Analyse et Conception d'une plate-forme de validation de diplômes basée sur la blockchain

Dédicaces

Je dédie ce mémoire à mes parents pour la confiance qu'ils ont placé en moi, pour les sacrifices qu'ils ne cessent de faire pour me permettre d'évoluer, de m'accomplir et de m'épanouir en tant que personne ; pour leur immense amour et tout le soutient dont ils font preuve à mon égard. Je ne vous remercierai jamais assez. Que Dieu vous bénisse et vous accorde une longue vie paisible.

Je dédie ce mémoire à mes frères pour le soutient qu'ils me démontrent chaque jour et le refuge qu'ils sont pour moi. Que Dieu vous bénisse et qu'il bénisse vos projets afin que vous connaissiez le succès et la prospérité.

Remerciements

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma reconnaissance.

Tout d'abord je voudrais adresser toute ma gratitude au **Dr Ousmane NDIAYE** pour sa disponibilité, ses judicieux conseils et la qualité de son encadrement durant la rédaction de ce mémoire.

J'adresse mes remerciements aux membres du jury : le **Pr. Cheikh Thiécoumba GUEYE**, le **Dr. Ousmane NDIAYE** et le **Dr. Chérif Bachir DEME** pour l'intérêt et le temps qu'ils ont porté à ce modeste travail.

Mes sincères remerciements aux **Pr Mamadou SANGHARE**, **Pr Oumar DIANKHA** et au **Pr Cheikh Thiécoumba GUEYE** pour avoir rendu possible cette formation.

Ma gratitude va également à l'endroit de mes camarades **Mbao GUEYE** et **Attoumane TAHAR** qui en plus d'avoir été de belles rencontres, m'ont soutenue et accompagnée pendant la rédaction de ce document.

Enfin je ne peux terminer sans exprimer ma gratitude à tous les professeurs de l'Université Cheikh Anta Diop pour leur dévouement dans leur formation et l'assistance dont ils font preuve à notre égard tout au long de nos études universitaires. Que la grâce de Dieu soit sur vous.

Avant-propos

Le Laboratoire d'Algèbre, de Cryptographie, de Géométrie Algébrique et Application du département de Mathématiques et Informatique de la Faculté des Sciences et Techniques de l'Université Cheikh Anta DIOP de Dakar a mis en place, sous la direction du professeur Mamadou SANGHARE en 2004, un Master de Transmission des données et Sécurité de l'Information (MTDSI) et un Master d'Algèbre Géométrie et Application avec comme Spécialité Cryptologie Tatouage Réseau Génie logiciel et base de données(MAGA).

Les masters TDSI et MAGA présentent deux filières : une filière recherche et une filière professionnelle. En TDSI, il est possible depuis l'année académique 2012-2013 de le commencer en Licence première année. Il a pour vocation essentielle de former des spécialistes en sécurité des systèmes d'information.

Les diplômés de ces masters sont des informaticiens de très haut niveaux capables de proposer des solutions de sécurité sur n'importe quel support, de développer des produits de sécurité adéquats et d'auditer n'importe quel système de sécurité.

Vu sa préoccupation principale et son domaine d'action qu'est la sécurité des systèmes d'information indispensable à la survie de toute entreprise, le MTDSI et MAGA sont des formations qui allient les bases théoriques de l'enseignement supérieur aux pratiques des techniques de l'entreprise où l'expertise professionnelle est de mise.

Ces Masters offrent beaucoup de débouchés aux étudiants dont les principaux peuvent être le Développement de logiciel cryptographique, l'audit et management de la sécurité des systèmes d'information, l'étude et la mise en place de réseaux sécurisés, un responsable de sécurité informatique (RSSI) dans les établissements privés ou publics, des chiffreurs, des administrateurs de bases de données, etc...

La formation MTDSI est assurée en deux années et au terme de cette formation, l'étudiant est tenu de travailler sur un projet de mémoire qu'il devra présenter devant les membres d'un jury.

C'est dans cette optique que nous soumettons à votre appréciation le contenu de ce projet de mémoire intitulé : Analyse et Conception d'une plate-forme de validation de diplômes basée sur la blockchain.

Résumé

Ce document est un mémoire de fin de cycle en vue de l'obtention du Diplôme de Master en Transmission de Données et Sécurité de l'Information (TDSI). Il présente la mise en œuvre d'un projet intitulé « Analyse et conception d'une plate-forme de validation de diplômes basées sur la blockchain ».

La tendance actuelle, dans le domaine des données, est à la blockchain qui de par ses caractéristiques constitue une approche intéressante pour les systèmes qui doivent être protégés de la corruption, de la falsification et des interventions humaines intentionnelles ou non.

Les diplômes sont des documents importants pour quiconque aimerait se voir embaucher et évoluer en grade dans une entreprise. Ils sont le témoignage qu'une personne a suivi une formation de qualité et que de ce fait cette dernière à les capacités et compétences nécessaires pour assumer un certain poste à responsabilité au sein d'une entreprise.

Malheureusement, la dynamique actuelle tend en une résurgence de la méfiance vis-à-vis des diplômes, qui s'explique en grande partie par une augmentation des cas avérés d'usage de diplôme falsifié ou de déclaration frauduleuse d'obtention de diplôme.

C'est à ce problème que le travail présenté par la suite tente d'apporter une solution. Dans le présent document on étudie et décrit la mise en œuvre d'une plate-forme dédiée basée sur la blockchain qui pourrait concilier et rétablir la confiance entre d'une part les émetteurs de diplômes et d'autre part ceux qui souhaiteraient vérifier l'authenticité de ces derniers.

Mots-clés : Blockchain, sécurisation, vérification, diplôme

Abstract

This document is a thesis for the Master's degree in Data Transmission and Information Security (TDSI). It presents the implementation of a project entitled "Analysis and design of a platform for validation of diplomas based on the blockchain".

The current trend in the field of data is blockchain, which due to its characteristics is an interesting approach for systems that need to be protected from corruption, falsification and intentional or unintentional human intervention.

Diplomas are important documents for anyone who would like to be hired and advance in rank in a company. They are a testimony that a person has undergone a quality education and therefore has the necessary skills and competencies to assume a certain position of responsibility within a company.

Unfortunately, the current dynamic tends to be a resurgence of distrust of diplomas, which is largely explained by an increase in proven cases of falsified diplomas or fraudulent declarations of graduation.

The work presented below attempts to address this problem. In this paper we study and describe the implementation of a dedicated blockchain-based platform that could reconcile and restore trust between diploma issuers on the one hand and those who would like to verify their authenticity on the other.

Keywords : Blockchain, security, verification, degree

Sommaire

Sigles et Abréviations

Sigle ou Abréviation	Signification
HLF	Hyperledger Fabric
CA	Certification Authority
MSP	Membership Service Provider
PoW	Proof of Work
PoS	Proof of Stake
HTTPS	HyperText Transfert Protocol Secure
CV	Curriculum Vitae
AC/CA	Autorité de Certification/Certification Authority
PKI	Public Key Infrastructure
QR	Quick Response

Table des figures

Liste des tableaux

Introduction Générale

Dans un monde où les diplômes et aptitudes professionnels sont de plus en plus convoités, les individus n'hésitent pas à recourir à la falsification pour atteindre leurs objectifs et accéder à certains postes et responsabilités au détriment des conséquences que ces actes pourraient avoir. En effet la falsification des diplômes est devenue monnaie courante dans nos sociétés. Quelques 37%[1] des candidats mentiraient sur leurs diplômes selon l'institut Florian Mantione, qui interroge régulièrement les recruteurs pour estimer l'ampleur de ces falsifications. Il y a peu, en Avril 2019 au Sénégal, un homme (Alpha Ibrahima BA)[2] se faisait passer selon les circonstances comme médecin généraliste, échographiste, urgentiste ou bien même gynécologue. Un autre (Amadou Samba)[3] se présentait comme médecin titulaire d'un MBA exécutif obtenu à l'ISM de spécialisation en développement industriel du médicament. Ce dernier aurait falsifié l'entête de l'Institut Pasteur afin d'effectuer des tests de coronavirus qu'il déclarait systématiquement négatifs.

Toutes ces pratiques sont favorisées par la difficulté voire l'absence de vérification des diplômes qui sont présentés par les candidats. En effet, la vérification de diplôme est, dans nos sociétés, une opération fastidieuse qui peut prendre des semaines voire des mois avant d'aboutir, cela s'explique entre autres par les lenteurs administratives. Ceci suffit à dissuader bon nombre de recruteurs de se lancer dans ce processus.

C'est dans le but de palier à cette problématique que notre projet propose la mise en place d'une plate-forme de validation de diplômes basée sur la blockchain. En effet, depuis l'avènement de la blockchain en 2015, la perception de la confiance a été totalement bouleversée. Des opérations en tout genre qui étaient fastidieuses et nécessitaient des tiers de confiance tels que les universités, se voient aujourd'hui simplifiées par la seule utilisation des blockchains. La blockchain étant un registre partagé à la fois transparent et sécurisé, les universités pourraient y publier en toute quiétude leurs diplômes et les recruteurs pourraient à leur tour vérifier en temps réel la validité des diplômes reçus. Il s'agit ici de mettre en place une

[1] https://www.sencampus.com/mentir-sur-un-diplome-peut-couter-cher/
[2] https://infomed.sn/le-faux-gynecologue-lexpert-des-medias-et-des-reseaux-sociaux-demasque/
[3] https://afrique.le360.ma/senegal/societe/2020/03/30/30008-senegal-coronavirus-lincroyable-histoire-dun-faux-toubib-qui-pratiquait-des-tests-sur-des

solution facile, efficiente et efficace qui faciliterait la publication, la validation et la vérification de l'authenticité des diplômes.

Afin d'atteindre cet objectif, ce travail est organisé en deux grandes parties. Dans la première partie nous allons exposer le cadre théorique, à travers le contexte, la problématique et les objectifs à atteindre. Puis en deuxième partie, nous allons aborder le cadre conceptuel du projet en exposant la méthodologie et en spécifiant les besoins avant de passer à l'implémentation proprement dite.

Partie I : Cadre Théorique

Chapitre 1 : Présentation Générale

Nous présenterons dans ce chapitre le sujet de notre projet en parlant de son contexte, de la problématique et des objectifs à atteindre.

1.1. Mise en contexte

Dans ce monde de la donnée où il est très facile d'accéder à internet, la société a désormais pris l'habitude d'avoir accès à l'information rapidement. Ceci est d'autant plus important pour les informations décisives. Nous estimons que les diplômes sont un actif important et l'un des facteurs les plus indispensables dans le cadre d'une démarche de recrutement.

Le diplôme en général, est un acte émanant d'une autorité légale et qui confère ou confirme un titre, un grade. Spécialement, c'est un titre ou grade émanant d'une université, d'une grande école, d'un organisme d'enseignement habilité, etc., sanctionnant la fin d'un cycle d'études et dont l'obtention est souvent nécessaire pour pouvoir exercer certaines professions[4].

En effet, il est la preuve qu'un individu a été formé et qu'il a acquis toutes les compétences indispensables à l'exécution de tâches ou la mise en œuvre de processus complexes. Ces tâches ou processus peuvent relever de domaines professionnels délicats tels que la médecine, où l'on peut avoir à pratiquer des opérations chirurgicales et où clairement la survie des patients dépend totalement des compétences du médecin ; ou encore l'aéronautique, où l'on ne confierait pas un avion à piloter à une personne dont on n'est pas sure des compétences.

Il est donc nécessaire pour les recruteurs de pouvoir s'assurer des compétences des candidats qu'ils reçoivent avant de leur confier leurs entreprises. Cela passe par des interviews entre recruteurs et candidat et est souvent conditionné par la présentation d'un diplôme de fin d'étude délivrer par organisme d'enseignement compétent.

[4] https://academie.atilf.fr/9/consulter/diplome?page=1

1.2. Problématique

Malheureusement force est de constater que les diplômes présenter par certains candidats ne sont pas toujours authentique et ne sont pas forcément délivrer par les autorités compétentes. Certains individus n'hésitent pas à falsifier les documents et à présenter de faux diplômes dans le but d'accéder à certains postes ou s'inscrire dans certaines formations. D'autres préfèrent user de la corruption ou de leur relation dans le milieu pour se voir attribuer des diplômes dont ils n'ont pas le mérite.

Ces pratiques sont d'autant plus répandues à cause de la difficulté que rencontre les recruteurs à vérifier l'authenticité d'un diplôme.

En effet, pour vérifier l'authenticité d'un diplôme un recruteur doit retrouver le contact téléphonique ou la boite électronique de l'école concernée puis leur faire part de son besoin de vérifier un diplôme attribuer à un individu. Dans le meilleur des cas, l'école accuse réception de la demande du recruteur puis se lance dans une recherche active dans les archives des dossiers des étudiants pour tenter de retrouver le diplôme ou attestation de diplôme dont il s'agit avant de pouvoir contacter en retour le recruteur pour lui faire part d'une réponse. Dans le pire des cas, le recruteur ne reçoit pas de réponse de l'école qu'il a contacté.

Tous ces facteurs font que les recruteurs plutôt que de se lancer dans cette procédure pour faire authentifier un diplôme préfèrent parfois s'en remettre aux documents présenter par le candidat.

Dans le cas d'une école qui recrute un étudiant venant de l'extérieur, il est parfois nécessaire que l'étudiant fasse établir une équivalence de son précédent diplôme auprès d'autorités compétente. La procédure administrative d'établissement d'équivalence est aussi fastidieuse que lente ce qui complexifie l'obtention de ce dernier. En effet elle met en collaboration plusieurs administrations telles que les écoles et les ambassades dont les procédures administratives sont déjà lentes. Ainsi plusieurs étudiants entament leurs études sans au préalable avoir reçu leurs équivalences de diplôme.

Aux vues de ces différents faits, des questions se posent : Comment mettre en place un système sécurisé, ergonomique et hautement disponible de publication et de vérification des diplômes, communs à toutes les structures de formation ? Comment protéger les données publiées contre toute modification d'administrateur au comportement arbitraire ?

1.3. Objectifs

L'objectif de ce projet est la mise en place d'une solution qui permettra la publication et la vérification de l'authenticité des diplômes. Pour ce faire, nous avons choisi d'exploiter la technologie de la blockchain.

Le choix de la blockchain comme technologie sous-jacente pourrait permettre de répondre en grande partie à cette problématique. Plus précisément l'utilisation d'une blockchain hybride permettrait non seulement de garantir que seules les entités autorisées peuvent publier les diplômes, mais également toute personne désireuse de vérifier un diplôme peut le faire par le biais d'une l'application cliente dédiée. Elle permettrait en sus de garantir que toute modification, ajout ou suppression d'un diplôme stocké soient historisés, et l'entité responsable identifié par le biais de primitives cryptographiques. Enfin elle garantirait une haute disponibilité même dans ce contexte ou les structures de formations ne sont pas toutes dotées d'une infrastructure interne robuste.

Le but de ce travail est donc de mettre à la disposition des organismes d'enseignement et recruteur un système qui pourra :

- Faciliter la vérification de l'authenticité d'un diplôme autant pour les particuliers que les organismes d'enseignement ;
- Empêcher la modification et l'attribution frauduleuse et arbitraire de diplômes ;
- Permettre aux organismes d'enseignement de pouvoir garder leur autonomie sans avoir à dépendre d'un tiers de confiance ;
- Permettre de vérifier tous les diplômes quel que soit leur provenance ;

Chapitre 2 : Rappel cryptographique

Nous allons dans ce chapitre faire un rappel sur les outils cryptographiques indispensables à la réalisation de notre projet et à la compréhension de la technologie blockchain.

La cryptographie, la science du secret, a pour but de protéger les informations de l'entreprise contre la lecture non autorisée, la modification malveillante ou pas, le refus de responsabilité lors d'une participation à une transaction, etc. C'est ainsi l'un des éléments principaux ayant rendu possible l'invention des crypto-monnaies et des blockchains utilisées dans le commerce.

2.1. Définitions et Terminologies

La cryptologie est la science du secret, c'est une science pure qui émet des idées et des principes qui servent de base à la transformation des informations claires en informations inintelligibles à toute personne non qualifiée pour les connaître, ou à réaliser le processus inverse grâce à des moyens matériels et logiciels conçus à cet effet. Elle conçoit et analyse des mécanismes permettant, entre autres, d'assurer l'authentification, l'intégrité, la non répudiation et la confidentialité des données.

La cryptologie est composée de deux branches :

- cryptographie : elle permet la conception de primitives cryptographiques ;
- cryptanalyse : elle permet l'analyse de la sureté des primitives cryptographiques ;

Les algorithmes cryptographiques peuvent être subdivisés en 3 grandes branches :

• **Cryptographie symétrique ou à clés secrètes** : De l'antiquité à 1976 c'était la seule forme de cryptographie existante. Dans cette configuration, deux personnes ont une méthode de chiffrement et de déchiffrement pour lesquelles elles partagent une même clé secrète.

• **Cryptographie asymétrique ou à clé publique** : Dans la cryptographie à clé publique un utilisateur possède une clé privée comme le cas symétrique mais également une clé publique. Les algorithmes asymétriques sont utilisés pour plusieurs applications notamment pour la signature numérique ou l'échange des clés.

• **Cryptographie hybride** : famille d'algorithmes faisant appel aux deux grandes familles.

