-mailing avant de les envoyer à de vrais utilisateurs [52]. Nous l'utiliserons pour tester la fonctionnalité des mails.

l. JWT

Le « JSON Web Token » ou JWT est une méthode sécurisée d'échange d'informations. JWT sont des jetons générés par un serveur lors de l'authentification d'un utilisateur sur une application Web, et qui sont ensuite transmis au client [53].

4.2.2 Langages de programmation

a. Javascript

JavaScript est un langage de programmation de scripts orienté objet qui s'exécute sur le navigateur, le serveur ou tout appareil disposant d'un moteur JavaScript [54].

Pour le frontend nous avons utilisé Next.js est un Framework JavaScript React qui permet de créer des applications web rapides et performantes [55]. Aussi pour faire le style nous avons utilisé Tailwind CSS qui est un Framework CSS utilitaire qui permet de créer des interfaces utilisateur (UI) rapidement et facilement [56].

Pour le backend nous avons utilisé Node.js qui est un environnement d'exécution JavaScript qui permet d'exécuter du code JavaScript côté serveur [57]. Nous avons utilisé Express.js qui est un Framework web open-source pour Node.js qui permet aux développeurs de créer des applications web rapides et évolutives [58]. Nous avons aussi utilisé WE3JS qui est une bibliothèque JavaScript qui permet de développer des applications décentralisées sur

Ethereum. Elle fournit une API complète pour interagir avec le réseau Ethereum, notamment pour créer et envoyer des transactions, interagir avec des contrats intelligents et lire les données de la blockchain [59].

b. Solidity

Solidity est un langage de programmation orienté objet de haut niveau utilisé dans l'implémentation de *smart contracts* sur diverses blockchains, et notamment Ethereum [60].

4.3 Architecture pratique

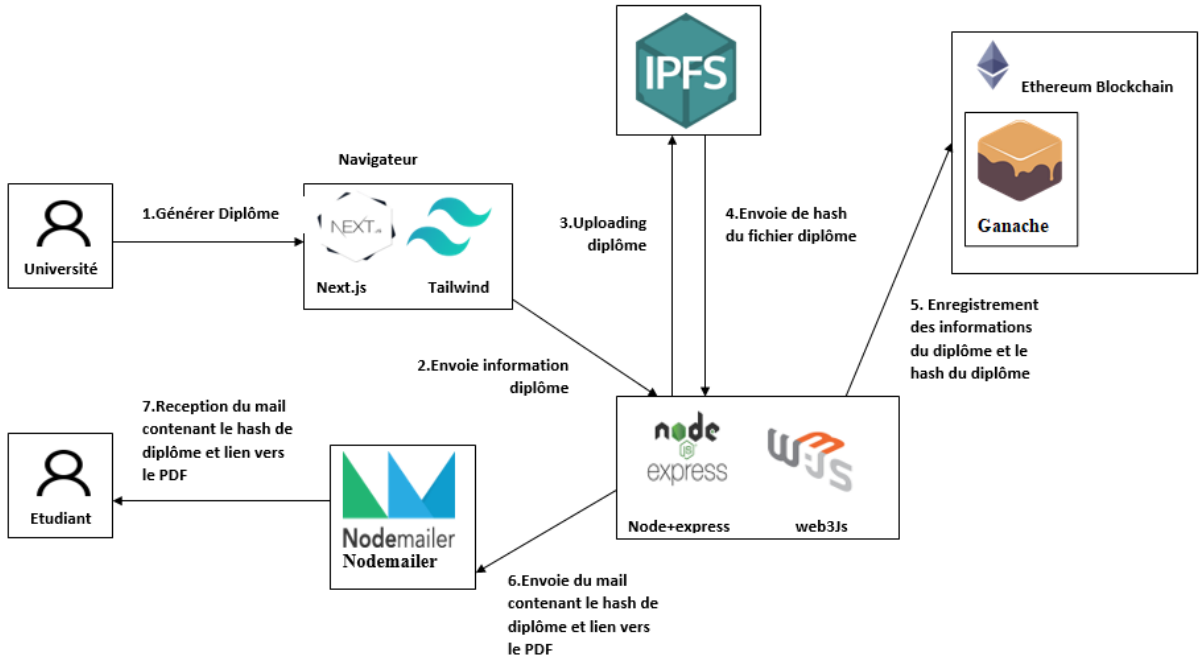


Figure 4-1 Création et envoie d'un diplôme et son hash

La figure 4-1 montre comment l'université parvient à générer un diplôme numérique et envoyer le hash du diplôme et le fichier PDF à l'étudiant.

La figure 4-2 montre comment l'université parvient à se connecter au système en envoyant par le formulaire l'adresse mail et le mot de passe requis.

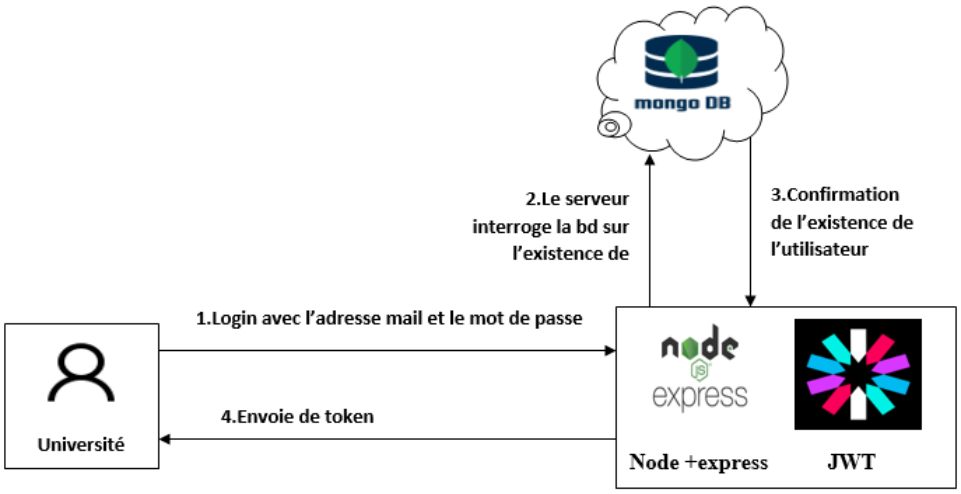


Figure 4-2 Connection au système

4.4 Déploiement du smart contract

La figure 4-3 montre la procédure que nous avons utilisé pour déployer un *smart contract*, nous avons utilisé truffle pour le déployer sur ganache.

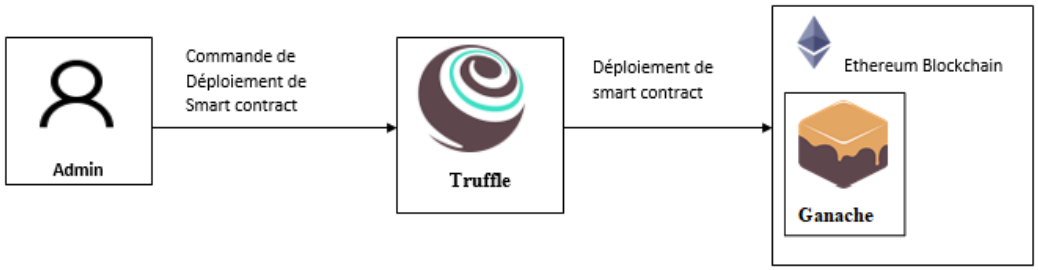


Figure 4-3 Procédure de déploiement

Voici le message renvoyé lorsque nous avons déployé le *smart contract*.

```
1_cert.js
=========

   Replacing 'Cert'
   ----------------
   > transaction hash:    0x86ec03dcddc4ab0c0622bc66265a1f8e2ecaf30081ca3bdbdb9b4ad6befc76f1
   > Blocks: 0            Seconds: 0
   > contract address:    0x5e13B301c6BF976788f4A9643c951bEC9447Cf40
   > block number:        89
   > block timestamp:     1695748591
   > account:             0x2a9f261F6bE0F7963B8895Cd92701BE1A27861d5
   > balance:             99.809700452619522234
   > gas used:            1051841 (0x100cc1)
   > gas price:           2.500029911 gwei
   > value sent:          0 ETH
   > total cost:          0.002629633961616151 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:     0.002629633961616151 ETH

Summary
=======
> Total deployments:   1
> Final cost:          0.002629633961616151 ETH
```

Figure 4-4 Message après avoir déployé le smart contract

Après déploiement, le message renvoyé comprend l'adresse de contract (contract adresse), l'adresse du compte qui a déployé le *smart contract* (account), la balance, les frais pour rémunérer les mineurs (gas price), …Ce que nous allons retenir de la figure 4-4, est que nous avons déployé un seul *smart contract* et le frais de transaction est 0.002629 ETH.