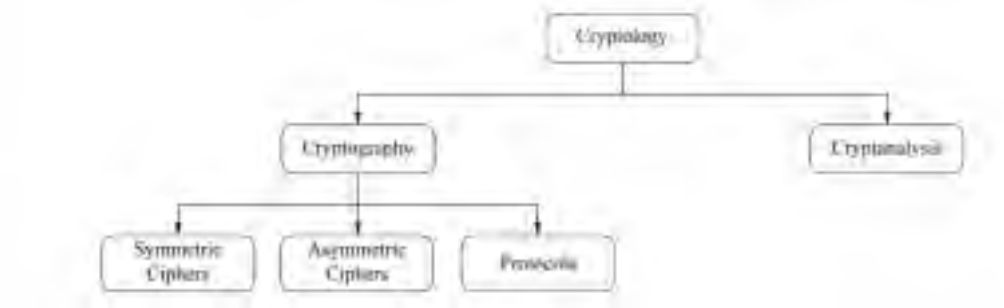

Figure 1: Branche informatique de la cryptographie

La cryptologie est utilisée dans toutes les branches de la sécurité dans lesquelles elle assure différents services :

1. Confidentialité : qui assure que le sens de l'information reste caché à tout utilisateur, entité ou processus non-autorisé. La confidentialité est assurée principalement grâce aux algorithmes de chiffrement symétrique ou à clés secrètes.

2. Intégrité : assure que le destinataire d'un message soit capable de vérifier si le message a été altéré ou modifié durant la transmission que ce soit de manière arbitraire (bruit du canal de communication), ou intentionnelle (par un attaquant). Personne ne devrait être capable de substituer un message ou une partie du message. L'intégrité est assurée principalement par les fonctions de hachage.

3. Authentification : assure que le destinataire d'un message puisse vérifier son origine. Personne ne devrait pouvoir envoyer un message à Bob en prétendant être Alice. L'authentification permet de s'assurer de l'identité de l'expéditeur de l'information, et par conséquent, de confirmer son identité. Elle est assurée par des protocoles d'identification.

4. Non répudiation : assure que l'on ne puisse pas nier d'avoir participer à une transaction à laquelle on a pris part. Elle permet d'obtenir la preuve de l'émission ou de la réception d'une information. Elle est assurée par la signature électronique.

2.2. Cryptographie symétrique

La cryptographie symétrique, également dite à clé secrète (par opposition à la cryptographie asymétrique), est la plus ancienne forme de chiffrement. Elle permet à la fois de chiffrer et de déchiffrer des messages à l'aide d'un même mot clé.

Elle fournit à deux parties, Alice et Bob, les moyens de communiquer de manière secrète. Pour mettre en place leur canal de communication sûr, ils doivent d'abord se mettre d'accords sur la clé *k*. Ils doivent garder leur clé partagée secrète.

Avant qu'Alice n'envoie un message *x* à Bob, Alice chiffre *x* en utilisant un algorithme de chiffrement *E* et la clé *k*. Elle obtient ainsi un cryptogramme $y = E_k(x)$ et envoi *y* à Bob. En utilisant l'algorithme de déchiffrement *D* et la même clé *k*, Bob déchiffre *y* pour retrouver le texte clair $x = D_k(y)$. Nous appelons ces méthodes de chiffrement : chiffrement symétrique car Bob retrouve le message *x* en utilisant la même clé *k* qu'Alice a utilisé pour chiffrer.

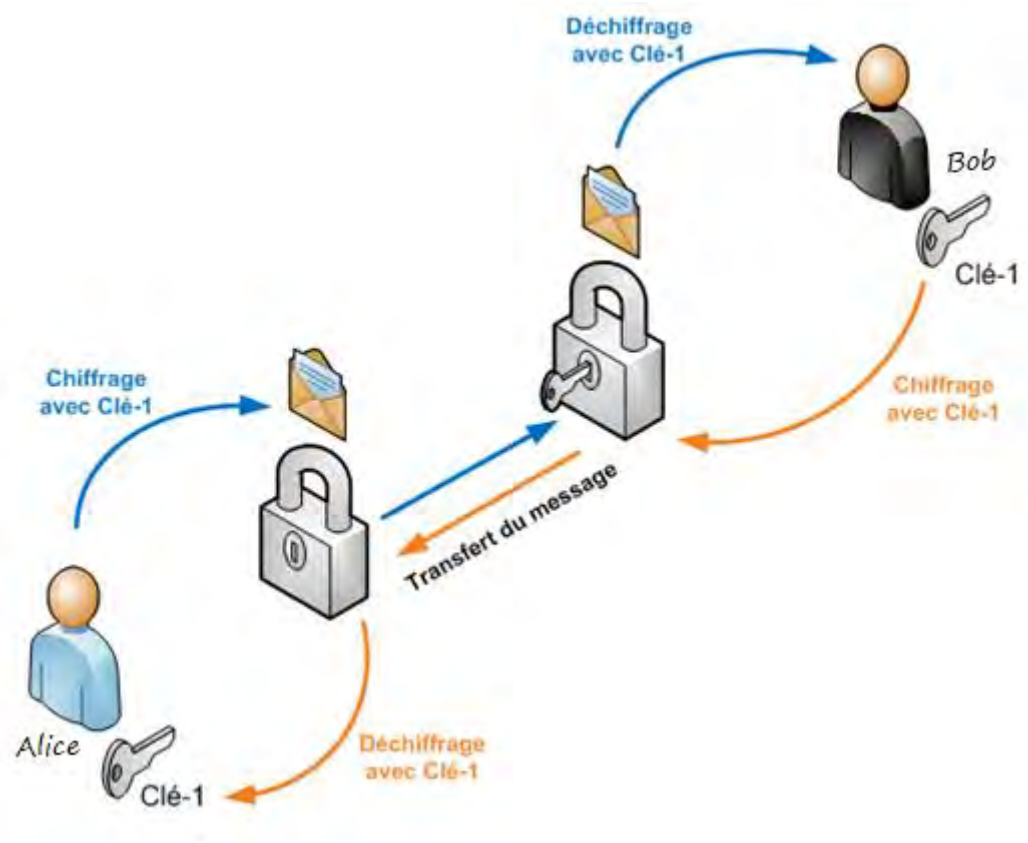

Figure 2 : Chiffrement Symétrique

2.3. Cryptographie asymétrique

Limite de la cryptographie symétrique : On rappelle que pour la cryptographie symétrique chaque participant à la communication doit avoir une copie de la même clé. Ce qui conduit à un problème de gestion des clés complexes, puisque pour un groupe de *n* individus, il faudrait $\frac{n(n-1)}{2}$ clés symétriques différentes pour que la communication entre deux individus quelconques du groupe soit possible. De plus la cryptographie symétrique pose un problème de distribution des clés. En effet, pour établir une communication, Alice et Bob doivent au préalable s'échanger la clé afin de disposer de la même clé ; or le canal de communication n'étant pas sûre, il ne peut donc pas être utilisé pour transmettre la clé, ce qui serait la manière la plus efficace de le faire. Enfin, la cryptographie symétrique ne permet pas de prévenir contre la tricherie. Alice et Bob ayant la même clé, elle les octroie les mêmes caractéristiques. Par conséquent, la cryptographie symétrique n'est pas adaptée à un usage dans lequel on souhaiterait éviter que l'une des parties Alice ou Bob ne puissent tricher. Par exemple dans les applications de commerce digital où il est souvent nécessaire de prouver qu'Alice à

effectivement commandé une voiture jaune. Si on utilise uniquement la cryptographie symétrique et que Alice change d'avis, elle pourrait accuser Bob, le vendeur, d'avoir générer lui-même la commande. Prévenir ce cas de figure c'est ce que l'on appelle la non répudiation.

2.3.1. Principes

Pour surmonter ces difficultés mentionnées ci-dessus, Diffie, Hellman et Merkle ont fait une proposition révolutionnaire basée sur l'idée suivante : *"Il n'est pas nécessaire que la clé que possède celui qui chiffre soit secrète. La partie cruciale est que Bob, le destinataire, puisse déchiffrer seulement en utilisant une clé secrète"*. Pour réaliser un tel système, Bob publie une clé de chiffrement publique qui sera connue de tous (au moins accessible par tous). Bob doit avoir une clé secrète correspondante qui sera utilisée pour le déchiffrement. Ainsi, la clé *k* de Bob est composée de deux parties, une partie publique ou clé publique, P_{KB}, et une partie privée ou clé privée S_{KB}.

2.3.2. Algorithmes à clé publique importants

Il existe plusieurs familles d'algorithmes asymétriques classées selon le type de problème utilisé. Il y a cependant trois grandes familles qui sont très largement implémentées à savoir :

- Algorithmes basés sur la factorisation d'entiers : Plusieurs algorithmes sont basés sur le fait qu'il est difficile de factoriser de grands entiers. Le plus connu de cette famille est l'algorithme de chiffrement RSA.
- Algorithmes basés sur le problème du logarithme discret : Il existe plusieurs algorithmes basés sur le DLP. Les plus connus étant le protocole d'échange de clé Diffie-Hellman, l'algorithme de chiffrement El Gamal ou le Digital Signature Algorithme (DSA)
- Algorithmes basés sur les courbes elliptiques : Il s'agit d'une généralisation du problème du logarithme discret sur *le groupe* des courbes elliptiques. Les exemples les plus connus inclus le protocole d'échange de clé Diffie-Hellman basé sur les courbes elliptiques (ECDH) et l'algorithme de chiffrement El Gamal basé sur les courbes elliptiques.

En plus de ces trois familles, il y a eu plusieurs propositions pour la cryptographie asymétrique. Certains manques de maturité cryptographiques comme celle basé sur les réseaux Euclidien ou fonctions multivariées quadratiques d'autre comme le crypto système de McEliece ont des problèmes d'implémentation (taille des clés trop grande de l'ordre de mégaoctets).

2.4. Signature numérique

La signature électronique partage certains éléments avec la signature manuscrite dans le sens où elles permettent de s'assurer qu'un message est émis spécifiquement par un individu donné. La signature numérique offre cependant plus de fonctionnalité.

Comme pour les signatures manuscrites, pour les signatures numériques seule la personne qui crée le message devrait être capable de le signer. Pour accomplir cela avec les primitives cryptographiques, nous devons partir des principes de la cryptographie asymétrique. L'idée de base étant que la personne qui signe un message utilise une clé privée, et celui qui reçoit le message utilise la clé publique associé pour le vérifier.

Ce schéma de signature pose le problème de l'authenticité des informations/clés publiques. En effet, comment Alice ou Bob peuvent être sûre qu'ils possèdent chacun la clé publique de l'autre ? Comment empêcher Oscar de distribuer de fausses clés publiques pour une attaque. Pour joindre à la signature électronique cette fiabilité on a souvent recours à un tiers de confiance sous la forme d'une Autorité de Certification (AC). Ce dernier lie l'identité d'une entité (Alice, Bob, Banque B) à un certificat numérique sur la base d'une infrastructure PKI. On parle de signature numérique.
La plupart des schémas de signature utilisent une fonction de hachage et appliquent la signature sur ce dernier, il sera supposé dans la suite qu'on se trouve dans ce cadre sauf mention du contraire.

2.5. Fonction de hachage

Les fonctions de hachage sont des primitives cryptographiques importantes largement utilisées dans les protocoles. Elles permettent de condenser un message d'une longueur quelconque en une courte chaine de bits de longueur fixe. Le message condensé ainsi formé

peut être considérer comme l'empreinte digitale dudit message. Les fonctions de hachage ne font pas usage de clé contrairement à plusieurs algorithmes cryptographiques.

Les fonctions de hachages ont de nombreuses applications et sont très connues pour le rôle qu'elles jouent dans les signatures numériques. Pour les systèmes de signatures basés sur les algorithmes asymétriques, la longueur du texte en clair est limitée. Dans le cas de RSA, par exemple, la longueur du message ne peut pas dépasser le module, qui est souvent compris entre 1024 et 3072 bits ; c'est-à-dire 128 à 384 octets. La plupart des messages sont plus long que cela. Calculer efficacement la signature de grands messages reviendrait alors à subdiviser ce dernier en blocs de tailles inférieures à la taille d'entrée autorisée par l'algorithme de signature, et à signer chaque bloc séparément comme l'illustre la figure ci-dessous :

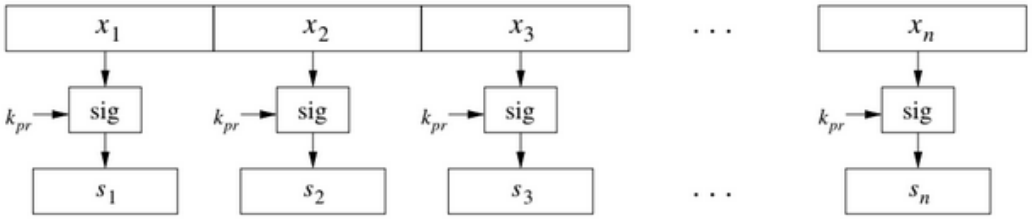

Figure 3: Signature d'un long message avec un algorithme asymétrique

Cependant cette approche pose 3 sérieux problèmes :

Les signatures numériques sont basées sur des opérations asymétriques à forte intensité de calcul ce qui induirait une charge de calcul élevée. Même si une seule opération consomme une petite quantité de temps et d'énergie, la signature de messages volumineux tels que les pièces jointes ou fichier multimédia prendrait trop de temps. Non seulement le signataire devra calculer la signature, mais le vérificateur devra aussi passer une quantité similaire de temps et d'énergie afin de vérifier la signature.

De plus, cette approche double la charge du message car il faut non seulement envoyer le message mais aussi la signature qui est de même longueur que le message. Par exemple un fichier de 1Mo donne lieu à une signature RSA de 1Mo, ce qui signifie qu'un total de 2Mo doit être transmis.

Enfin cette approche induit des problèmes de sécurité car un attaquant pourrait supprimer des blocs du message et leurs signatures, ou bien il pourrait réordonner les messages et les signatures, ou encore il pourrait réassembler de nouveaux messages et signatures à partir

de fragments antérieurs. La sécurité est assurée à l'intérieur des blocs individuels mais pas sur l'ensemble du message.

Par conséquent pour des raisons de performances et de sécurités, le mieux serait d'avoir une signature courte pour un message de taille arbitraire. C'est à cette problématique que les fonctions de hachage apportent une solution. En effet, la fonction de hachage calcule, à partir d'un message de taille arbitraire, une emprunte unique de taille fixe qui pourrait être utilisé pour effectuer l'opération de signature comme le montre la figure ci-dessous :

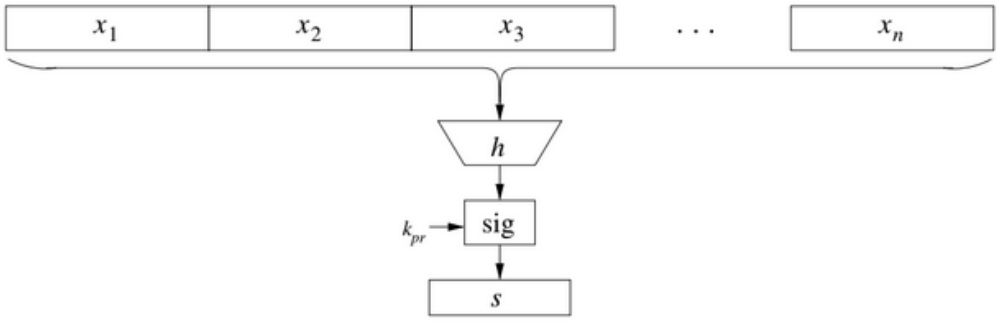

Figure 4: Signature d'un long message avec une fonction de hachage

Supposons que Bob désire envoyer un message signé numériquement à Alice. Bob détermine le hash du message à partir d'une fonction de hachage puis signe ce hash avec sa clé privée. A la réception, Alice détermine la valeur du hash du message reçu avec la même fonction de hachage puis vérifie la signature avec la clé publique de bob. La génération de la signature de même que la vérification de cette dernière s'effectuent donc sur le hash du message et non sur le message lui-même. Par conséquent, la valeur du hachage représente le message.

Les fonctions de hachage sont efficaces sur le plan calculatoire. Ils permettent de retrouver à la sortie une emprunte de longueur fixe, indépendamment de la longueur du message à l'entrée. L'emprunte déterminée est très sensible aux bits à l'entrée ; cela signifie que même si des modifications mineures sont apportées au message initial, les empruntes générées sont totalement différentes. Toutes ces propriétés sont illustrées sur la figure ci-dessous :

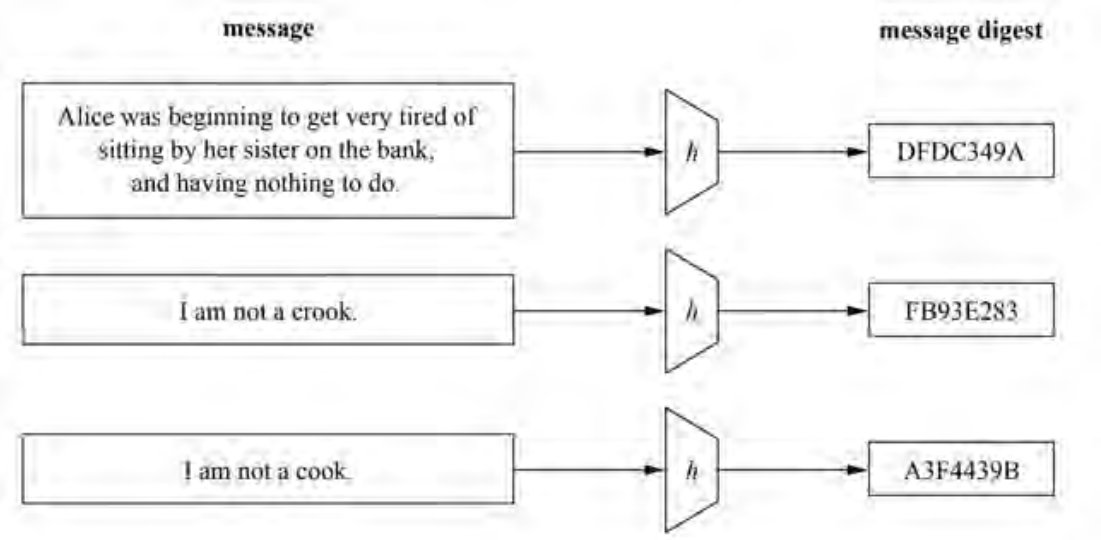

Figure 5: Comportement d'entrée-sortie des fonctions de hachage

Propriétés des fonctions de hachage

1. La taille arbitraire du message : $h(x)$ peut être appliquée à des messages x de toute taille.

2. Longueur de sortie fixe : $h(x)$ produit une valeur de hachage z de longueur fixe.

3. L'efficacité : $h(x)$ est relativement facile à calculer.

4. Résistance à la pré-image : Pour une sortie donnée z, il est calculatoirement difficile de trouver une entrée x telle que $h(x) = z$, c'est-à-dire que $h(x)$ est à sens unique.

5. Résistance à la deuxième préimage : Étant donné x_1, et donc $h(x_1)$, il est calculatoirement difficile, d'un point de vue informatique, de trouver un x_2 tel que $h(x_1) = h(x_2)$.

6. Résistance à la collision : Il est calculatoirement difficile, d'un point de vue informatique, de trouver une paire quelconque $x_1 \neq x_2$ telle que $h(x_1) = h(x_2)$.

Il existe deux types généraux de fonctions de hachage :

- Fonctions de hachage dédiées : Il s'agit d'algorithmes spécifiquement conçus pour servir de fonctions de hachage.

Ex : MD4, MD5, SHA256, SHA512, SHA384

- Fonctions de hachage basées sur un chiffrement par blocs : Il est également possible d'utiliser des chiffrements par blocs tels qu'AES pour construire des fonctions de hachage.

Ex : Oracle Password haching algorithme basé sur 3DES, LANMAN basé sur DES

2.6. Certificat et PKI

Grâce à la cryptographie asymétrique, nous sommes capables de communiquer de manière sécurisée avec les personnes dont nous détenons les clés publiques. Cependant, un certain nombre de problèmes restent non résolus. Par exemple, comment communiquer avec des personnes que nous n'avons jamais rencontrées ? Comment stocker et révoquer des clés publiques ? Mais surtout comment le faire à une échelle planétaire avec des milliards d'individus et des millions de serveurs. Une multitude de problèmes que les public-key infrastructure (PKI) tentent d'adresser. Le rôle d'une PKI est de permettre une communication sécurisée entre deux entités qui ne se sont jamais « rencontrées ». Le modèle utilisé aujourd'hui repose sur des tiers de confiance appelé Autorité de Certification (AC, ou CA pour Certificate Authority) qui génèrent des certificats de confiance et fournissent un mécanisme de vérification de la validité de ces derniers.

Certificats

Un certificat est un document numérique qui contient une clé publique, des informations sur l'entité associée à cette dernière et une signature numérique de l'autorité de certification.
Un certificat consiste en plusieurs champs qui sont principalement :

• Version : Il y a 3 versions de certificat supportant chacun plus ou moins de champs d'information.

• Numéro de série : Un numéro de série qui identifie de manière unique un certificat délivré par une autorité de certification donnée.

• Algorithme de signature : Spécifie l'algorithme de certification.

• Issuer : contient le 'distinguished name' (DN) de l'autorité de certification.

• Validity : Spécifie la date de début et la date de fin de validité du certificat.

• Subject : Il s'agit du DN de l'entité associé à la clé publique pour laquelle le certificat est générer.

• Public key : contient la clé publique sous un format de représentation spécifique (RFC 3279)

D'autres champs sont disponibles comme extension introduit avec la version 3 qui ajoute de la flexibilité au format de certificat "rigide" précédent.
Le standard X.509 utilisé dans les communications internet par exemple dans S/MIME, IPsec et SSL/TLS utilise les champs standards selon la structure ci-dessous :

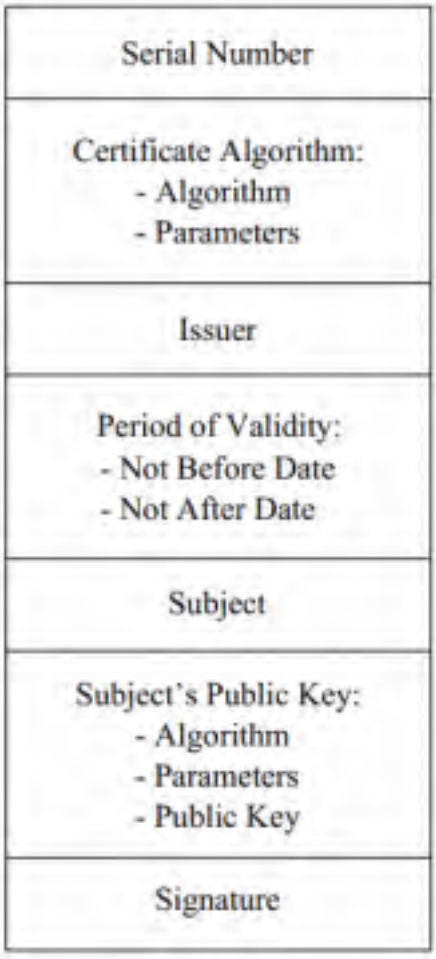

Figure 6: Structure des certificats X.509

Chaîne d'autorité de certification

Dans un monde idéal, il n'y aurait qu'une seule autorité de certification qui fournirait des certificats pour le monde entier. Cependant, ce n'est pas le cas. La plupart des pays ont leurs

propres CA national "officiel". Les sites web quant à eux ont des certificats provenant de plus de 50 CAs différents et les entreprises peuvent générer différents certificats pour leurs employés agissant ainsi comme des CAs. Il est donc virtuellement impossible pour un utilisateur de vérifier la validité des certificats pour tous les CAs. Pour résoudre ce problème les CAs se certifient les uns les autres et créent ainsi une chaîne de CAs.

Dans la pratique, les CAs peuvent être rangés par ordre hiérarchique, dans lequel chaque CA signe la clé publique de l'autorité de certification se trouvant en dessous d'elle. Alternativement, les CAs peuvent se certifier mutuellement sans nécessairement dégager une hiérarchie.

Chapitre 3 : Présentation de la technologie Blockchain

Nous allons dans ce chapitre définir les concepts nécessaires à la compréhension de la technologie blockchain.

3.1. Qu'est-ce que la blockchain

Définir la blockchain en quelques mots n'est pas chose facile. Voici plusieurs définitions qui, *crescendo*, devraient permettre de mieux comprendre ce qu'est la blockchain :

- Généraliste : une blockchain est une technologie pour une nouvelle génération d'applications transactionnelles qui, grâce à un mécanisme de consensus collectif couplé avec l'utilisation d'un grand livre de compte public, décentralisé et partagé, établit la confiance, la responsabilité et la transparence tout en rationalisant les processus d'affaires.
- Technique : une blockchain est une nouvelle technologie de base de données s'appuyant et tirant pleinement profit d'Internet, du protocole libre, de la puissance de calcul et de la cryptographie. Cette base de données transactionnelle distribuée est comparable à un grand livre comptable (registre ou *ledger*) dans lequel chaque nouvelle transaction est écrite à la suite des autres, sans avoir la possibilité de modifier ou d'effacer les précédentes. Ce registre est actif, chronologique, distribué, vérifiable et protégé contre la falsification par un système de confiance répartie (consensus) entre les membres ou participants (nœuds).

On peut aussi proposer cette définition, qui résume l'ensemble des précédentes : une blockchain est une base de données transactionnelle distribuée, comparable à un grand livre comptable décentralisé et partagé, qui stocke et transfère de la valeur ou des données *via* Internet, de façon transparente, sécurisée, et autonome car sans organe central de contrôle. Ce registre est actif, chronologique, distribué, vérifiable et protégé contre la falsification par un système de confiance répartie (consensus) entre les membres ou participants (nœuds). Chaque membre du réseau possède une copie à jour du grand livre (en temps quasi réel) et le contenu est toujours en phase avec l'ensemble des participants.

Ainsi, la blockchain :

- permet l'automatisation de la transaction en supprimant les tiers ;
- est un système de consensus distribué et de confiance partagée ;
- est une infrastructure de certification et de notarisation.

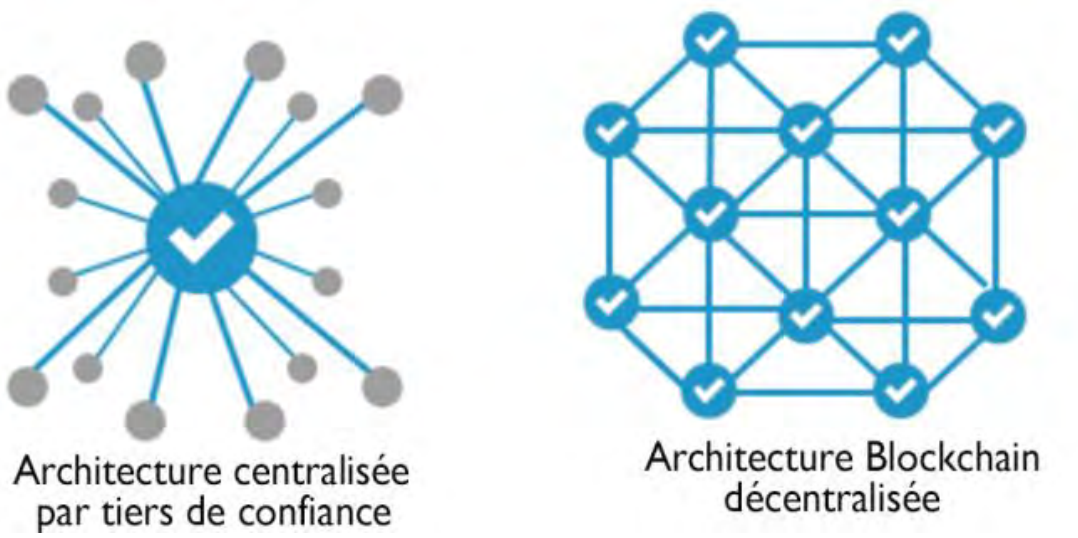

Figure 7 : Architecture centralisée vs architecture décentralisée

3.2. Principe de base et fonctionnement

Toute blockchain publique fonctionne nécessairement avec une monnaie ou un token (jeton) programmable. Bitcoin est un exemple de monnaie programmable.

Les transactions effectuées entre les utilisateurs du réseau sont regroupées par blocs. Chaque bloc est distribué puis validé par les nœuds du réseau appelés les 'mineurs', selon des techniques qui dépendent du type de blockchain. Par exemple, dans la blockchain du bitcoin cette technique est appelée le 'Proof-of-Work', preuve de travail, et consiste en la résolution de problèmes algorithmiques.

Une fois validé, le bloc est horodaté et ajouté à la suite de la chaîne de blocs. La transaction est alors visible pour le récepteur ainsi que l'ensemble du réseau.

La technologie de la blockchain est un système décentralisé et distribué, elle nécessite un réseau d'ordinateurs individuels. Un réseau de blockchain est essentiellement un réseau

d'ordinateurs ou de nœuds sur Internet qui sont incités à vérifier et à stocker les enregistrements des transactions dans un bloc, à approuver et à ajouter le bloc dans la blockchain et à stocker la copie actuelle de la blockchain. Un bloc ne peut être créé que lorsque l'ordinateur ou le nœud du réseau de la chaîne de bloc a accompli la preuve du travail qui consiste à résoudre un puzzle ou un problème cryptographique. Ensuite, ce nœud diffuse le bloc nouvellement créé aux autres nœuds du réseau. Lorsque les autres nœuds du réseau constatent que les transactions du bloc sont valables, ils ajoutent le bloc à leur copie de la blockchain.

La figure suivante illustre simplement le fonctionnement de la blockchain.

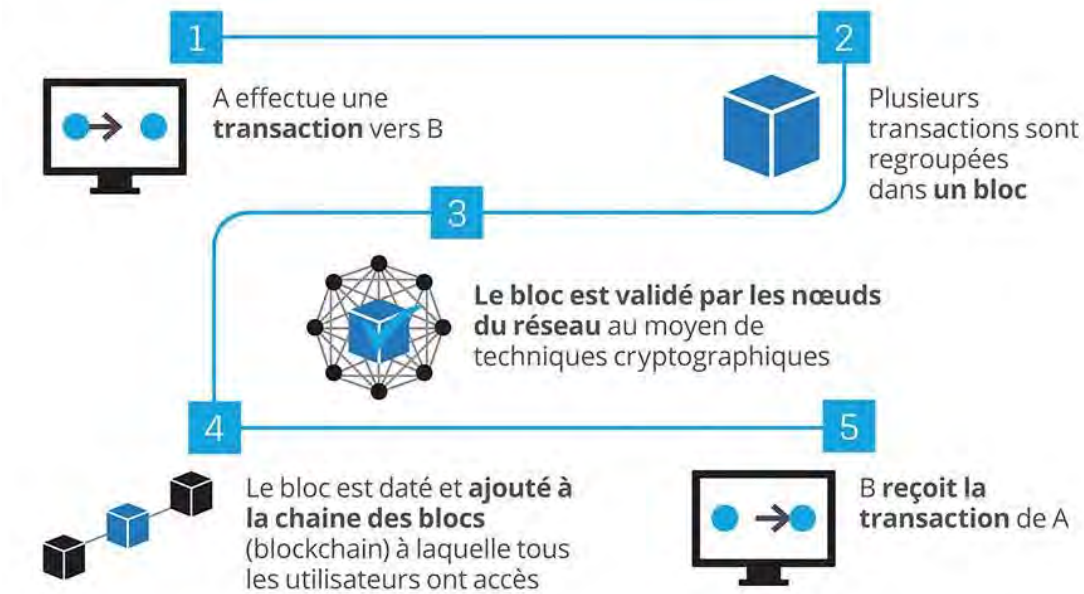

Figure 8: Fonctionnement de la blockchain

Les blocs sont enchaînés ou se développent chronologiquement et cryptographiquement. La valeur de hachage de l'en-tête du bloc précédent est stockée dans le bloc actuel avec les transactions récentes. Et la valeur de hachage de l'en-tête de ce bloc sera stockée dans le bloc suivant avec les transactions à venir.

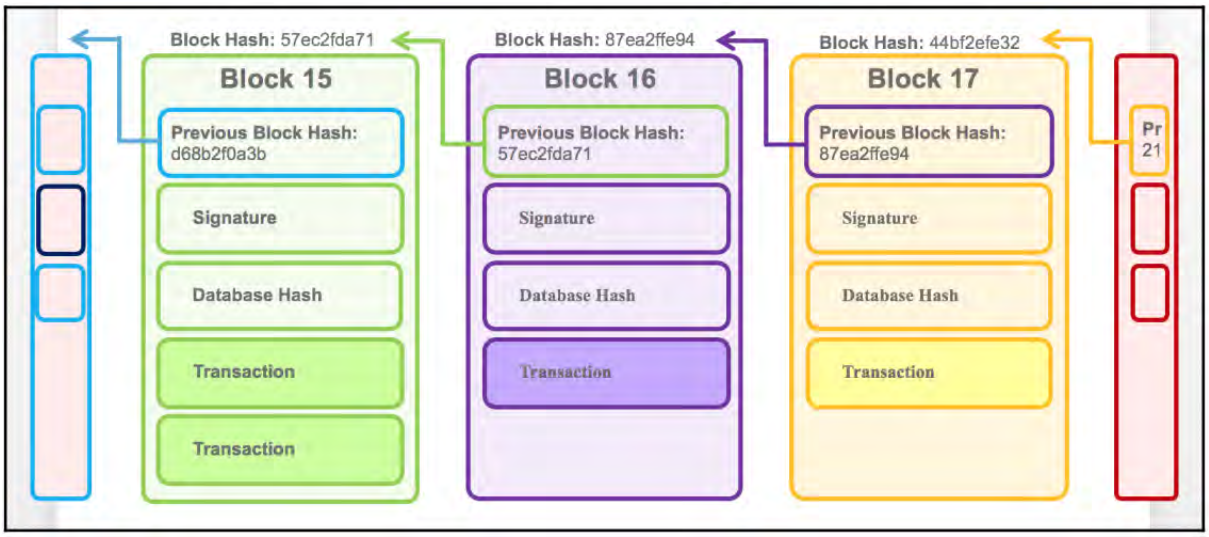

Figure 9 : Structure d'un bloc

Une copie de cette blockchain est stockée dans un système distribué sur le réseau plutôt que dans un serveur ou un ordinateur. Comme il est pratiquement impossible d'inverser le hachage cryptographique et de dépasser la puissance de calcul des nœuds ou ordinateurs honnêtes existants dans le réseau qui créent ces blocs, il est donc pratiquement impossible de modifier tout enregistrement ou transaction stocké dans la blockchain. Nous pouvons donc vérifier les transactions par calcul et les stocker en toute sécurité dans des systèmes répartis sur le réseau, ce qui signifie que nous n'avons pas besoin de faire confiance ou d'utiliser un intermédiaire pour vérifier et stocker nos transactions.