On peut voir sur la figure 4-5 que le *smart contract* « Cert » est déployé et l'adresse associée à ce *smart contract*.

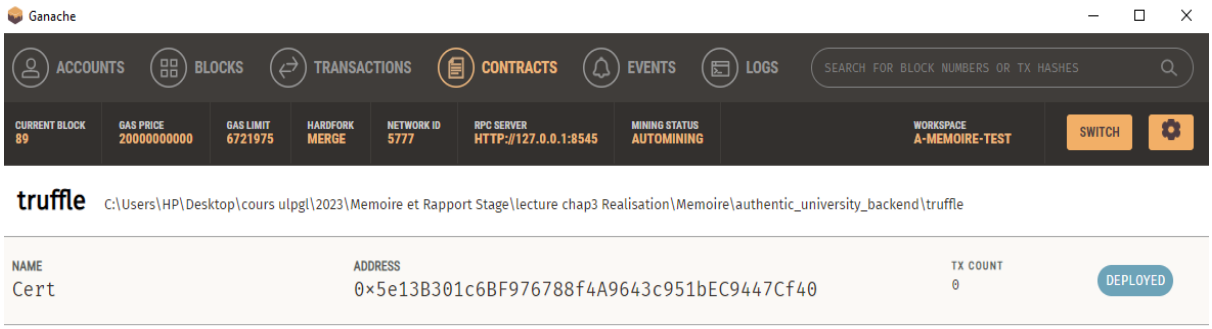


Figure 4-5 Smart contract déployer sur Ganache

4.5 Description du système

Dans cette partie, nous allons présenter notre système, nous allons montrer des captures d'écran de ses pages web principales, en commençant par l'accueil.

Vérification du diplôme

La figure 4-6 présente la capture de la page d'accueil. En arrivant à cette page, les institutions d'enseignement, les entreprises et toutes personnes possédant le hash du diplôme, peuvent vérifier l'authenticité du diplôme en cliquant sur le bouton « vérifier document ».

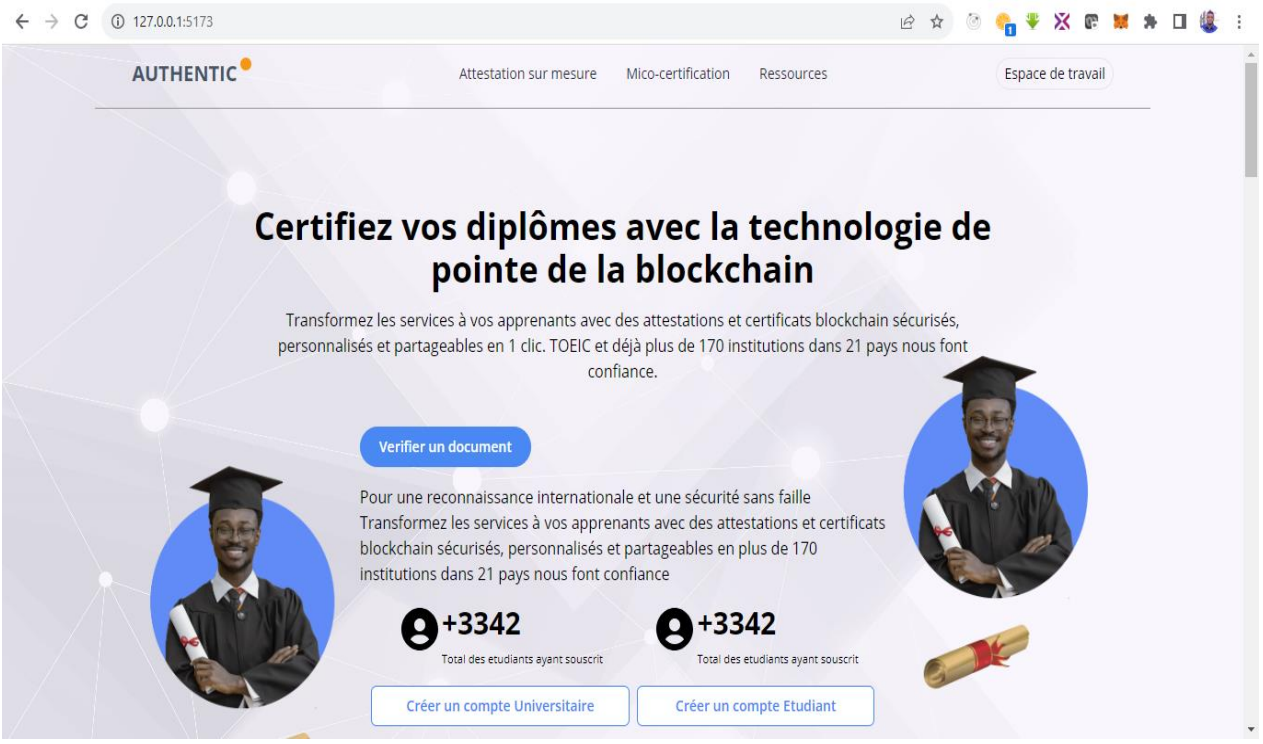


Figure 4-6 Page d'accueil

La figure 4-7 représente le modal dans lequel les utilisateurs doivent introduire le hash du diplôme pour obtenir le résultat.