Il existe de nombreux domaines d'application tels que la gestion des contrats de mariage, des votes électroniques ou encore l'envoie d'argent à l'autre bout du monde. Dans notre cas, il s'agira de se baser sur la blockchain pour mettre à disposition une plate-forme qui facilitera la vérification de l'authenticité des diplômes.

3.3. Les propriétés

Les principes sur lesquels est fondée la blockchain sont les suivants :

- ***Décentralisation***

Dans la technologie de la blockchain, il n'y a pas de stockage centralisé des données ni d'autorité centrale sur la gestion des données. Dans le stockage traditionnel des données, il y a un serveur de données et des personnes qui ont l'autorité d'accéder aux données pour les manipuler. Dans la technologie de la blockchain, une copie du registre ou de la base de données est stockée dans tous les ordinateurs du réseau de la blockchain. Si quelqu'un détruit de quelque manière que ce soit un ordinateur, des milliers d'autres ordinateurs du réseau possèdent la copie de la blockchain. Si quelqu'un réussit à modifier des données dans un bloc quelconque de la plus longue chaîne de blocs, ce qui est pratiquement impossible, les autres ordinateurs du réseau compareront leur copie de la blockchain avec celle qui a été modifiée. Si elle ne correspond pas à la copie de la blockchain de la majorité ou de la plupart des participants au réseau, le réseau de blockchain n'acceptera pas de prendre la copie modifiée de la blockchain, et la copie modifiée sera donc perdue du réseau.

- ***Transparence***

La transparence de l'information est une demande croissante, cependant avec notre système économique et numérique actuel, ce n'est pas tout à fait possible. Mais avec la technologie des blockchains, il est possible de créer un stockage de données décentralisé hautement transparent. Toute transaction entre deux utilisateurs qui est stockée sur la blockchain peut être visible par tous les utilisateurs, bien que les utilisateurs puissent être anonymes s'ils ne partagent pas leurs clés publiques. Toute personne ayant accès à la blockchain est en mesure de voir les données et leur historique. Nous savons comment fonctionne Google doc. Chaque participant peut voir qui a fait quoi et à quel moment. De même, dans la blockchain, tous les participants au réseau peuvent voir toutes les modifications apportées aux données. La blockchain est constamment mise à jour et chaque participant du réseau a accès à la blockchain valide.

- ***Immuabilité***

Une fois que les données sont enregistrées, elles sont immuables. Si un seul nœud ou ordinateur le souhaite, il ne modifiera pas les données de la blockchain, à moins que 51 % des nœuds ou des ordinateurs du réseau de la blockchain ne le souhaitent. Mais dans la pratique, comme le réseau de la blockchain est constitué d'un grand nombre de nœuds,

il n'est pas possible de faire changer d'avis un grand nombre de nœuds pour effectuer des tâches malhonnêtes.

- ***Haute disponibilité***

  Les données qui sont stockées ont un degré de disponibilité très élevé par rapport aux technologies traditionnelles. Parce que la copie de la blockchain est stockée dans des milliers de nœuds à travers le monde. Chaque nœud du réseau travaille pour la sécurité et l'intégrité de la blockchain. Même si tous les nœuds d'un lieu géographique donné ont perdu leur copie de la blockchain, les autres nœuds d'un autre lieu auront toujours la copie de la blockchain.

- ***Consensus***

  Les protocoles de consensus créent un système d'accord irréfutable entre différentes parties au sein d'un réseau blockchain, tout en empêchant l'exploitation malveillante du système. Ils permettent de garantir la synchronisation entre tous les nœuds du réseau. De manière fonctionnelle, tout le processus de validation découle d'un consensus atteint par l'ensemble des validateurs de la blockchain avant l'insertion de la transaction dans un bloc.

3.4. Les types de consensus

Les blockchains sont de puissants outils car elles créent des systèmes de confiance qui se corrigent automatiquement sans avoir besoin d'un tiers pour appliquer les règles. Elles accomplissent l'application des règles par leur algorithme de consensus.

Le consensus consiste à élaborer un accord au sein d'un groupe d'actionnaires généralement méfiants. Ce sont les nœuds pleins sur le réseau. Ils valident les transactions qui sont entrées dans le réseau pour être enregistrées dans le registre.

Le mécanisme de consensus a été introduit afin de résoudre le problème de la double dépense sans passez par une autorité centrale. La double-dépense désigne le fait que les mêmes

unités d'une cryptomonnaie pourraient être dépensées deux fois, il est donc crucial d'éliminer ce risque sur le plan numérique. Le problème de la double dépense peut se résumer ainsi :

Prenons le cas où la blockchain permet de gérer une crypto monnaie. Supposons qu'Oscar possède 5 pièces de cette monnaie. Oscar initie une transaction pour transférer ses 5 pièces à Alice et initie à la suite une autre transaction pour transférer les 5 pièces à Bob en même temps. Alors ces transactions vont se retrouver avec l'ensemble des transactions non validées. On distinguera deux cas de figures :

1. Une transaction, par exemple celle qui vers Alice, est prise en compte en premier et est validé. La transaction vers Bob sera donc rejetée

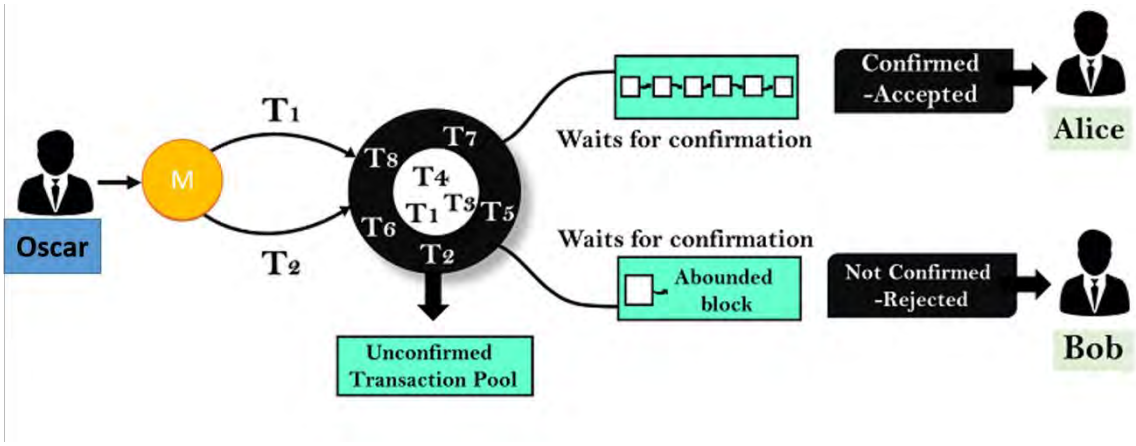

Figure 10 : Double Dépense

2. Les deux transactions sont prises en charge en même temps et ils reçoivent la validation de la transaction en même temps.

Dans ces deux cas seul un mécanisme de consensus nous permet de définir la démarche à suivre par les membres du réseau pour la suite du processus.

Il existe de nombreux modèles pour créer un consensus. Parmi eux les plus connus sont :

- ***Le Proof-of-Work (PoW)***

  Le protocole de Proof-of-Work est le plus utilisé de tous les consensus blockchain. Depuis 2009, il a pu démontrer sa résistance et sa sécurité aux différentes tentatives d'attaques.

Dans le protocole de Proof-of-Work, les différents nœuds du réseau sont appelés mineurs. Pour confirmer une transaction, les mineurs doivent résoudre un problème mathématique complexe réclamant une puissance de calcul importante.

Pour cela, ils utilisent un procédé mathématique appelé fonction de hachage. Le hachage permet d'inscrire les données de transaction dans les blocs et de les raccorder entre eux. Il en existe différents types, comme le SHA 256, utilisé sur Bitcoin. Une fois le hash inscrit dans la blockchain, celui-ci est infalsifiable.

Un mineur est récompensé pour chaque bloc qu'il parvient à approuver et confirmer. Ses revenus sont proportionnels à la puissance de calcul qu'il est à même de déployer pour répondre au problème.

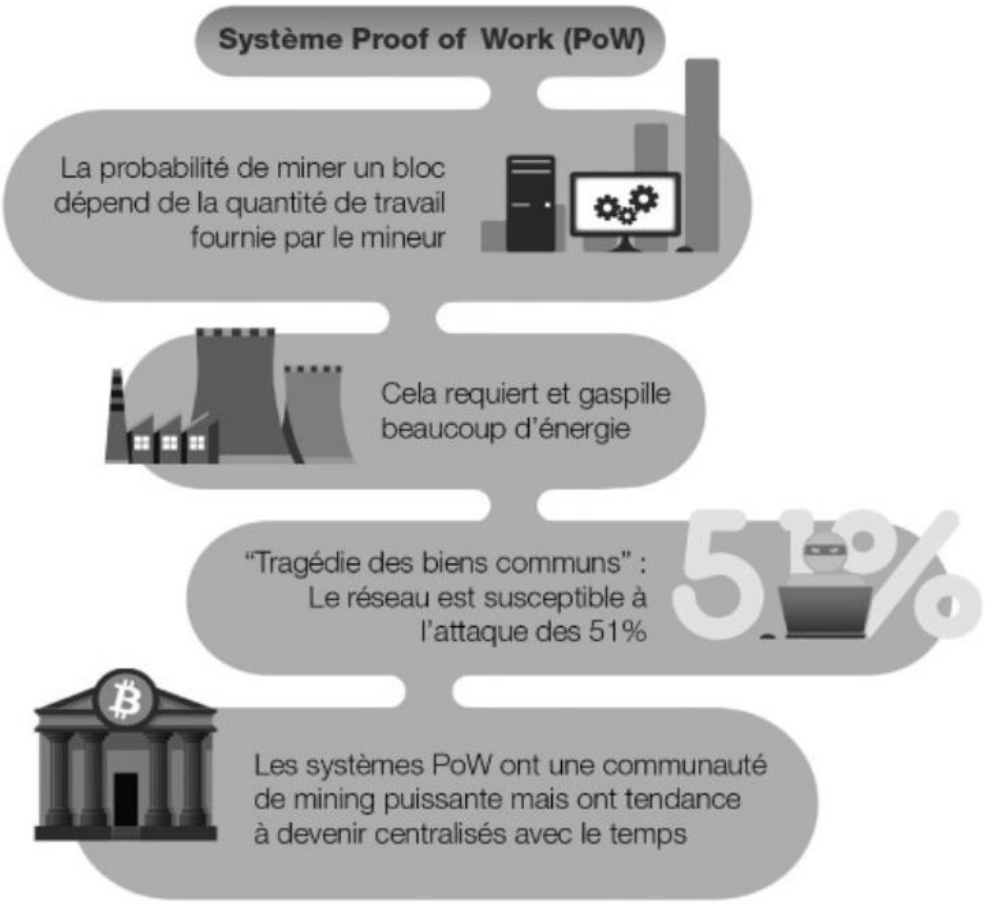

Figure 11 : Le système Proof of Work

- ***Le Proof-of-Stake (PoS)***

La Proof-of-Stake, ou preuve d'enjeu, repose sur une logique tout à fait différente que la preuve de travail et ne nécessite pas de puissance de calcul particulière. Dans la Proof-of-Stake, les participants du consensus peuvent être assimilés à des actionnaires d'une entité business ayant le privilège de participer à son mécanisme de consensus.

Le PoS est un consensus nettement moins onéreux que le PoW, car elle ne réclame ni dépenses énergétiques, ni matériel particulier. Concrètement, pour valider un bloc, les nœuds doivent ici prouver leur possession d'une certaine quantité de cryptomonnaie, et la mettre en gage sur le réseau. Plus cette quantité est importante, plus un nœud aura de chances d'être choisi pour mettre à jour le registre d'une blockchain. Le consensus de PoS considère en effet que ces personnes sont les plus susceptibles de vouloir lutter contre une attaque du réseau, qui pourrait entièrement les ruiner.

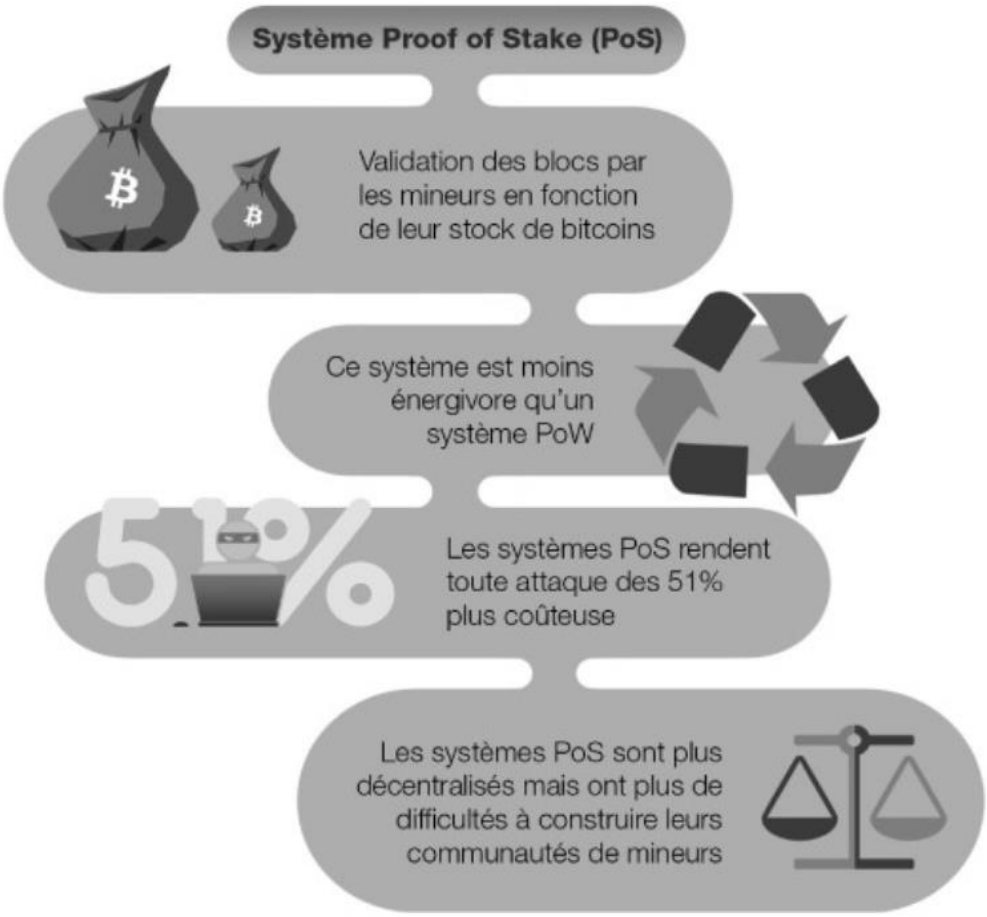

Figure 12 : Le système Proof of Stake

Tableau 1 : Comparatif des types de consensus

Critères	Preuve de travail	Preuve d'enjeu	Forme hybride entre preuve de travail et preuve d'enjeu
Consommation énergétique	Très importante	Faible	Très importante
Besoin de matériel informatique spécialisé	Très important	Pas nécessaire	Important
Risque de séparation du réseau (*forking*)	Possible, lorsque deux nœuds trouvent le bon hash au même moment	Très improbable	Probable
Vulnérabilité aux attaques des 51%	Existante	Faible	Existante, mais moins que pour la preuve de travail simple
Vitesse de création des blocs	Lente, dépend de plusieurs variables	Rapide	Lente, dépend de plusieurs variables
Risque de regroupement en *pools*	Oui, mais peut être prévenu	Oui, mais difficile à prévenir	Oui
Exemples	Bitcoin	Nextcoin	PPcoin, Blackcoin

3.5. Smart Contrat

Un smart contrat ou contrat intelligent est une application ou un programme informatique qui gère l'exécution des termes d'un contrat lorsque les conditions sont remplies. Lorsqu'un smart contrat fonctionne sur une blockchain, il fonctionne exactement comme prévu sans possibilité de fraude ou d'interférence d'un tiers et sans temps d'arrêt. Dans un contrat normal, il est nécessaire de disposer d'un cadre juridique ou de s'appuyer sur la confiance pour s'assurer que chaque partie au contrat se comportera conformément aux termes. Une partie intermédiaire peut provoquer un comportement trompeur accidentel ou intentionnel pour exécuter le contrat. Dans un smart contrat, les clauses contractuelles sont transformées en script informatique, ce script s'exécute automatiquement lorsque les conditions des contrats sont remplies. Si les conditions ne sont pas remplies, le smart contrat ne permettra pas de se comporter ou de faire des transactions en dehors de l'accord. Par conséquent, le smart contrat réduit le besoin d'un intermédiaire de confiance pour exécuter le processus ou les tâches programmées dans celui-ci.

Le smart contrat peut être utilisé dans différentes situations comme le contrat de séquestre, le vote, le contrat de travail, la vente aux enchères où un intermédiaire distribue les actifs sur la base des termes du contrat. Par exemple, un agent fiduciaire, en tant que tiers, détient un actif au nom de deux autres parties. Lorsque deux parties remplissent les conditions

convenues dans le contrat, l'agent fiduciaire transfère l'actif d'une partie à l'autre. Nous devons donc faire confiance à l'agent fiduciaire pour qu'il fasse son travail. Un smart contrat peut faire la même chose qu'un agent fiduciaire. Le smart contrat retient la transaction jusqu'à ce que les exigences respectées par les deux parties soient satisfaites. La principale différence est que la preuve de la transaction et du respect des exigences doit être activée numériquement ou sur la plate-forme numérique. Comme le smart contrat est dans un système de blockchain, personne ne peut altérer ou modifier le code ou la règle d'exécution du contrat. Il fera son travail lorsque les exigences contractuelles seront satisfaites. Vous n'avez donc pas besoin de faire confiance au smart contrat, ce qui en fait un système sans confiance.

3.6. Les types de blockchain

Il existe principalement deux types de blockchain : les blockchains privées et les blockchains publiques. Toutefois, il existe également plusieurs variantes, comme les blockchains Consortium et Hybride. Avant d'entrer dans les détails des différents types de blockchains, voyons d'abord quelles sont leurs similitudes. Chaque blockchain consiste en un groupe de nœuds fonctionnant sur un système de réseau peer-to-peer (P2P). Chaque nœud d'un réseau possède une copie de la blockchain qui est mise à jour en temps utile. Chaque nœud peut vérifier les transactions, initier ou recevoir des transactions et créer des blocs.

3.6.1. Blockchain publique

Une blockchain publique est un système de registre distribué, non restrictif et sans autorisation. Toute personne ayant accès à l'internet peut se connecter sur une plate-forme de blockchain publique pour devenir un nœud autorisé et faire partie du réseau de blockchain. Un nœud ou un utilisateur qui fait partie de la blockchain publique est autorisé à accéder aux enregistrements actuels et passés, à vérifier les transactions ou à faire une preuve de travail pour un bloc entrant, et à faire de l'exploration. L'utilisation la plus élémentaire des blockchains publiques est l'extraction et l'échange de cryptocurrencies. Ainsi, les blockchains publiques les plus courantes sont les blockchain Bitcoin et Litecoin. Les blockchains publiques sont

généralement sûres si les utilisateurs respectent strictement les règles et les méthodes de sécurité. Cependant, le risque n'est présent que lorsque les participants ne suivent pas sincèrement les protocoles de sécurité.

Exemple : Bitcoin, Ethereum, Litecoin

3.6.2. Blockchain privée

Une blockchain privée est une blockchain restrictive ou d'autorisation qui ne fonctionne que dans un réseau fermé. Les blockchains privées sont généralement utilisées au sein d'une ou plusieurs organisations dont seuls des membres sélectionnés participent au réseau de blockchain. Le niveau de sécurité, les autorisations, les permissions, l'accessibilité sont entre les mains de l'organisation qui les contrôle. Les blockchains privées sont donc utilisées de la même manière que les blockchains publiques, mais avec un réseau restreint et restrictif. Les réseaux de blockchain privées sont déployés pour le vote, la gestion de la chaîne d'approvisionnement, l'identité numérique, la propriété des actifs, etc.

Les exemples de blockchains privés sont : Corda, Hyperledger (Fabric, Sawtooth), etc.

3.6.3. Blockchain hybride

Une blockchain hybride est une combinaison de la blockchain privée et publique. Elle utilise les caractéristiques des deux types de blockchain, c'est-à-dire que l'on peut avoir un système privé basé sur des autorisations et un système public sans autorisation. Avec un tel réseau hybride, les utilisateurs peuvent contrôler qui a accès à quelles données stockées dans la blockchain. Seule une partie sélectionnée des données ou des enregistrements de la blockchain peut être autorisée à la lecture et être rendue publique, le reste étant considéré comme confidentiel dans le réseau privé. Le système hybride de la blockchain est flexible, de sorte que les utilisateurs peuvent facilement rejoindre une blockchain privée avec plusieurs blockchains publiques. Une transaction dans un réseau privé d'une blockchain hybride est généralement vérifiée au sein de ce réseau. Mais les utilisateurs peuvent également la publier dans la blockchain publique pour la faire vérifier. Les blockchains publiques augmentent le hachage et

impliquent davantage de nœuds pour la vérification. Cela renforce la sécurité et la transparence du réseau de blockchain.

Dragonchain est un exemple de blockchain hybride.

Tableau 2 : Comparatif des types de blockchain

	Blockchain publique	**Consortium**	**Blockchain privée**
Participants	Non-permissionnée Anonyme ou pseudonyme. Les nœuds peuvent être malicieux	Permissionnée Identifié. Digne de confiance.	Permissionnée Identifié. Digne de confiance.
Consensus	Proof of Work, Proof of Stake, ... Consommation d'énergie importante.	Algorithme de vote ou un consensus multipartite Consensus modulable Peu de consommation d'énergie	Algorithme de vote ou un consensus multipartite Consensus modulable Peu de consommation d'énergie
Temps de validation des transactions	Long Bitcoin : 1 transaction validée toutes les 10 mns ou plus	Court 100 x msec	Court 100 x msec

3.7. Bitcoin et Ethereum

Bitcoin et Ethereum sont les deux blockchains ayant des crypto-monnaies les plus connues et utilisées dans le monde. Nous ne pouvons aborder le sujet des blockchains sans présenter brièvement ces deux blockchains.

3.7.1. Bitcoin

Définition

Le terme « bitcoin » provient de la contraction des termes anglais bit, unité d'information binaire, et coin, pièce de monnaie. Bitcoin désigne à la fois un protocole informatique (Bitcoin) à travers le réseau internet et l'unité de compte (bitcoin) utilisée par ce système de paiement. La blockchain Bitcoin est une technologie libre et ouverte qui fonctionne en réseau pair à pair (peer-to-peer ou P2P), sans autorité centrale (sans passer par une institution financière) et qui permet l'échange d'unités (bitcoin ou BTC) tout en enregistrant chaque transaction (horodatage) dans un grand livre de compte (ledger) dans lequel toute modification est impossible. La gestion des transactions et la création de bitcoins sont prises en charge collectivement par le réseau et sa conception est publique ; personne ne possède ni ne contrôle la blockchain Bitcoin et chacun peut s'y joindre. Grâce à plusieurs de ses propriétés uniques, Bitcoin rend possible des usages prometteurs qui ne pourraient pas être couverts par les systèmes de paiement actuels. La devise bitcoin, contrairement aux autres devises monétaires, n'est pas l'incarnation de l'autorité d'un État, d'une banque ou d'une entreprise et chaque bitcoin est identifiable dans un grand livre de compte par un historique de toutes les transactions dans lesquelles il est impliqué depuis sa création.

Fonctionnement

La blockchain Bitcoin repose sur un protocole cryptographique notamment pour :

- D'une part, résoudre le problème dit « de la double dépense », qui avait jusqu'alors empêché l'émergence d'un tel type de monnaie (A donne à B en s'assurant qu'il n'a pas donné à C en parallèle) ;
- D'autre part, garantir l'impossibilité de falsifier les identifiants des parties prenantes et la valeur du stock de bitcoins figurant dans les porte-monnaie électroniques

Le fonctionnement de la blockchain Bitcoin suit quatre étapes :

- Deux personnes s'accordent sur une transaction ;
- Grâce à la blockchain la transaction est encryptée et validée par consensus (proof of work/minage) ;
- Elle est ensuite inscrite puis verrouillée dans le dernier bloc de la blockchain ;

- Enfin la blockchain est répliquée dans tous les nœuds (participants) du réseau.

Impossible de parler de la blockchain sans aborder Ethereum qui est souvent présenté par rapport à Bitcoin comme le Bitcoin 2.0 ou le prochain Bitcoin.

3.7.2. Ethereum

Définition

À l'instar de la blockchain Bitcoin, Ethereum est une blockchain publique dont la particularité est de permettre la création par les utilisateurs de contrats intelligents, grâce à un langage Turingcomplet102, contrats qui sont basés sur un protocole informatique permettant de vérifier ou de mettre en application un contrat mutuel, et qui sont déployés et consultables publiquement dans la blockchain.

Ethereum utilise une unité de compte dénommée ether comme moyen de paiement des contrats. Le sigle correspondant, utilisé par les plateformes d'échange, est ETH. L'ether est la deuxième plus importante monnaie cryptographique décentralisée, après le bitcoin, avec une capitalisation supérieure à 1 milliard d'euros107.

Fonctionnement

On peut considérer Ethereum comme un ordinateur mondial (constitué de milliers d'ordinateurs) à travers le monde, auquel tout le monde peut accéder. Sa puissance de calcul provient des mineurs, qui sont rétribués en « gaz ».

Les mineurs exécutent collectivement les opérations nécessaires (vérification, ajout de données, exécution de smart contracts) au fonctionnement de la blockchain Ethereum en échange de cette rétribution. Le gaz peut être échangé contre des ethers, qui eux-mêmes peuvent être échangés contre des monnaies fiat sur les plateformes de marchés. On peut donc stocker ce que l'on souhaite sur la blockchain Ethereum, même du code. Cette blockchain est à disposition des particuliers comme des professionnels, qui peuvent s'en servir librement.

Ethereum se distingue des autres blockchains par les smart contracts (contrats intelligents) et les DAO (decentralized autonomous organizations, organisations autonomes

décentralisées). Ce système permet la réduction du nombre de contentieux ainsi qu'une gestion plus aisée de ceux-ci. Dans ce système, il n'est pas besoin de faire confiance au partenaire, ni à une autorité centrale. C'est le système informatique totalement automatisé qui garantit l'honnêteté de la transaction.

Tableau 3 : Comparatif Bitcoin & Ethereum

Caractéristiques	Ethereum	Bitcoin
Inventeur	Vitalik Buterin	Satoshi Nakamoto
Date de création	2013	2008
Date de lancement	2015	2009
Type de blockchain	Publique	Publique
Monnaie	Ether (ETH)	bitcoin (BTC)
Valeur d'une unité	1 ETH = 9,4 $	1 BTC = 730 $
Valorisation totale	809 434 701 $	11 687 716 530 $
Monnaie totale en circulation	86 337 860 ETH	16 008 575 BTC
Volume max de monnaie à terme	aucun	21 000 000 BTC
Algorithme de minage	Proof-of-work	Proof-of-work
Algorithme de minage à terme	Proof-of-stake	Proof-of-work
Délai de validation d'un bloc	15 secondes	10 minutes
Montant touché pour un bloc miné	5 ETH	12,5 BTC
Possibilité de Smart Contracts	OUI	NON
Possibilité de Decentralized Autonomous Organization	OUI	NON

3.7.3. Autres applications de la blockchain

Ripple :

C'est le nom d'un réseau de paiement et du protocole utilisé. Développé et publié en 2012 par une société du même nom, afin de permettre des « transactions financières mondiales sécurisées, instantanées et quasiment gratuites », il est construit sur des principes similaires à ceux de la blockchain Bitcoin, donc parfois considéré comme une crypto-monnaie. Mais le code est privé, donc invérifiable par des tiers. De nombreuses banques l'utilisent comme base de leur propre infrastructure de règlement.

Litecoin :

C’est une blockchain tirée de Bitcoin. Ce type de crypto-monnaie est appelé alt-coin. Comme Bitcoin, il utilise l’algorithme de Proof of work pour valider et ordonner les transactions, mais de difficulté différente. Conséquemment, les blocs sont validés plus rapidement, et leur taille est plus importante.

Partie II : Cadre Conceptuel

Chapitre 4 : Spécification et Analyse des besoins

Il est nécessaire, pour commencer, de définir les fonctionnalités qui devront être implémenté dans notre système. Ces fonctionnalités seront représentées à travers des diagrammes de cas d'utilisation et de séquence avec le langage de modélisation UML (Unified Modeling Language). Nous allons à travers ce chapitre présenter les détails des différentes fonctionnalités du système.

4.1. Méthodologie et définitions

La mise en place d'une application est composée de plusieurs activités prenant en compte les aspects d'organisation techniques et humains afin que le produit fini corresponde aux besoins de l'utilisateur.

Les différentes activités qui composent le processus de développement sont :

- L'étude de la faisabilité qui permet d'analyser les besoins des utilisateurs ;
- La spécification des besoins fonctionnels qui consiste à établir dans le détail 'ce que doit faire' l'application du point de vue de l'utilisateur ;
- La conception qui permet de décrire 'comment faire l'application' en élaborant une structure interne du système ;
- L'implémentation qui est la traduction en langage de programmation des concepts qui ont été définis pendant la phase de conception ;
- Les tests qui consistent à détecter les erreurs en effectuant la validation et la vérification de l'implémentation ;
- Le déploiement permettant de mettre à disposition de l'utilisateur le produit final ;
- La maintenance qui permet d'améliorer l'application durant sa phase d'exploitation.

L'ensemble de ces activités est illustré dans la figure ci-dessous :

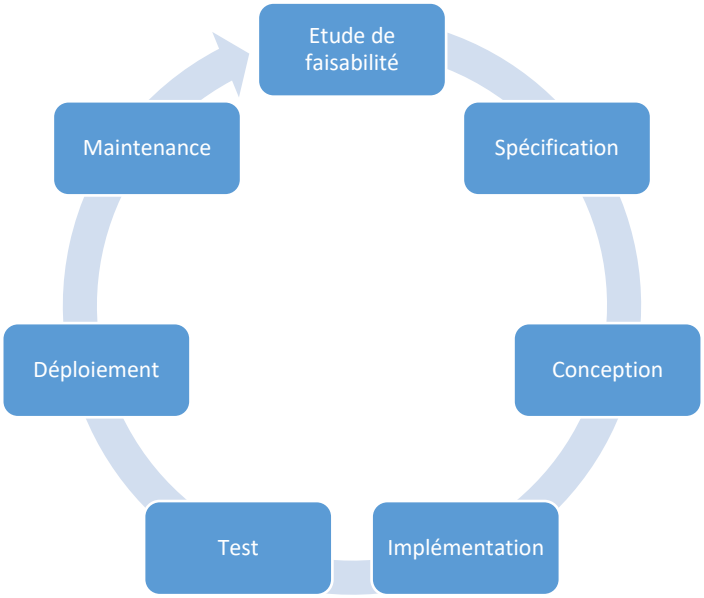

Figure 13 : Processus de développement

L'objectif de ces différentes activités est la réalisation d'une application de qualité afin de satisfaire les exigences de l'utilisateur. Dans notre cas ces activités ont été mis en œuvre afin d'analyser et de concevoir une plate-forme de validation de diplômes qui serait basée sur la blockchain. Pour ce faire nous avons utilisé le langage UML comme méthode de modélisation.

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) est un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie.

Il se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent.

4.2. Analyse de besoins

4.2.1. Besoins fonctionnels

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée/sortie du Système. Notre application doit pouvoir fournir les différentes fonctionnalités recueillies à savoir :

- Module Gestion des diplômes
    1- Insertion des diplômes ;
    2- Modification des diplômes ;
    3- Validation des diplômes ;
    4- Affichage des diplômes ;
    5- Vérification de l'authenticité des diplômes ;

- Module Contrôle d'accès
    1- Identification ;
    2- Gestion des niveaux d'accès en fonction du profil de l'utilisateur connecté ;
    3- Création de compte d'utilisateur ;

4.2.2. Besoins en sécurité

Afin de permettre aux usagers de la plate-forme d'avoir confiance en ses données, notre système doit donc être capable d'assurer un minimum de sécurité à savoir :

• Authentification : il doit pouvoir certifier et attester que l'université qui à délivrer les diplômes est bien celle qu'elle prétend être.

• Non répudiation : Une université ne doit pas pouvoir renier avoir délivrer un diplôme à un tel étudiant et l'avoir publié. Ce besoin est assuré par la signature.

• Intégrité des données : Le système doit être capable assurer que le diplôme présenté est bien authentique, qu'il n'a été ni modifié ni falsifié de quelque façon que ce soit.

4.2.3. Besoins opérationnels

- Haute disponibilité : l'application doit être en mesure de toujours répondre à la requête d'un utilisateur.

- Ergonomie : les fonctionnalités de l'application doivent être compréhensible et facilement manipulable pour l'utilisateur. L'application doit offrir une bonne expérience utilisateur.

4.2.4. Autres Besoins

Il est question ici d'enrichir le système de fonctionnalités qui pourraient répondre aux besoins des utilisateurs sans toutefois tendre vers le superflu. Les besoins qui s'imposent dans ce domaine sont principalement la possibilité, lors de l'ajout d'un diplôme ou d'une transaction, de pouvoir importer un fichier (Excel, PDF) et d'enregistrer directement les informations qu'il contient dans notre Blockchain. On peut ajouter aussi un système de notifications pour plus d'ergonomie.

Nous venons d'exposer l'ensemble des besoins auxquels doit répondre notre système. La sous-section suivante nous permettra de présenter les acteurs de notre système.

4.2.5. Identification des acteurs

Nous allons ici définir les différents acteurs de la plate-forme.

- Acteur : désigne une entité physique humaine ou matérielle qui a la possibilité d'interagir avec le système modélisé.
- Système : fait référence à l'entité à concevoir. Dans notre cas le système est la plate-forme de validation de diplôme.

Nous avons deux principaux acteurs :

- L'administrateur : il désigne la personne responsable de la création et de la publication des diplômes pour l'école. Il a accès à toutes les fonctionnalités de la plate-forme citées ci-dessus.
- L'utilisateur : il désigne la personne désireuse de vérifier l'authenticité d'un diplôme. Il a uniquement accès à la fonctionnalité de vérification.