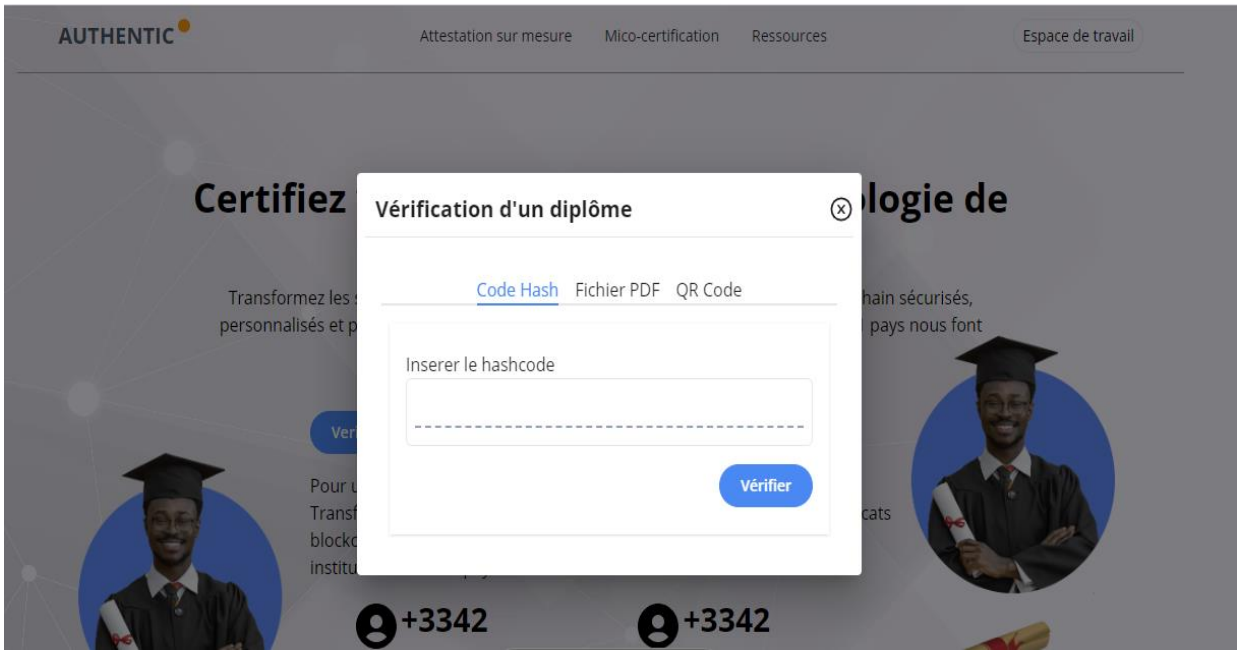


Figure 4-7 Formulaire de vérification de hash

La figure 4-8, représente le résultat de vérification lorsque l'utilisateur a introduit dans le formulaire un hash de diplôme qui existe dans la blockchain et qui n'est pas révoqué.

Figure 4-8 Résultat positif de vérification

La figure 4-9, montre le résultat de vérification lorsque l'utilisateur a introduit dans le formulaire un hash de diplôme qui n'existe pas dans la blockchain ou qui est révoquer.

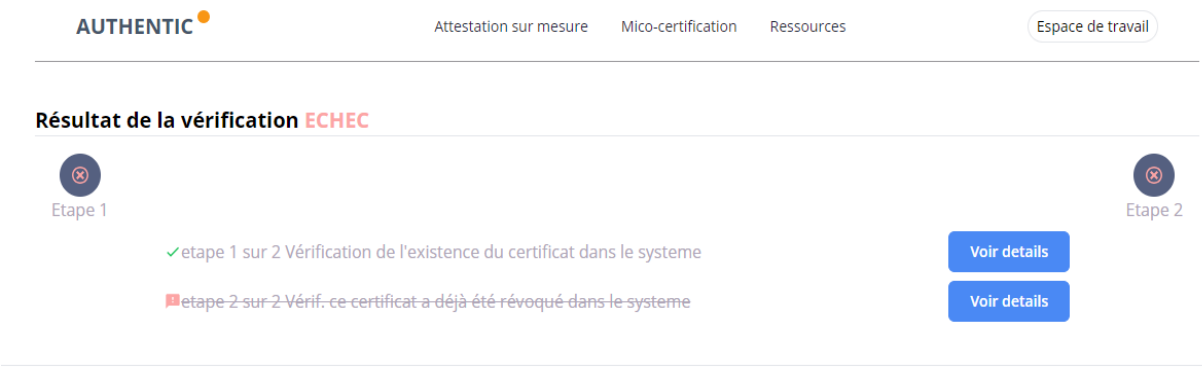


Figure 4-9 Résultat négatif de vérification

Connection au système par l'université

La figure 4-10 donne la vue de la page à partir de laquelle l'université doit se connecter, en introduisant l'adresse mail et le mot de passe.

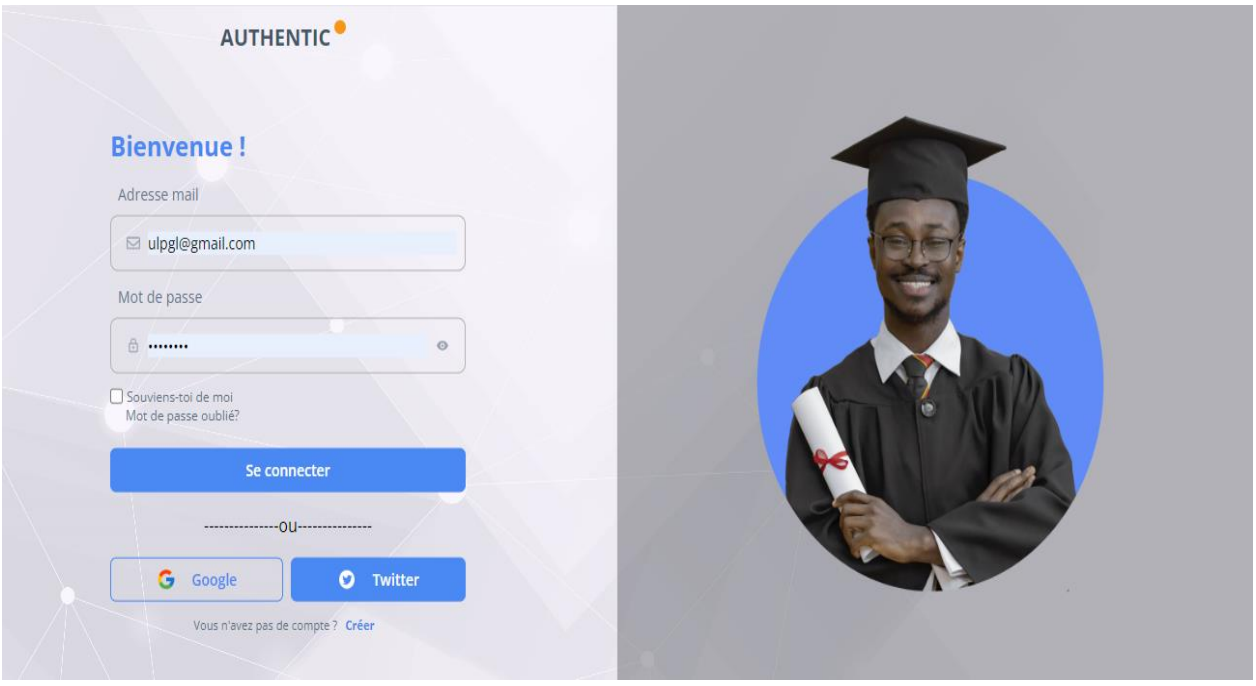


Figure 4-10 La page à partir de laquelle l'université doit se connecter

Accès au tableau de bord par l'université

L'université accède à cette page après qu'elle ait été connectée au système. La figure 4-11 montre la page à travers laquelle sont listés les diplômes déjà enregistrés dans la blockchain pour l'université. Ici l'utilisateur peut cliquer sur « Diplômes Révoqués » pour voir la liste des diplômes révoqués ; il peut aussi cliquer sur « Charger les diplômes » pour charger les fichiers csv contenant la liste des diplômés. Dans le tableau l'utilisateur a la possibilité de visualiser un diplôme particulier et révoquer un diplôme particulier.