Le diagramme ci-dessous permet de résumer le contexte statique de notre système

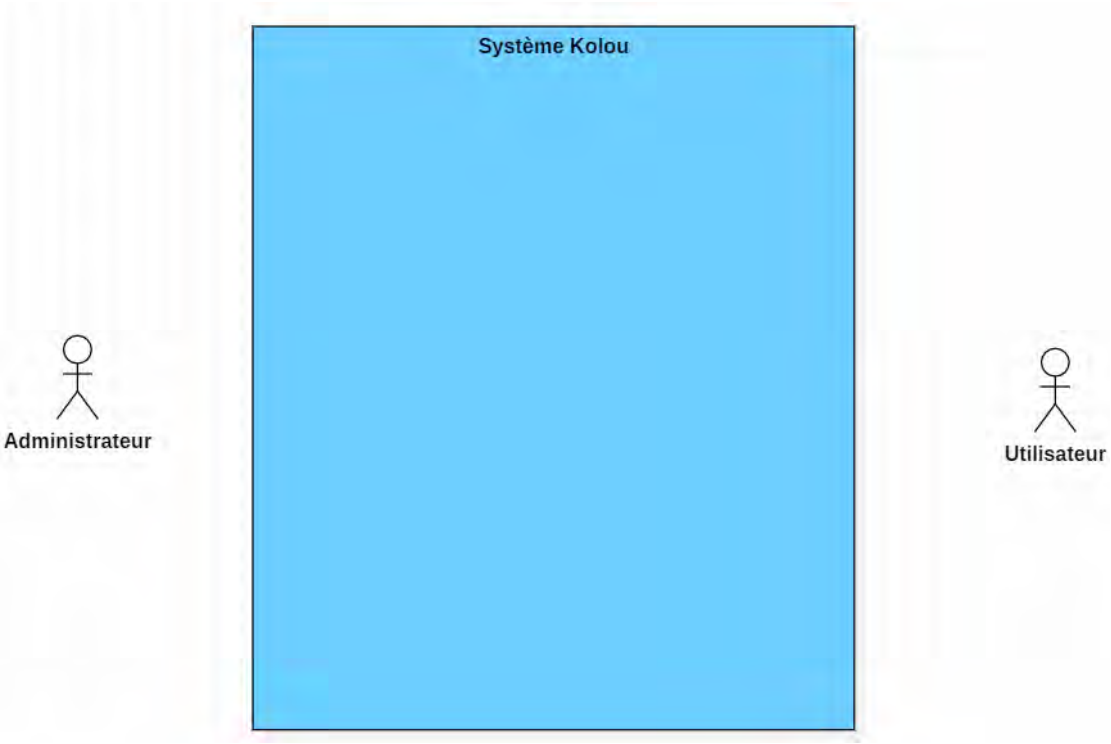

Figure 14 : Diagramme de contexte statique du système

4.3. Spécification des besoins

4.3.1. Comportement fonctionnel du système

Dans cette partie, nous allons exposer le rôle de chaque acteur du système et les fonctionnalités auxquels sont rôle donnent accès. Certaines fonctionnalités ne sont accessibles que si une action est préalablement effectuée. C'est ce que représente la relation d'inclusion. La figure ci-dessous permet de représenter cette vision globale du système.

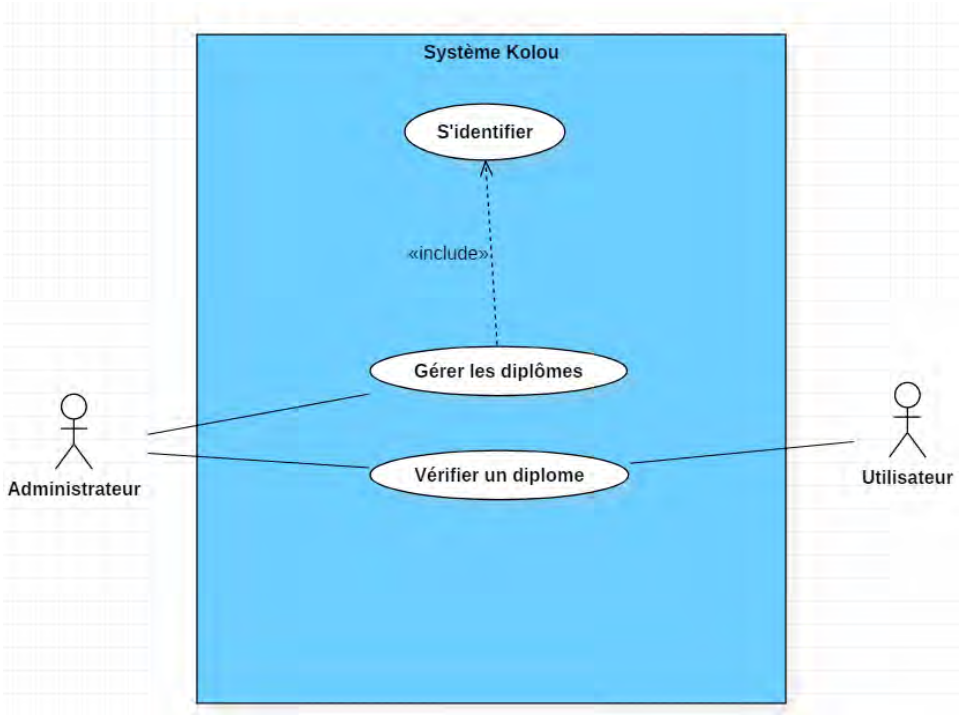

Figure 15 : Vue globale des fonctionnalités du système

- ***S'identifier***

Cette fonctionnalité permet à l'acteur de s'authentifier auprès du système dans le but d'avoir accès aux ressources. Cela permet au système de reconnaitre un utilisateur enregistré et de lui donner accès aux fonctionnalités correspondant à son profil. Certaines fonctionnalités comme ceux du module gestion des diplômes requiert son exécution alors que d'autres non. Le tableau suivant décrit l'étape d'identification.

Tableau 4: Description textuelle de la fonctionnalité S'identifier

Libellé	Description
Nom	S'identifier
Objectif	Ce cas d'utilisation a pour objectif de permettre à un l'utilisateur habilité d'accéder aux ressources de la plate-forme
Acteur Principal	Administrateur

Pré condition	Pour ce cas d'utilisation, l'utilisateur doit accéder à la plate-forme
Scénario nominal	1. L'utilisateur demande l'accès au système 2. La plate-forme affiche la page de connexion 3. L'utilisateur saisi son identifiant et son mot de passe 4. La plate-forme autorise l'accès et affiche la page d'accueil
Scénario alternatif	4.1. La plate-forme refuse l'accès 4.2. La plate-forme affiche un message d'erreur Retour scénario nominal point 2
Post condition	Accéder à la plate-forme

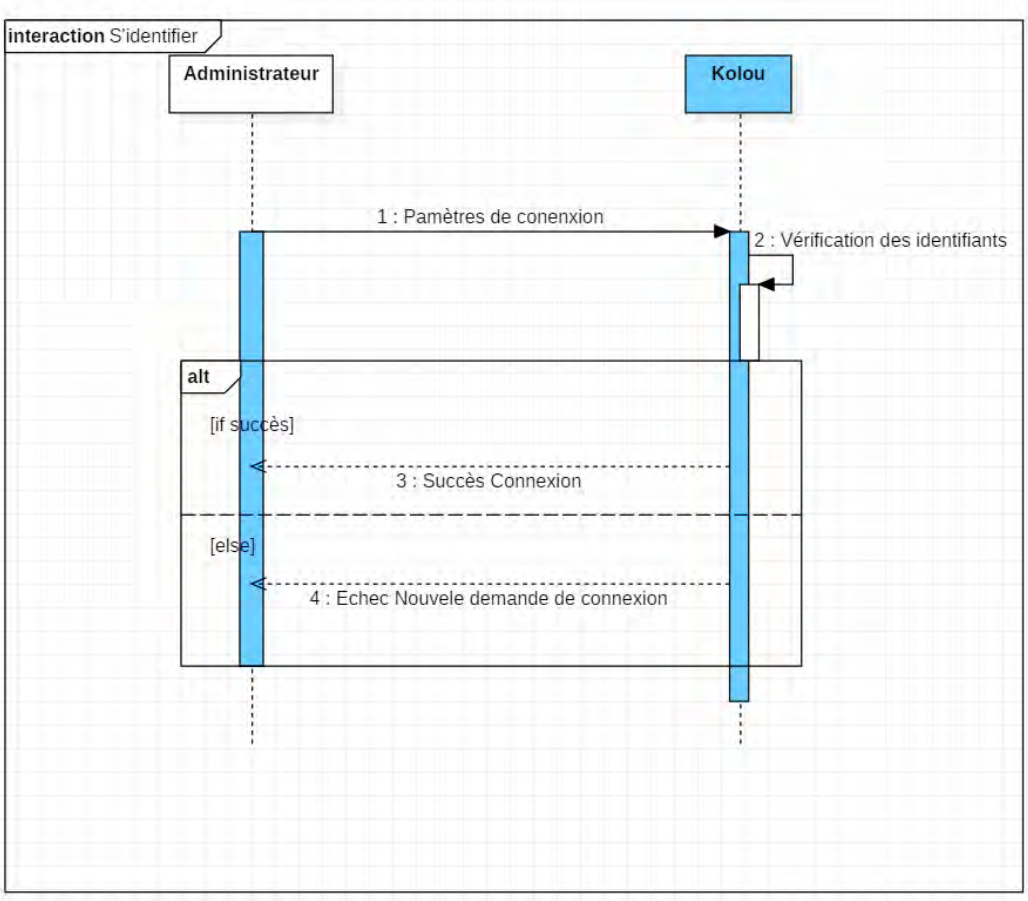

Figure 16 : Diagramme de séquence de l'identification

- ***Gestion des diplômes***

Ce cas d'utilisation est général car il réunit en réalité 3 sous cas d'utilisation à savoir : l'insertion des diplômes, la modification des diplômes et la validation des diplômes. Le diagramme ci-dessous permet de donner un aperçu global sur ce cas d'utilisation. Nous allons par la suite détailler les descriptions de chacun de ces cas d'utilisation.

Tableau 5 : Description textuelle de la fonctionnalité Gérer les diplômes

Libellé	Description
Nom	Gérer les diplômes
Objectif	L'objectif de ce cas d'utilisation est de gérer les diplômes qui seront disponible sur la plate-forme. Gérer les diplômes revient à créer des diplômes sur la plate-forme, modifier un diplôme ou valider un diplôme.
Acteur Principal	Administrateur
Pré condition	- L'utilisateur doit s'être identifié
Scénario nominal	1. L'utilisateur choisi d'ajouter un diplôme 2. L'utilisateur choisi de modifier un diplôme 3. L'utilisateur choisi de valider un diplôme
Scénario alternatif 1	1.1. L'utilisateur renseigne les informations du diplôme à ajouter 1.2. La plate-forme vérifie les informations saisies 1.3. La plate-forme enregistre les informations
Scénario alternatif 2	2.1. L'utilisateur modifie les informations du diplôme à corriger 2.2. La plate-forme vérifie les informations saisies 2.3. La plate-forme enregistre les informations
Scénario alternatif 3	3.1. La plate-forme envoie une demande de confirmation 3.2. L'utilisateur confirme la demande de validation 3.3. La plate-forme valide le(s) diplôme(s) concerné(s)
Post condition	Prise en compte des opérations effectué par l'utilisateur dans le système

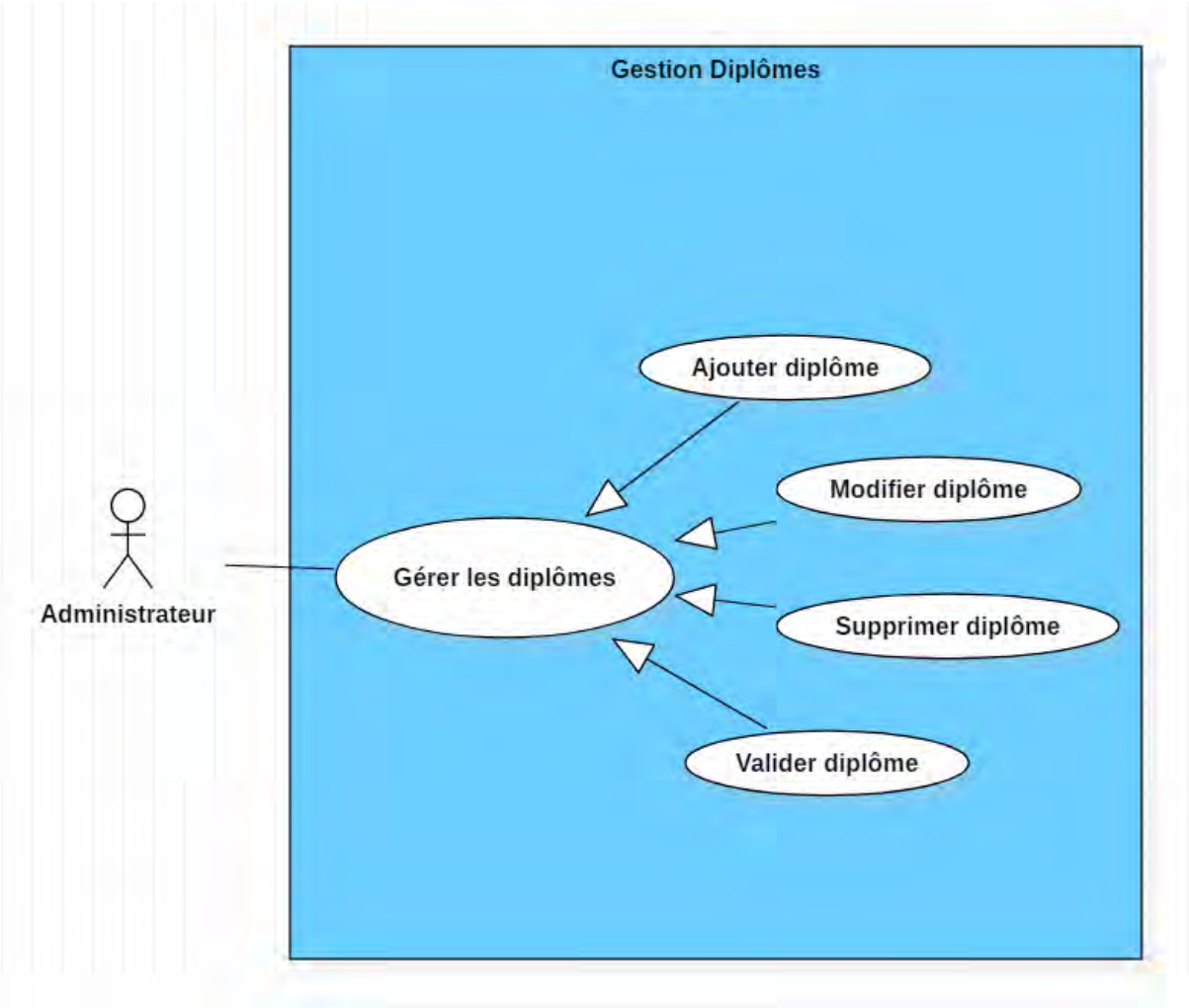

Figure 17 : Diagramme de séquence de la fonctionnalité Gérer les diplômes

- ***Ajouter un diplôme***

Tableau 6 : Description textuelle de la fonctionnalité Ajouter un diplôme

Libellé	Description
Nom	Ajouter un diplôme
Objectif	L'objectif de ce cas d'utilisation est d'insérer un nouveau diplôme encore inexistant dans le système
Acteur Principal	Administrateur
Pré condition	- L'utilisateur doit s'être identifié

	- L'utilisateur doit avoir choisi l'option d'ajout de diplôme dans le système
Scénario nominal	1. L'utilisateur clique sur le bouton ajouter un diplôme 2. La plate-forme affiche le formulaire d'ajout de diplôme 3. L'utilisateur saisi les informations du diplôme qu'il désire ajouter puis valide 4. La plate-forme contrôle les informations saisies 5. La plate-forme enregistre les informations
Scénario alternatif	La plate-forme n'enregistre pas les informations La plate-forme renvoie un message d'erreur Retour au scénario nominal point 2
Post condition	Un nouveau diplôme est créé dans le système

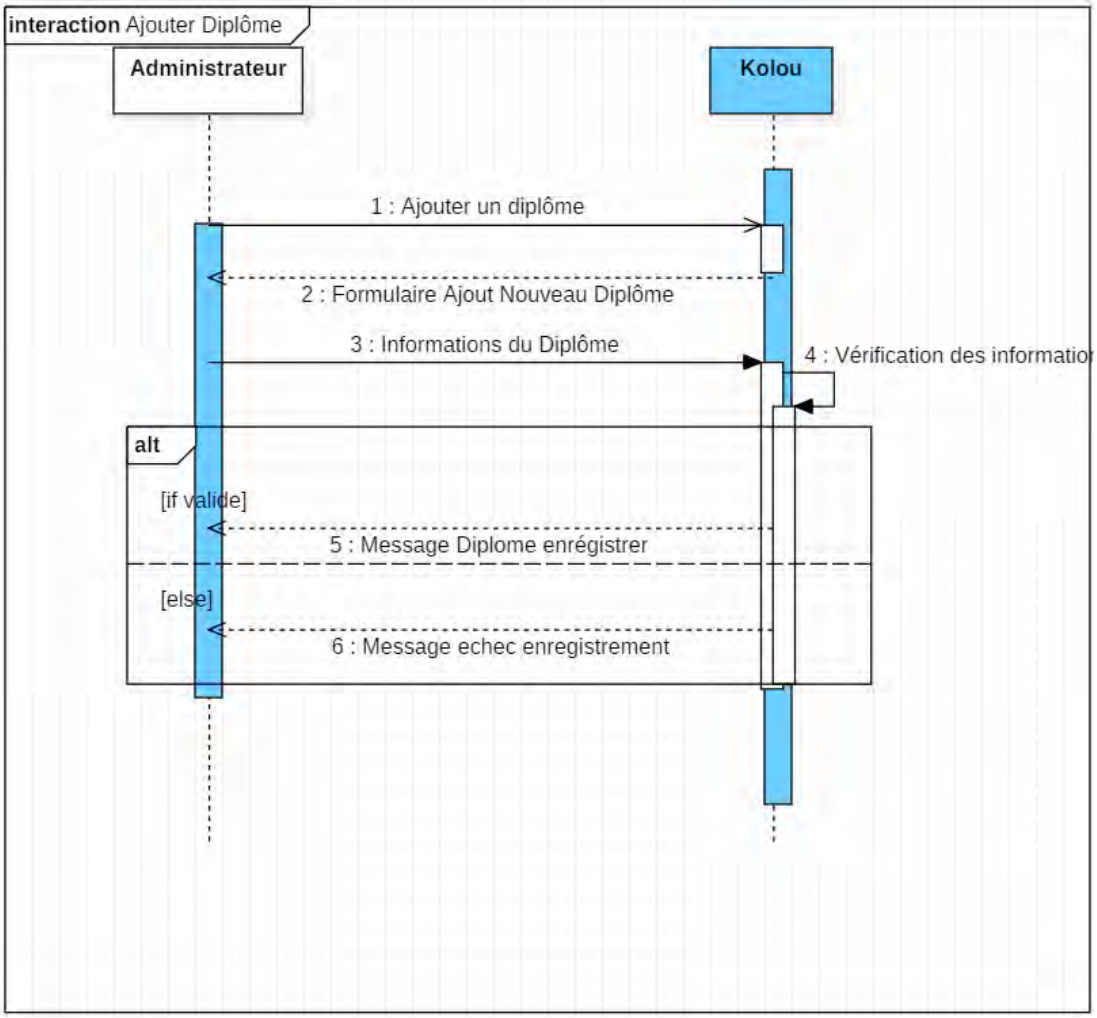

Figure 18 : Diagramme de séquence de la fonctionnalité Ajouter diplôme

- ***Modifier un diplôme***

Tableau 7 : Description textuelle de la fonctionnalité Modifier un diplôme

Libellé	Description
Nom	Modifier un diplôme
Objectif	Ce cas d'utilisation permet à l'utilisateur de modifier les informations concernant un diplôme existant déjà dans le système en cas d'erreur
Acteur Principal	Administrateur
Pré condition	- L'utilisateur doit s'être identifié - L'utilisateur doit avoir choisi l'option de modification d'un diplôme dans le système

Scénario nominal	1. L'utilisateur clique sur le bouton modifier un diplôme 2. La plate-forme affiche le formulaire de modification du diplôme avec les informations par défaut 3. L'utilisateur modifie les informations du diplôme qu'il désire corriger puis valide 4. La plate-forme contrôle les informations saisies 5. La plate-forme enregistre les informations
Scénario alternatif	La plate-forme n'enregistre pas les modifications La plate-forme renvoie un message d'erreur Retour au scénario nominal point 2
Post condition	Correction des informations modifiées

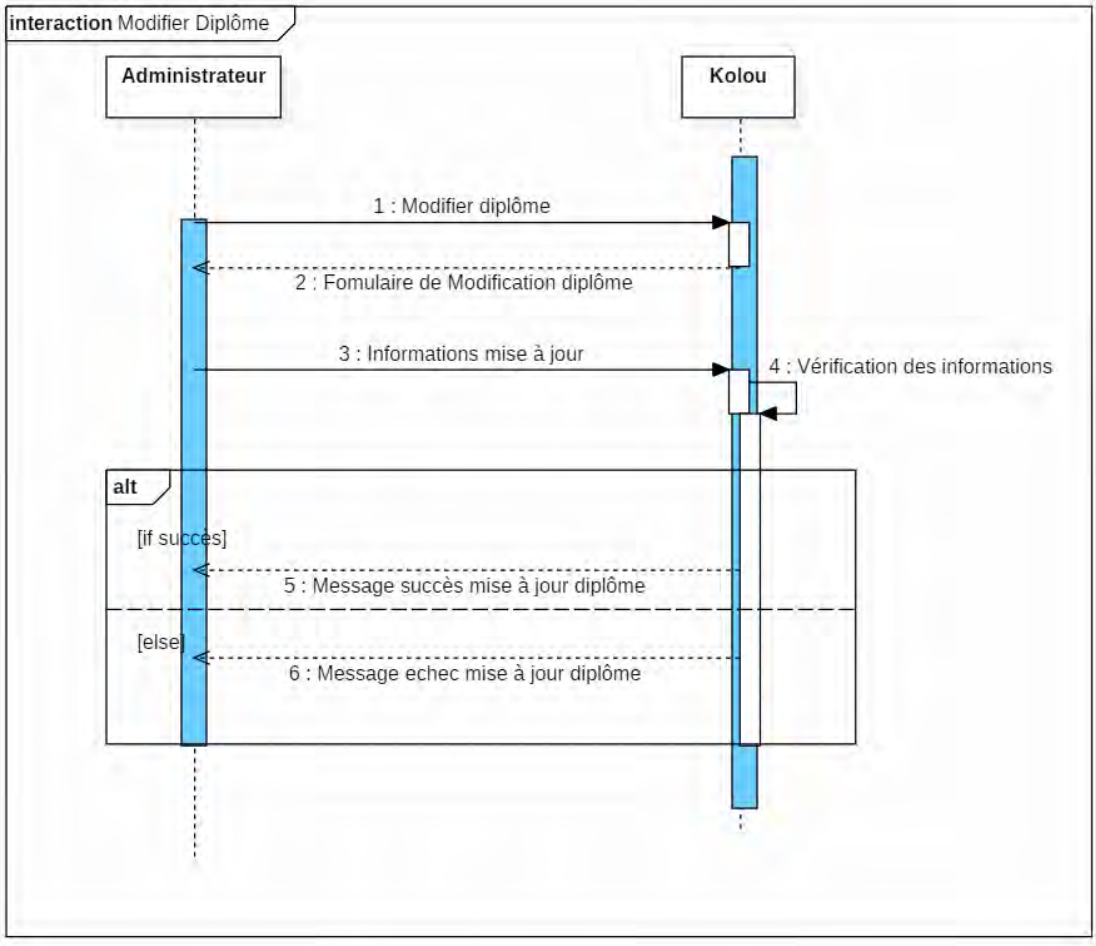

Figure 19 : Diagramme de séquence de la fonctionnalité Modifier diplôme

- ***Valider un diplôme***

Tableau 8 : Description textuelle de la fonctionnalité Valider un diplôme

Libellé	Description
Nom	Valider un diplôme
Objectif	L'objectif de ce cas d'utilisation est de publier le diplôme enregistré dans notre blockchain et de le rendre ainsi disponible et infalsifiable
Acteur Principal	Administrateur
Pré condition	- L'utilisateur doit s'être identifié - L'utilisateur doit avoir choisi l'option de modification d'un diplôme dans le système
Scénario nominal	1. L'utilisateur clique sur le bouton modifier un diplôme 2. La plate-forme affiche le formulaire de modification du diplôme avec les informations par défaut 3. L'utilisateur modifie les informations du diplôme qu'il désire corriger puis valide 4. La plate-forme contrôle les informations saisies 5. La plate-forme enregistre les informations
Scénario alternatif	La plate-forme n'enregistre pas les modifications La plate-forme renvoie un message d'erreur Retour au scénario nominal point 2
Post condition	Validation du diplôme dans le système

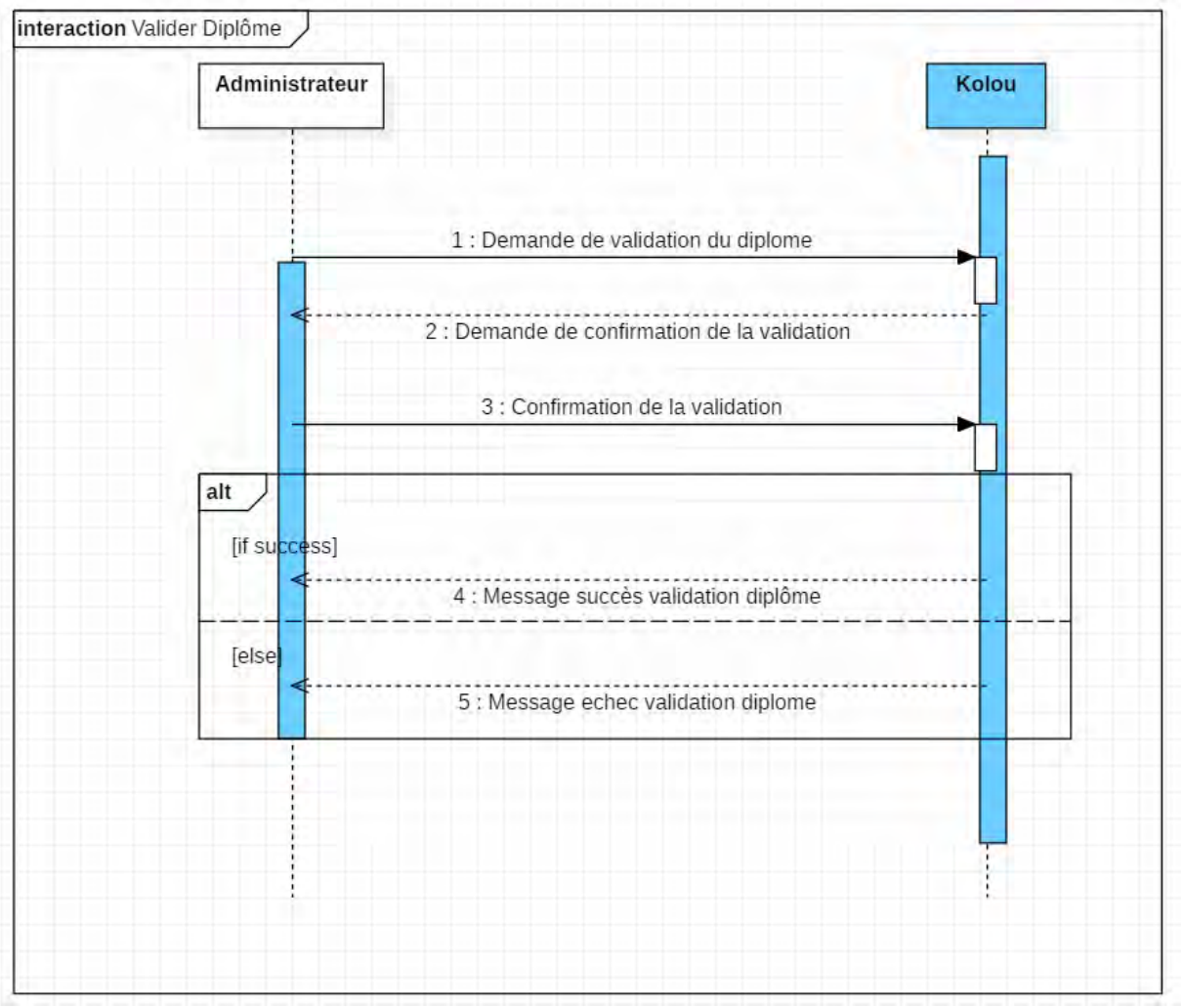

Figure 20 : Diagramme de séquence de la fonctionnalité Valider diplôme

- ***Vérifier un diplôme***

Tableau 9 : Description textuelle de la fonctionnalité Vérifier un diplôme

Libellé	Description
Nom	Vérifier un diplôme
Objectif	L'objectif de ce cas d'utilisation à un utilisateur de vérifier l'authenticité d'un diplôme qui a préalablement été enregistré dans le système

Acteur Principal	Administrateur, utilisateur
Pré condition	- L'utilisateur doit avoir choisi l'option de vérification d'un diplôme au niveau de la plateforme
Scénario nominal	1. L'utilisateur clique sur le bouton vérifier un diplôme puis choisi le mode de vérification 2. La plate-forme affiche le formulaire de vérification de diplôme 3. L'utilisateur insère les informations du diplôme qu'il désire vérifier puis valide 4. La plate-forme vérifie les informations saisies 5. La plate-forme renvoie le résultat résultant du traitement
Scénario alternatif	La plate-forme renvoie un message d'erreur Retour au scénario nominal point 2
Post condition	Vérification de l'existence du diplôme dans le système

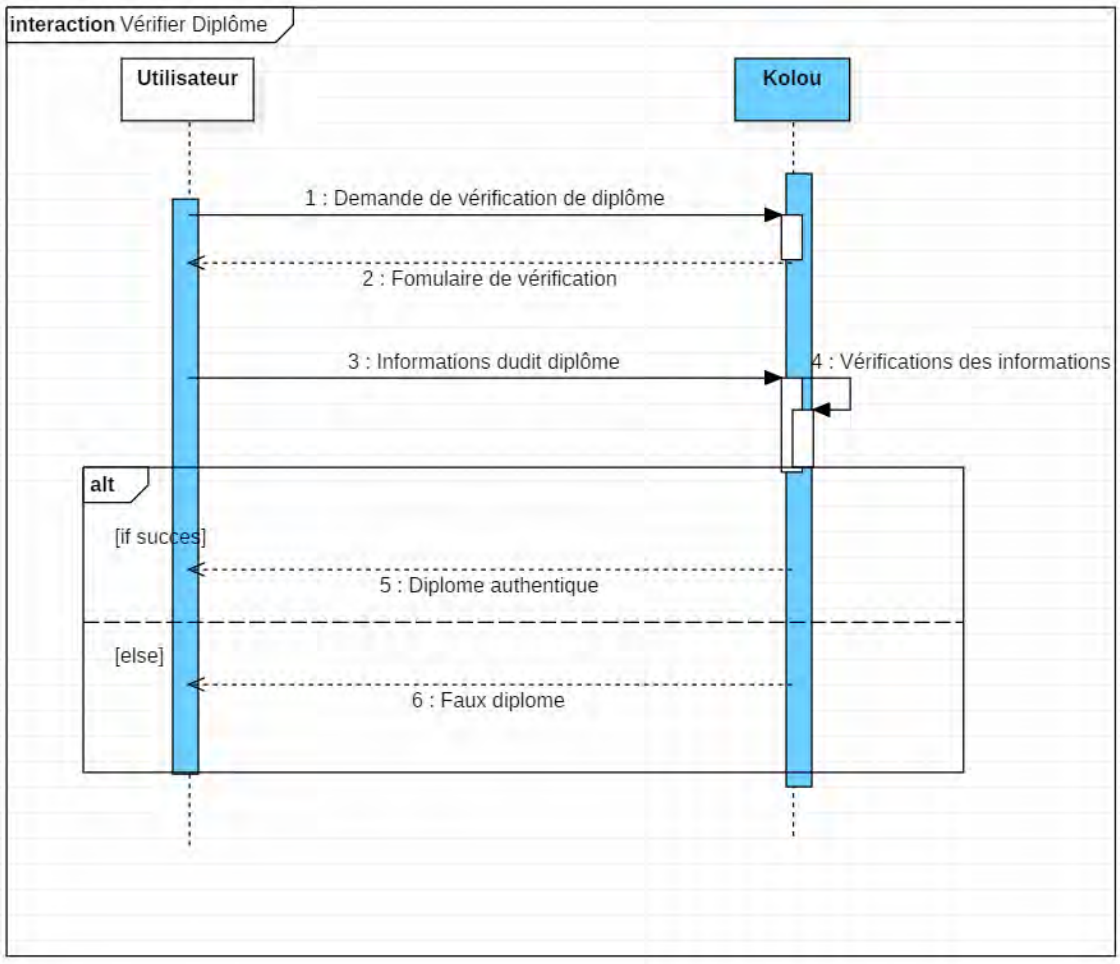

Figure 21 : Diagramme de séquence de la fonctionnalité Vérifier diplôme

4.4. Architecture de la solution proposée

L'architecture logicielle de la solution est composée de quatre entités principales. Il s'agit de l'autorité de certification, du client, de l'API Rest et de la blockchain.

L'autorité de certification permet de délivrer un certificat digital aux universités qui ont à leur charge la publication des diplômes. Ce certificat se compose d'une clé publique, des informations de l'université et de la signature de l'autorité de certification qui l'a délivré.

Le client ici représente la plateforme. C'est à partir de là que l'utilisateur va se connecter et envoyé des requêtes de demande ou d'insertion de données vers la blockchain.

La communication avec la blockchain est rendue possible grâce à l'utilisation d'une API Rest. C'est une interface de programmation d'application qui facilite la création et l'intégration

de logiciel. Elle permet d'interroger un server tiers en utilisant les mêmes méthodes que ceux proposés par l'affichage d'une page web. Elle exploite les méthodes http et https et utilise les requêtes GET, POST, PUT et DELETE.

La figure ci-dessous résume l'architecture logicielle de la solution.