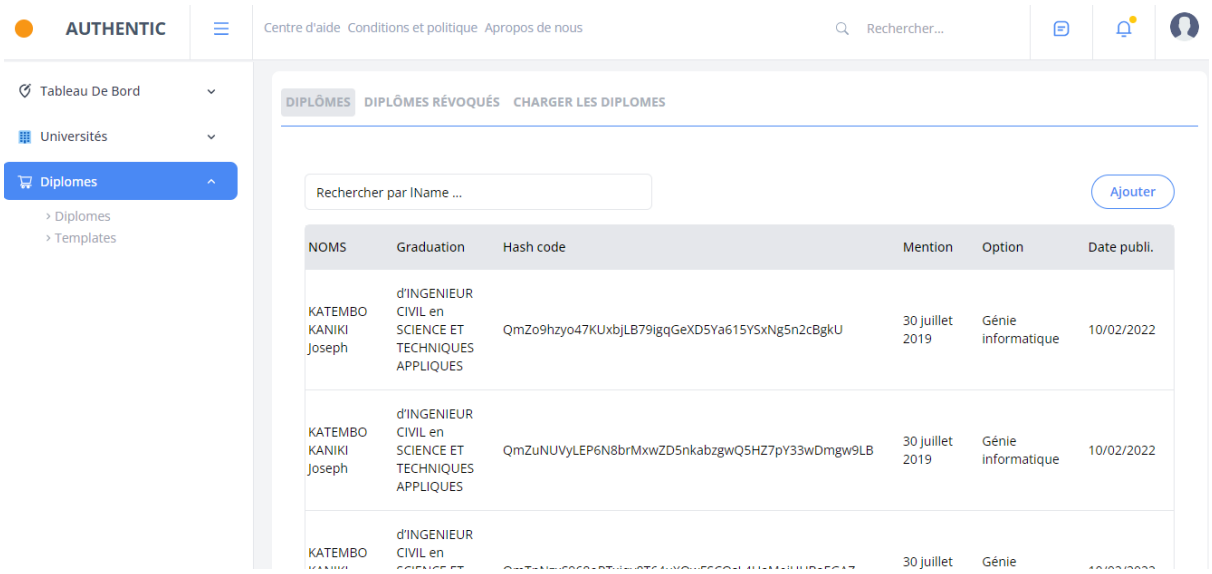


Figure 4-11 Liste des diplômes

Visualisation d'un diplôme

La figure 4-12 montre la vue du diplôme, lorsqu'on clique sur « voir » dans la colonne « visualiser » du tableau de diplômes.

Figure 4-12 Visualisation d'un diplôme

Liste des diplômes révoqués

La figure 4-13 montre le tableau des diplômes révoqués, lorsqu'on clique sur « Diplômes Révoqués » dans la colonne « visualiser » du tableau de diplômes.

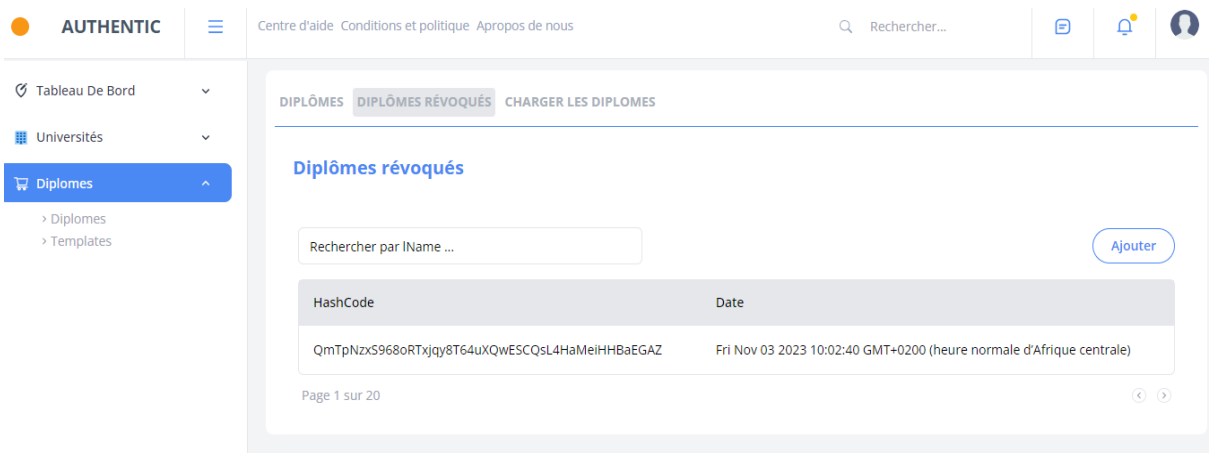


Figure 4-13 Tableau des diplômes révoqués

Envoie du diplôme à l'étudiant

La figure 4-14 donne la vue de la page à travers laquelle on soumet le fichier csv en cliquant sur « Choisir un fichier ».

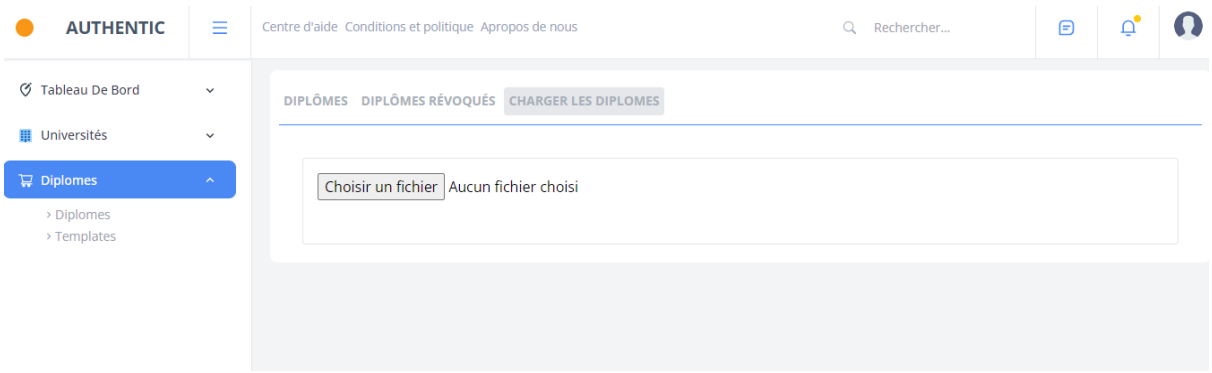


Figure 4-14 Choix du fichier csv

La figure 4-15 présente la liste de diplômés enregistrés dans le fichier csv. On doit visualiser les informations pour se rassurer que les informations dans le fichier csv soient exactes

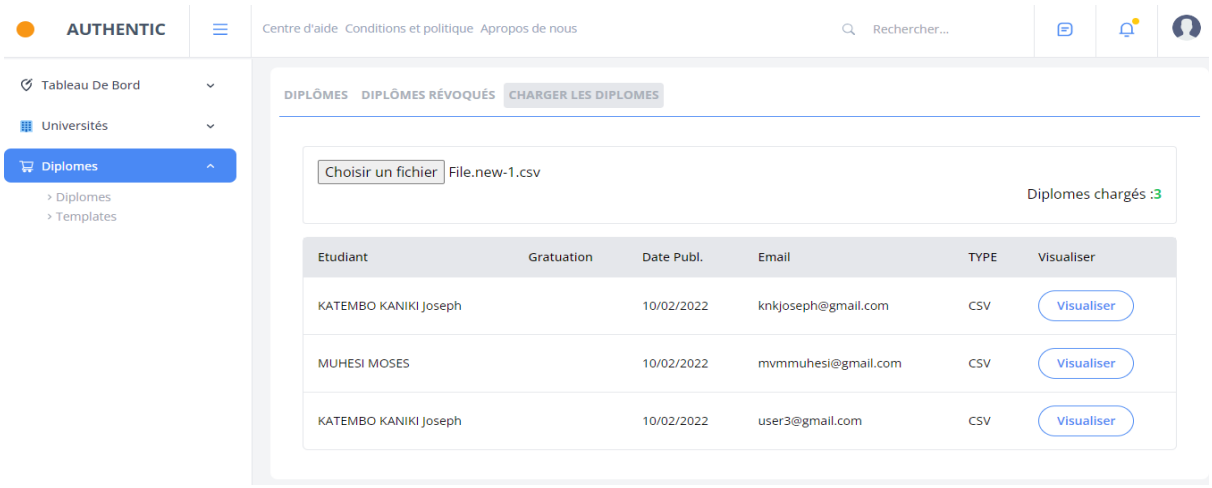


Figure 4-15 Liste de diplômés enregistrés sous format csv

La figure 4-16 montre l'opération de chargement de diplôme dans le système. Cette opération se déroule en deux phases, premièrement on génère le PDF en cliquant sur le bouton « Générer PDF », deuxièmement on charge le PDF dans le système en cliquant sur le bouton « Charger dans le système ». Le mail contenant le diplôme et son hash sera envoyé à l'étudiant après un certain temps.

Figure 4-16 Opération de chargement de diplôme dans le système.

Réception du diplôme et son hash par l'étudiant

La figure 4-17 présente un exemple de la visualisation du mail reçu par l'étudiant. L'étudiant peut télécharger le fichier pdf et reçoit aussi le code hash pour vérification de l'intégrité de son diplôme.

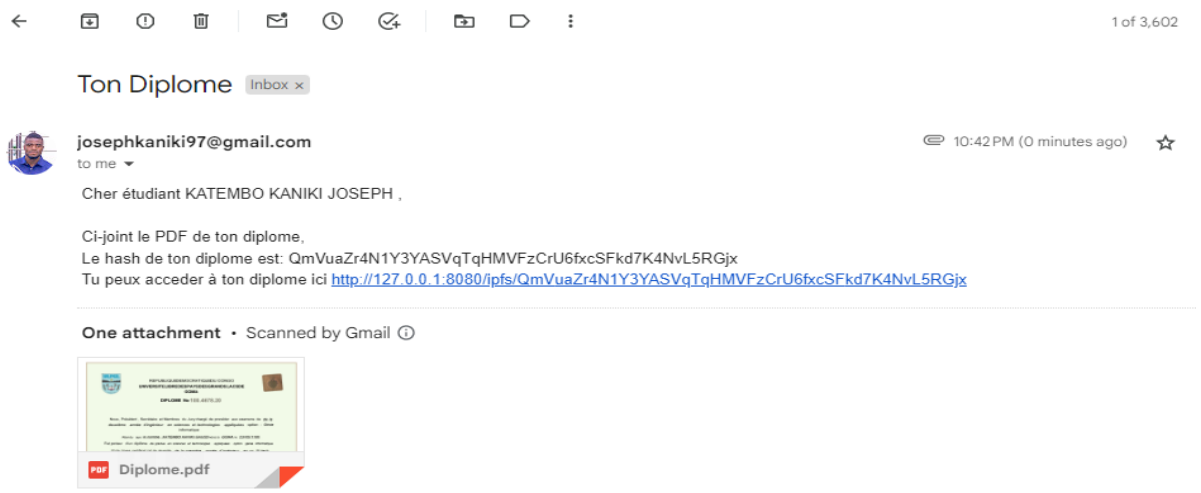


Figure 4-17 Réception du mail par l'étudiant

Liste de Templates des diplômes disponibles

La figure 4-18 illustre le tableau qui liste les Templates disponibles (Déjà codé) dans le système

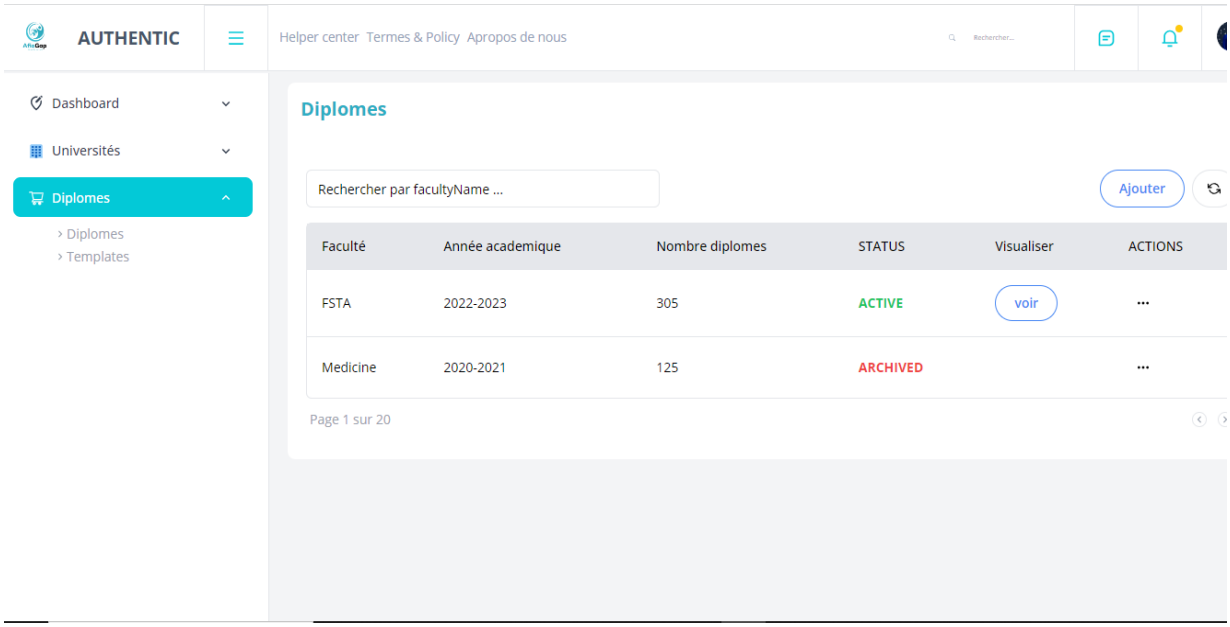


Figure 4-18 Templates disponible dans le système

4.6 Conclusion partielle

Ce chapitre a été consacré à l'implémentation et à la réalisation du système. Nous y avons présenté les outils qui nous ont servi dans le processus de son développement et le langage de programmation. Après le développement, nous avons déployé notre smart contrat dans un réseau de test et nous avons présenté les interfaces résumant le fonctionnement de notre système.

Conclusion générale

Le présent travail a consisté en la réalisation d'un « système décentralisé basé sur la blockchain pour la certification de diplômés d'une université : cas ULPGL ». En préambule, nous nous sommes fixés des objectifs à atteindre et des hypothèses à vérifier dans le cadre de ce travail, afin de trouver les réponses à nos questions de recherche qui étaient entre autres :

- Comment peut-on mettre en place une solution informatique pour prévenir de façon optimale le problème d'usurpation des diplômes ?
- Comment le système proposé servirait-il dans le processus de traitement des dossiers de candidature dans une entreprise ou institution d'enseignement ?
- Pourquoi le système proposé aurait-il un effet positif sur les institutions qui octroient les diplômes ?

Nous avons débuté notre exploration en abordant brièvement le domaine de la cryptographie, un élément clé dans la préservation de l'intégrité et de la confidentialité des données stockées sur la blockchain. Par la suite, dans le deuxième chapitre, nous avons exposé l'état actuel de la technologie blockchain. Enfin, dans le troisième chapitre, nous avons élaboré le système en utilisant le langage de modélisation UML comme notre outil principal de conception. Le choix de UML trouve son intérêt dans le désir de mieux concevoir et exprimer de façon claire les besoins du système sous forme de diagrammes UML couvrant les aspects fonctionnels, dynamiques et statiques de tout le développement. Enfin, au quatrième chapitre nous avons présenté en premier lieu les outils, les langages de programmations, l'architecture du système et la procédure de déploiement d'un *smart contract* ; et en second lieu nous avons présenté une vue détaillée de l'ensemble des interfaces du système.

La première hypothèse stipule qu'il serait possible de prévenir de façon optimale le problème d'usurpation des diplômes en mettant en place un système informatique décentralisé basé sur

la blockchain vu qu'il rendrait impossible la modification des documents et des données. Cela est démontré dans la littérature que nous avons exposée, laquelle décrit les techniques de sécurité informatique visant à assurer l'intégrité des données. Ces techniques sont utilisées dans la blockchain qui est un nouveau paradigme offrant de nombreux avantages dans la sécurité des données. En effet, la blockchain étant sécurisée par la cryptographie, cela rend très difficile la falsification ou la modification des données enregistrées sur celle-ci. Aussi, les données enregistrées sur la blockchain sont immuables, c'est-à-dire qu'elles ne peuvent pas être modifiées ou supprimées. Cela garantit l'authenticité des données. En outre, la blockchain garantit la traçabilité puisqu'elle permet de suivre l'historique des données, donnant la possibilité de lutter contre la contrefaçon et la fraude.

Deuxièmement, nous avons affirmé que le système décentralisé basé sur la blockchain pour la certification des diplômés améliorerait le processus de traitement des dossiers de candidature dans une entreprise ou institution d'enseignement vu qu'il permettrait de stocker toutes les informations pertinentes d'un candidat, et ainsi les utilisateurs voulant y accéder pourraient avoir accès selon le besoin. A l'issue de nos recherches nous avons démontré que le système proposé permet la vérification facile et rapide de l'authenticité des diplômes, vu qu'il suffit de soumettre le hash d'un diplôme dans le formulaire de vérification et avoir la confirmation sur l'authenticité de diplôme.

Dans la troisième hypothèse, nous avons soutenu que le décentralisé basé sur la blockchain pour la certification des diplômés aurait un effet positif sur les institutions octroyant les diplômes car il donnerait une assurance sur l'authenticité des diplômes. En effet, vu que la blockchain est sécurisée par la cryptographie, les données enregistrées sur la blockchain sont immuables et que la blockchain permet la traçabilité, cela garantit que les diplômes délivrés par les institutions soient authentiques et fiables. Ainsi donc, les institutions utilisant un système blockchain, comme celui conçu dans ce travail pour certifier leurs diplômes, bénéficient d'une meilleure réputation.

Étant donné que le domaine de l'informatique est extrêmement vaste et évolue rapidement, et étant donné que de nouvelles contraintes émergent constamment, nous reconnaissons que notre approche ne couvre pas tous les aspects de ce sujet. Par conséquent, nous encourageons d'autres chercheurs et développeurs intéressés par ce domaine à explorer une approche qui permettrait d'intégrer le système que nous proposons au système de gestion des diplômes de l'université. Cette intégration permettrait au système de distribuer automatiquement les diplômes et les codes de hachage aux étudiants ayant réussi dès le moment de la délibération. De plus, nous suggérons aux chercheurs d'ajouter une fonctionnalité au système proposé permettant de vérifier l'authenticité des diplômes à l'aide d'un code QR.

Bibliographie

[1] B. Hamza, «Application décentralisée basée sur la blockchain pour la signature et la vérification des Documents (front-end),» SMART TRANSFORMATION, 2021.

[2] G. M. Bihari Krishna, «A Graduation Certificate Verification Model via Utilization of the Blockchain Technology,» *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, pp. 3-2, 2018.

[3] RAPHAEL et G. YENDE, LE DOUTEUX DE LA FALSIFICATION NUMERIQUE DES DOCUMENTS DANS LE SECTEUR EDUCATIF EN RDC : les enjeux des NTIC.

[4] N. Lutfiani, «Academic Certificate Fraud Detection System Framework Using Blockchain Technology,» *Blockchain Frontier Technology(B-Front)*, vol. 1, n° %12, 2022.

[5] M. Gupta, La Blockchain pour les Nuls, Édition limitée IBM, IBM, 2018.

[6] M. PIGNEL, LA TECHNOLOGIE BLOCKCHAIN Une opportunité pour l'économie sociale ?, Pour la solidarité, 2019.

[7] R. Rokia, «Les dossiers médicaux sur Blockchain,» Universit2 Mohamed Khider, 2021, p. 61.

[8] R. Dumont, «Cryptographie et Sécurité informatique,» Liège, Université de Liège, 2010, p. 213.

[9] G. Dubertret, Initiation à la Cryptologie, 2019.

[10] D. LAMAS, La cryptographie, Genève: Haute École de Gestion de Genève (HEG-GE), 2015.

[11] A. DAIF, Les codes correcteurs au service de la cryptographie symétrique et asymétrique, Université paris8, 2019.

[12] Vayel, La cryptographie asymétrique avec RSA, Zeste du savoir, 2023.

[13] D. Lucile, «Programme d'initiation au chiffrement RSA,» 2023.

[14] Y. Shou, Cryptographie sur les courbes elliptiques et tolérance aux pannes dans les réseaux de capteurs, HAL, 2014.

[15] D. Hankerson, Guide to Elliptic Curve Cryptography, Springer, 2004.

[16] I. Bashir, Mastering Blockchain, MUMBAI: packt, 2023.

[17] C. Boura, Analyse de fonctions de hachage cryptographiques, HAL, 2012.

[18] D. PERARD, Blockchain et stockage efficace, Toulouse: Université de Toulouse, 2020.

[19] M. A. BAKHOUM, La Blockchain pour la Sécurisation, UNIVERSITE ASSANE SECK DE ZIGUINCHOR, 2019.

[20] C. Julien, Implémentation d'attaques de type Man In The Middle sur un réseau, Université d'auvergne, 2016.

[21] A. S. Ikram, «Proposition d'un système à base de blockchain pour la gestion des opérations sur les véhicules au niveau national,» Université Aboubakr Belkaïd– Tlemcen, 2018, p. 104.

[22] K. WAMUHINDO, Écrivain, *COURS DE SECURITE INFORMATIQUE.* [Performance]. ULPGL, 2023.

[23] D. Mohammed, Écrivain, *Analyse de quelques algorithmes de consensus dans la blockchain.* [Performance]. Universit´e Mohamed Seddik Benyahia de Jijel, 2020.

[24] M. QUINIOU, Glossaire Blockchain, Chaire UNESCO ITEN, 2019.

[25] C. H. M. F. COUTOR Sophie, «BLOCKCHAIN ET IDENTIFICATION NUMERIQUE,» Ministère de l'Intérieur france, 2020.

[26] M. M'HAMED, Écrivain, *Conception et réalisation d'un modèle de Blockchain intelligent.* [Performance]. Université Mohamed Khider – BISKRA, 2020.

[27] P. D. A. H. G. Dr. Burcu Sakız, «Blockchain Technology and its Impact on the Global

Economy,» chez *INTERNATIONAL CONFERENCE ON EURASIAN ECONOMIES*, 2019.

[28] X. Cai, The Future of Blockchain and Its Implications, University of Pennsylvania, 2019.

[29] P. Marrast, Blockchain : Éléments d'explication et de vulgarisation, CERTOP CNRS, 2018.

[30] D. F. ALSUBAEI, BLOCKCHAIN ADOPTION IN THE GULF STATES, Middle east Institute, 2019.

[31] F. GOMBERT, La blockchain et le Géomètre-Expert, évolutions, perspectives et limites, LBP Etude et conseil, 2021.

[32] V. FAURE-MUNTIAN, «Comprendre les blockchains : fonctionnement et enjeux de ces nouvelles technologies,» 2018.

[33] X. M. Z. Z. Mingxiao Du, «A Review on Consensus Algorithm of Blockchain,» chez *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Banff, 2017.

[34] E. G. Ribera, Design and Implementation of a Proof-of-Stake Consensus Algorithm for Blockchain, Barcelone: UPC BARCELONATECH, 2018.

[35] M. Z. A. M. Yusoff Jannah, «A Review: Consensus Algorithms on Blockchain,» *Journal of Computer and Communications scientific research publishing*, vol. 10, pp. 37-50, 22.

[36] ZoneBitcoin, «Centralisation et décentralisation : Ce qu'il faut savoir,» 14 10 2022. [En ligne]. Available: https://zonebitcoin.co/reseaux-decentralisees-definition/. [Accès le 12 9 2023].

[37] Y. Z. C. Y. Zhu Ruiqi, «A Decentralized Resource Allocation system,» 30 mars 2020.

[38] MIT, «MIT,» 2017. [En ligne]. Available: https://registrar.mit.edu/transcripts-records/diplomas/digital-diplomas. [Accès le 5 Septembre 2023].

[39] MIT, «MIT,» 2017. [En ligne]. Available: https://credentials.mit.edu/. [Accès le 5 Septembre 2023].

[40] M. M. Amine, COURS Introduction au genie logiciel (IGL), ESI, 2011.

[41] P. P. ROQUES, UML2 Modéliser une application web, Eyrolles 4e édition, 2017.

[42] J. PARIS, Apports des Smart Contracts aux Blockchains et comment créer une nouvelle crypto-monnaie, Haute École de Gestion de Genève, 2017.

[43] R. Anja, CAS D'UTILISATION DE LA TECHNOLOGIE BLOCKCHAIN DANS LE DOMAINE DE VOTE EN TANT QU'APPLICATION WEB, ANTANANARIVO: UNIVERSITE D'ANTANANARIVO, 2019.

[44] A. K. D. J. S. A. Bhosale Kumar, «Blockchain based Secure Data Storage,» *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, 2019.

[45] U. Kambli, «Blockchain Based Certification System,» *International Journal of Research Publication and Reviews*, vol. 4, n° %14, p. 5, 2023.

[46] M. Alizadeh, «Efficient Decentralized Data Storage Based on Public Blockchain and IPFS,» p. 8, 2019.

[47] K. Banker, MongoDB in Action, MANNING, 2016.

[48] M. Plainer, Study of Visual Studio Code, 2020.

[49] C. H. C. McGahon, AN OVERVIEW OF SOLIDITY, FCAT, 2023.

[50] O. luxemburg, *Test API avec Postman*, luxemburg: Oxione luxemburg, 2023.

[51] M. Fofana, «Envoi d'email via Nodemailer en utilisant Gmail avec xoauth2,» Medium, 1 3 2019. [En ligne]. Available: https://medium.com/@mfofana/envoi-demail-via-nodemailer-en-utilisant-gmail-avec-xoauth2-6b80328d5593. [Accès le 27 9 2023].

[52] B. Silia, Design and Realization of Cloud SaaS Multi-Tenant Application, IBNKHALDOUN UNIVERSITY – TIARET, 2022.

[53] S. E. Peyrott, The JWT Handbook, Auth0 Inc, 2018.

[54] C. Wenz, JavaScript L'ESSENTIEL DU CODE ET DES COMMANDES, Sams Publishing, 2019.

[55] F. Copes, Next.js Handbook, Vercel, 2019.

[56] Scrimba, Tailwind CSS Crash Course, Scrimba, 2023.

[57] M. Mohamed, Conception et développement d’une application web de modèles de planning en ReactJS et NodeJS pour l’anticipation des besoins des grands projets puis l’ordonnancement des taches., Université Mouloud MAMMERI de TIZI-OUZOU, 2019.

[58] A. Bennour, Développement d'une plateforme de reconnaissance de caractères et d'impression documentaire, Université Abou Bakr Belkaid– Tlemcen, 2019.

[59] F. Vogelsteller, web3.js Documentation, WEB3, 2019.

[60] U. D. QUEBEC, COMME EXIGENCE PARTIELLE DE LA MAITRISE EN MATHEMATIQUES ET INFORMATIQUE APPLIQUEES, NATHAN SICARD, 2023.

[61] M. Hamza, «étude et comparaison des principaux systèmes de cryptage,» UNIVERSITE MOHAMED BOUDIAF - M’SILA, 2016.

[62] A. Holemans, Implémentation d’une base de données NoSQL de données géospatiales de l’AIDE, LIÈGE: UNIVERSITÉ DE LIÈGE, 2017.

[63] Ethereum, «Solidity Documentation,» Ethereum, 2022.

Annexes

Code pour le smart contract Cert

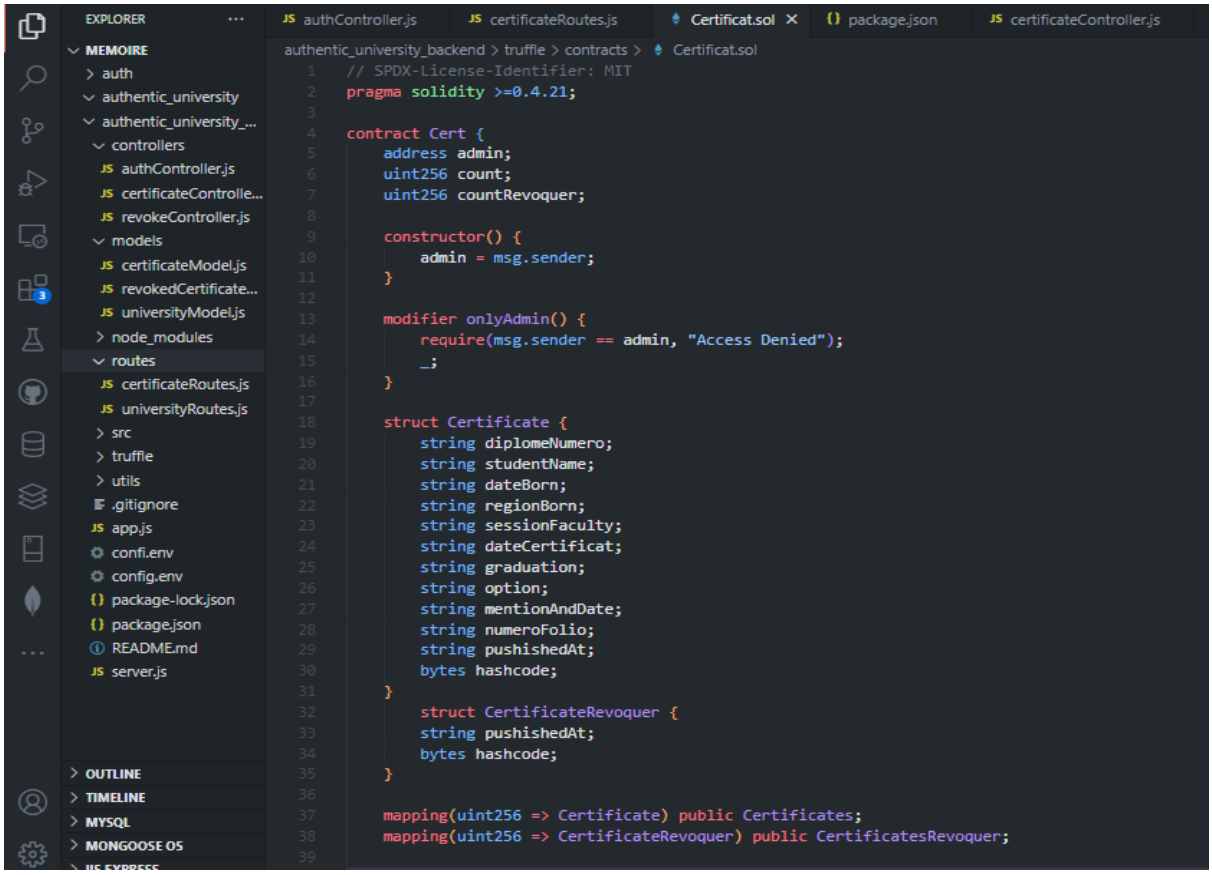


Figure 0-1 Début du de code qui implémente le smart contract Cert

Fonction pour insert les informations d'un diplôme dans la blockchain

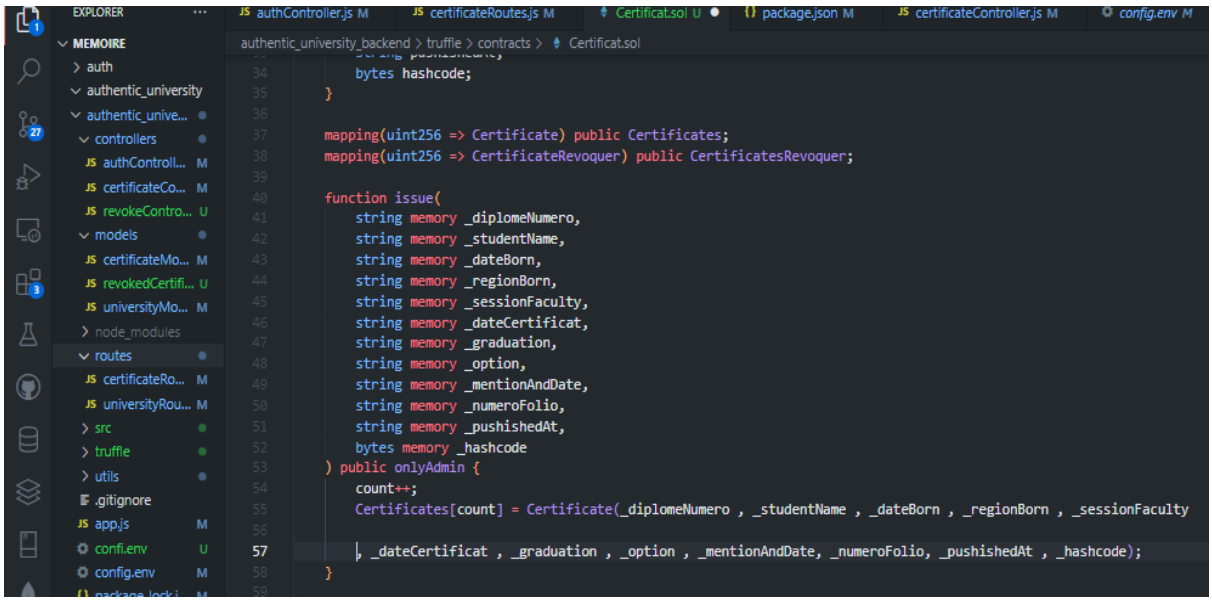


Figure 0-2 Fonction pour insérer les informations d'un diplôme dans la blockchain

Une partie du code pour l'enregistrement du fichier sur IPFS et diplôme sur la blockchain

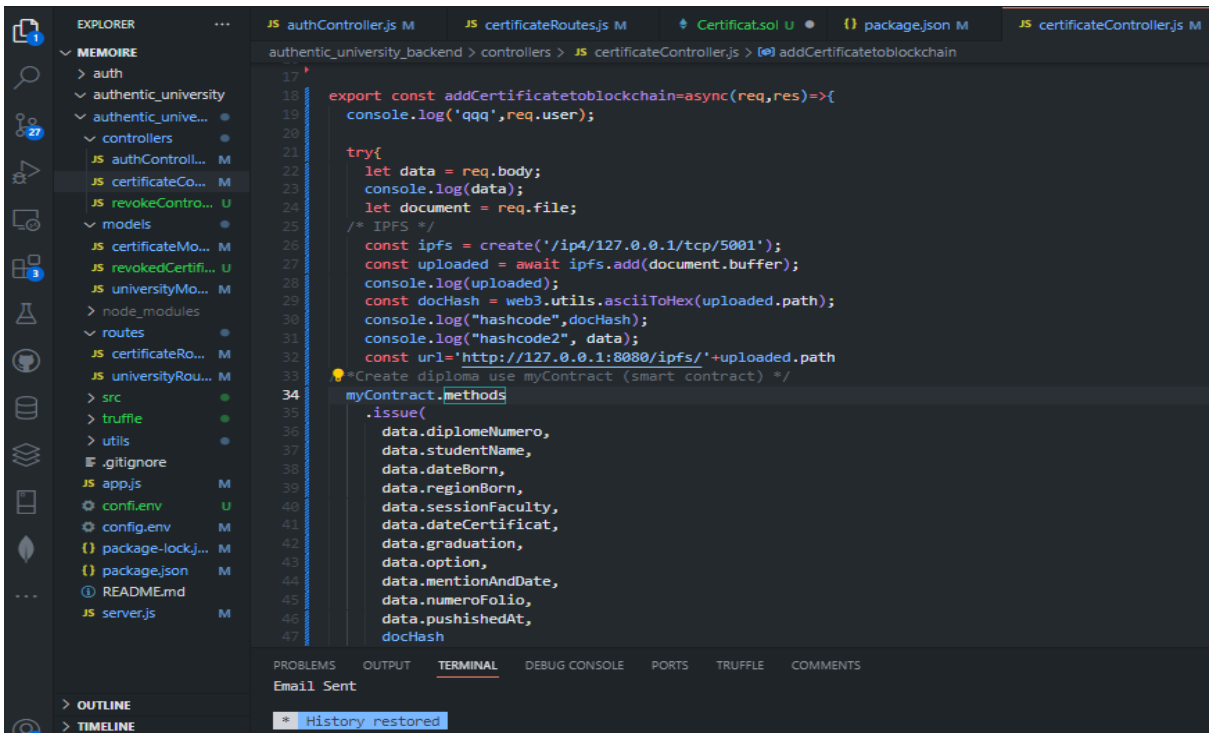


Figure 0-3 Enregistrement du fichier sur IPFS et diplôme sur la blockchain

Les transactions effectuer sur Ganache en utilisant le smart contrant Cert

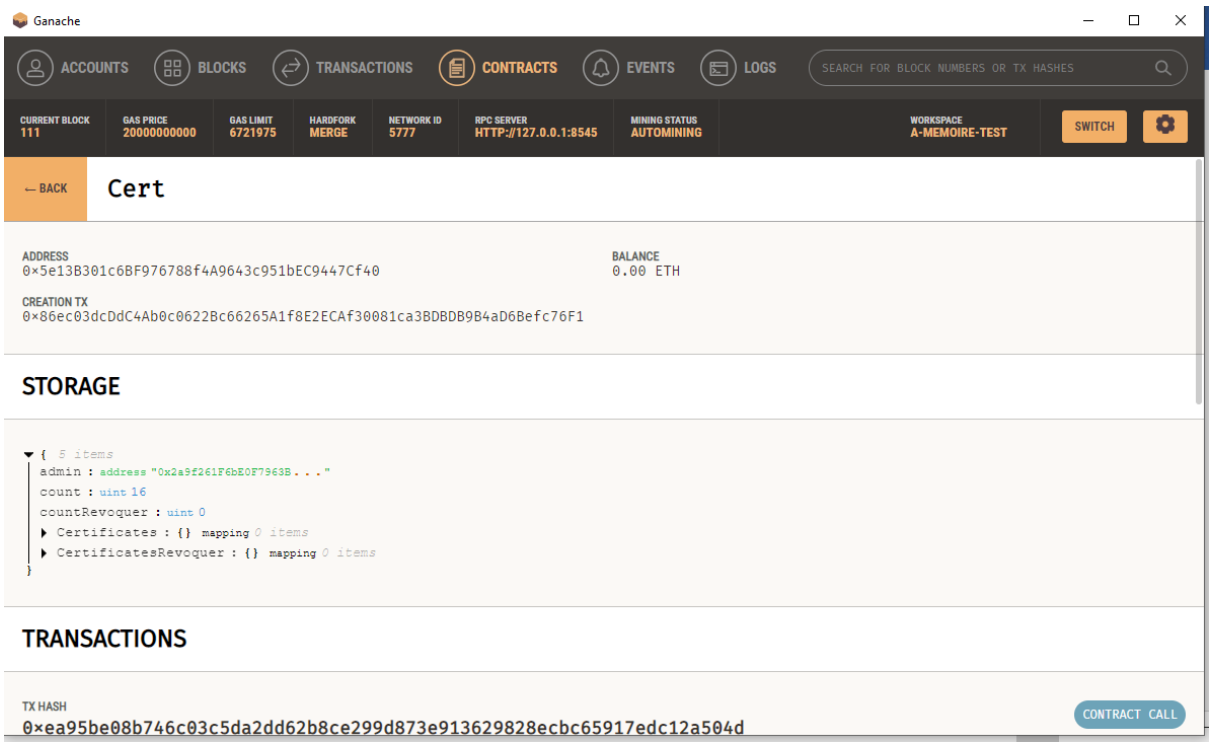


Figure 0-4 Les transactions effectuer sur Ganache en utilisant le smart contrant Cert