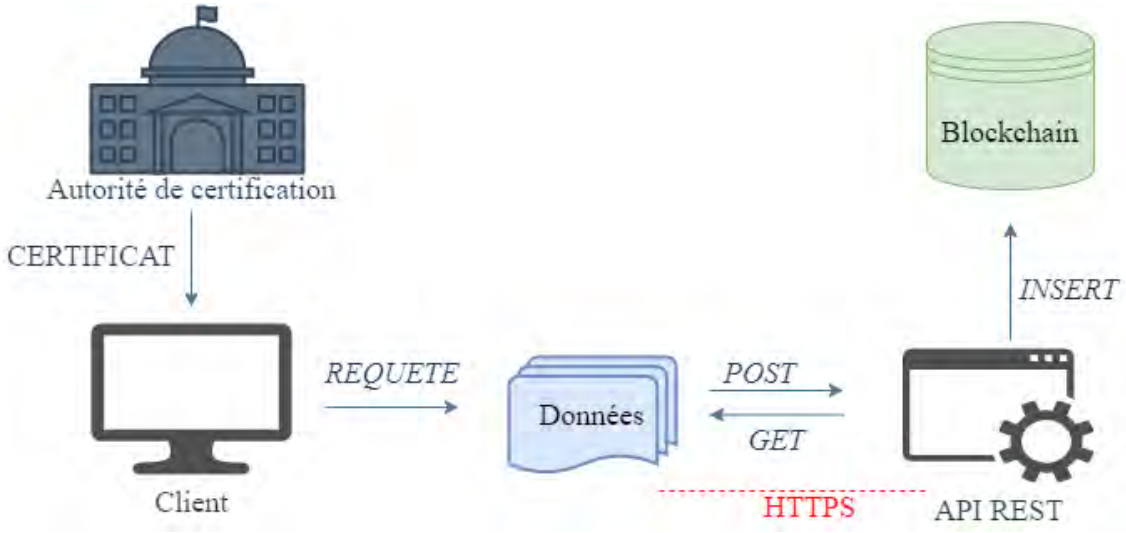

Figure 22 : Architecture logicielle de la solution

Du point de vue de la blockchain, l'architecture se présente comme le montre la figure suivante. Les universités sont organisés en pair qui sont connecté les uns aux autres et disposent chacun d'un registre dans lequel sont enregistrés toutes les transactions de publication de diplôme. Toutes les universités sont aussi connectées à l'Ordering service qui est chargé de recevoir, d'horodater et d'ordonner toutes les transactions avant de les distribuer dans le canal. Tous les pairs sont dans le même canal et le processus de consensus est établi entre tous les pairs. Les clients des universités disposent de certificat qui leur permet d'effectuer des transactions et donc de participer à la construction de la blockchain. Tandis que les autres clients ne peuvent pas effectuer de transactions. Ils ont uniquement la possibilité de se connecter au pair d'une université et de consulter les données du registre.

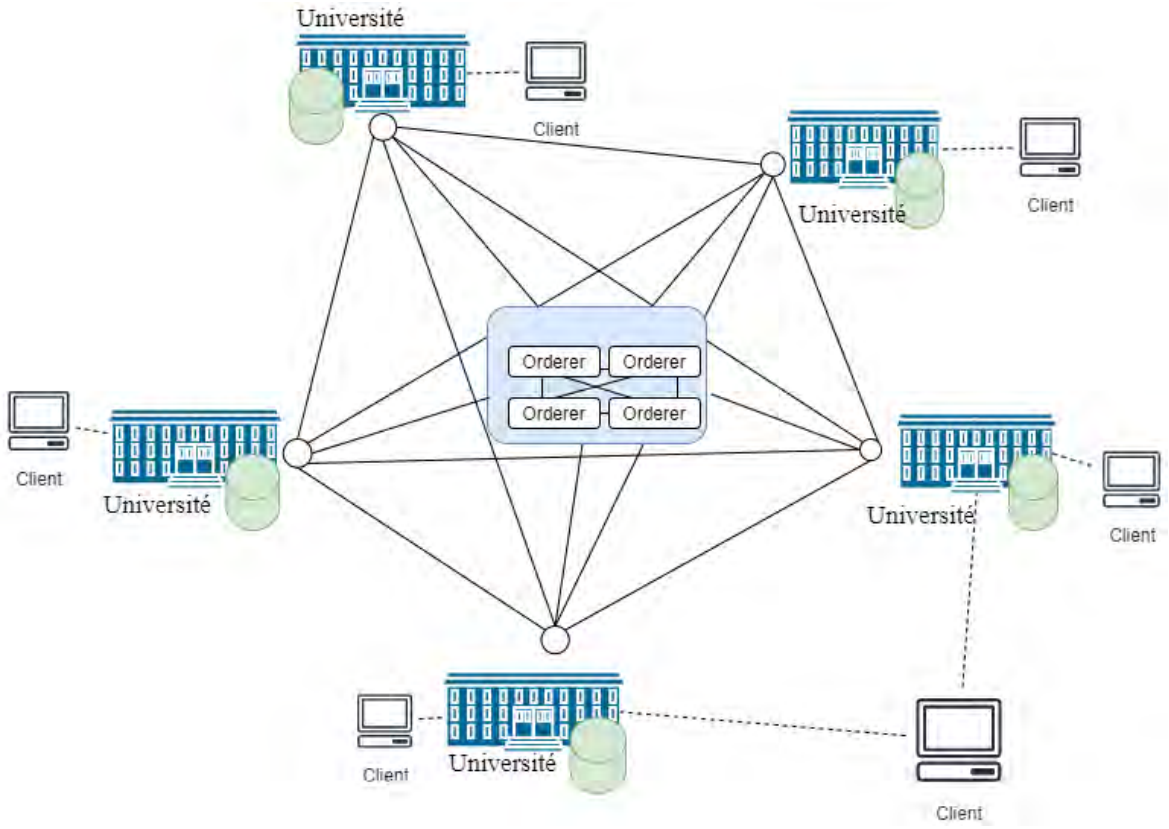

Figure 23 : Architecture de la solution proposée

Chapitre 5 : Outils utilisés

Nous allons dans ce chapitre présenter les détails des différents outils qui nous ont servi à mettre en place la solution et préciser quelle a été leur utilité.

5.1. Hyperledger Fabric

Hyperledger est un projet open source issu de la Fondation Linux et créé pour faire progresser les technologies de la blockchain intersectorielle. Il s'agit d'une collaboration mondiale en matière de logiciels libres impliquant des leaders de nombreuses industries. Plutôt que de déclarer une norme unique pour les blockchains, elle encourage une approche collaborative du développement des technologies de blockchain via un processus communautaire, avec des droits de propriété intellectuelle qui encouragent le développement ouvert et l'adoption de normes clés au fil du temps.

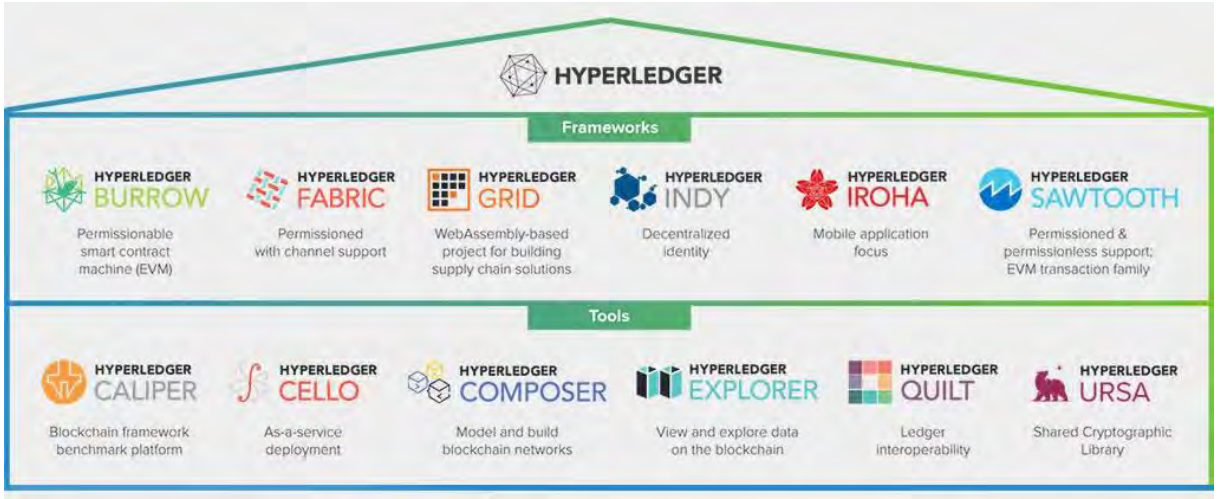

Figure 24 : Hyperledger Frameworks and Tools

Il y a cinq Framework de blockchain dans le projet Hyperledger, à savoir :

Hyperledger Iroha : Iroha, conçu pour les projets de développement mobile, est basé sur Hyperledger Fabric et a bénéficié de la contribution de Soramitsu, Hitachi, NTT Data et Colu. Il se caractérise par une conception C++ moderne, axée sur le domaine, ainsi que par un nouvel algorithme de consensus *Byzantine Fault Tolerant* basé sur une chaîne, appelé Sumeragi.

Hyperledger Sawtooth : Sawtooth a été mis au point par Intel et comprend un nouvel algorithme de consensus qu'Intel a mis au point, appelé Proof of ElapsedTime (PoET). L'objectif de PoET est de parvenir à un consensus distribué aussi efficacement que possible. L'Hyperledger Sawtooth a un potentiel dans de nombreux domaines, avec un support interdisant les déploiements autorisés et non autorisés et la reconnaissance des diverses exigences. Sawtooth est conçu pour être polyvalent.

Hyperledger Burrow : L'Hyperledger Burrow, qui a été fourni par Monax et Intel à l'origine, est une chaîne de blocs modulaire qui a été construite par le client selon les spécifications de la machine virtuelle Ethereum (EVM).

Hyperledger Indy : Contribué initialement par la Fondation Sovrin, Indy est un projet Hyperledger fait pour supporter l'identité indépendante sur les grands livres distribués. Hyperledger Indy fournit des outils, des bibliothèques et des composants réutilisables pour fournir des identités numériques ancrées sur des blockchains ou d'autres grands livres distribués.

Hyperledger Fabric (HLF) : Hyperledger Fabric, fourni par IBM, est conçu pour servir de base au développement d'applications ou de solutions à architecture modulaire. Il permet d'utiliser des composants "plug-and-play", tels que des services de consensus et d'adhésion, et utilise des conteneurs pour héberger les smart contrats, appelés "*chaincode*", qui constituent la logique applicative du système. Le reste de cette section consacré à l'Hyperledger Fabric et à sa conception, ses composants et son architecture.

5.1.1. Qu’est-ce que Hyperledger Fabric

Figure 25 : Hyperledger Fabric

Hyperledger Fabric est l'un des projets de la chaîne de production d'Hyperledger. Comme d'autres technologies de blockchain, il possède un registre, utilise des contrats intelligents et constitue un système permettant aux participants de gérer leurs transactions.

Là où Hyperledger Fabric se distingue d'autres systèmes de blockchain, c'est qu'il est privé et autorisé. Plutôt qu'un système ouvert sans autorisation qui permet à des identités inconnues de participer au réseau (ce qui nécessite des protocoles comme la "preuve de travail (proof of work)" pour valider les transactions et sécuriser le réseau), les membres d'un réseau Hyperledger Fabric s'inscrivent par l'intermédiaire d'un fournisseur de services aux membres (MSP) de confiance.

Hyperledger Fabric offre également plusieurs options modulables. Les données du registre peuvent être stockées dans plusieurs formats, les mécanismes de consensus peuvent être échangés et différents MSP sont pris en charge.

Hyperledger Fabric offre également la possibilité de créer des canaux, permettant à un groupe de participants de créer un registre séparé de transactions. C'est une option particulièrement importante pour les réseaux où certains participants peuvent être des concurrents et ne veulent pas que chaque transaction qu'ils effectuent - un prix spécial qu'ils offrent à certains participants et pas à d'autres, par exemple - soit connue de tous les participants. Si deux participants forment un réseau, alors ces participants - et aucun autre - disposent de copies du registre des transactions pour ce réseau.

Dans la suite les termes Hyperledger et Fabric seront utilisés interchangeablement pour désigner Hyperledger Fabric

5.1.2. Fonctionnalités

Hyperledger Fabric fournit les fonctionnalités suivantes :

- ***Gestion des identités***.

  Hyperledger fournit un service d'identité des membres connu sous le nom de fournisseur de services aux membres (MSP), qui gère et authentifie tous les participants sur le

réseau. En outre, des listes de contrôle d'accès peuvent être utilisées pour fournir des autorisations supplémentaires sur le réseau. Pour exemple :

Dans le cas d'accréditations académiques sur Hyperledger, seules les universités/institutions éducatives ont les autorisations de créer de nouvelles transactions sur le réseau ou de déclencher une chaincode sur le réseau. D'autre part, les étudiants n'ont que des droits de lecture sur le réseau.

- ***Vie privée et confidentialité.***

  Hyperledger Fabric permet aux entreprises et à tout groupe d'utilisateurs d'effectuer des transactions privées et confidentielles, de coexister sur le même réseau. Il permet la création de canaux privés qui fournissent des transactions privées et confidentielles pour des sous-ensembles spécifiques de membres du réseau.

  Toutes les données sur le réseau Hyperledger peuvent être rendues inaccessibles à moins que les permissions appropriées soient fournies au participant du réseau.

- ***Traitement efficient.***

  Hyperledger permet le traitement parallèle et concurrent des transactions en séparant le service qui ordonne les transactions du service qui les commit. Ceci augmente l'efficacité du traitement sur chaque peer et accélère la livraison des transactions. La séparation des transactions permet aussi de décharger le service qui ordonne les blocs des demandes d'exécution de transactions et de tenue de registre, tandis que les peers sont libérés de la mise à jour de l'état du canal.

- ***Chaincode.***

  Les chaincodes sont des applications qui encodent la logique invoquée par le déclenchement de transactions, effectuées sur le canal. Le Chaincode utilisé pour un changement de propriété d'un bien garantit que toutes les transactions de transfert de propriété sont soumises aux mêmes règles et exigences. Hyperledger dispose également d'une *chaincode système* qui définit les paramètres de fonctionnement de l'ensemble du canal.

- ***Conception modulaire.***

Hyperledger met en œuvre une architecture modulaire sous la forme de composants enfichables. Vous pouvez spécifier des algorithmes pour l'identité, l'ordonnancement (consensus) et le cryptage, dans n'importe quel réseau Hyperledger Fabric. Le réseau qui en résulte est une architecture blockchain universelle qui peut être adoptée, à travers différents marchés, réglementations et frontières géographiques différents.

5.1.3. Composant

Hyperledger Fabric vous permet de créer un réseau distribué composé de nombreux nœuds qui communiquent entre eux. Voici les composants qui constituent l'architecture complète de la Fabric :

- ***Actifs***

Les actifs sont représentés dans Hyperledger Fabric comme une collection de paires clé-valeur, avec l'historique de leur changement d'état enregistrés en tant que transactions dans le registre du canal où les actifs sont définis. Les actifs peuvent être des entités tangibles (disques durs, disques compacts, etc.) ou intangibles (contrats, etc.). Les actifs sont représentés au format binaire ou JSON.

- ***Chaincode***

On peut définir le chaincode comme similaire à la logique métier dans les applications traditionnelles.

- Chaincode applique les règles de lecture ou de modification des paires clé-valeur ou d'autres informations de la base de données d'état.

- Les fonctions chaincode s'exécutent par rapport à la base de données d'état actuelle du registre et sont initiées par une proposition de transaction.

- L'exécution du Chaincode aboutit à un ensemble d'écritures de valeurs de clés qui peuvent être soumises au réseau et appliquées au registre de tous les pairs.

- ***Registre***

Le registre fournit un historique vérifiable de tous les changements d'état réussis, c'est-à-dire les transactions valides, et des tentatives infructueuses de changement d'état, c'est-à-dire les transactions invalides, survenus pendant le fonctionnement du système.

Le registre est maintenu par tous les pairs du réseau et, éventuellement, chez un sous-ensemble d'orderer. Dans le contexte d'un orderer, on fait référence au registre en tant que 'OrdererLedger', tandis que dans le contexte d'un peer, on fait référence au registre en tant que 'PeerLedger'.

- ***La base de données d'état***

  Le dernier état de la blockchain (ou, simplement, l'état) est stocké comme un enregistrement clé-valeur versionné, où les clés sont des noms et les valeurs des blobs arbitraires. Ces entrées sont manipulées par les chaincodes fonctionnant sur la blockchain par le biais d'opérations put et get. L'état est stocké de manière persistante et les mises à jour de l'état sont enregistrées.

- ***Canal***

  Hyperledger Fabric permet à une organisation de participer simultanément à plusieurs réseaux de blockchain distincts via des canaux. Les canaux permettent un partage efficace de l'infrastructure tout en préservant la confidentialité des données et des communications. Ils sont suffisamment indépendants pour aider les organisations à séparer leur trafic de travail avec différentes contreparties, mais suffisamment intégrés pour leur permettre de coordonner des activités indépendantes si nécessaire.

- ***MSP (Membership Service Provider)***

  Malgré son nom, le fournisseur de services aux membres ne fournit rien en réalité. Le MSP est plutôt un ensemble de dossiers qui sont ajoutés à la configuration du réseau et qui sont utilisés pour définir une organisation à la fois vers l'intérieur (les organisations décident qui sont leurs administrateurs) et vers l'extérieur (en permettant aux autres organisations de vérifier que les entités ont l'autorité pour faire ce qu'elles tentent de faire). Alors que les autorités de certification génèrent les certificats qui représentent les identités, le MSP contient une liste d'identités autorisées.

- ***Consensus***

Le consensus est défini comme "la vérification complète de la justesse d'un ensemble de transactions constituant un bloc".

Le consensus est établi lorsque l'ordre et le résultat des transactions répondent aux critères de la politique établie sur le canal. Les avantages suivants sont fournis par le consensus :

- Vérification du versionnage de l'état actuel du registre.
- Protection contre les opérations de double dépense.
- Validation par les couches hiérarchiques.
- Vérification des transactions depuis l'initiation de la transaction jusqu'à ce qu'elle soit commit.

- ***Transactions***

Les transactions sont créées lorsqu'un chaincode est invoqué à partir d'une application cliente pour lire ou écrire des données dans le registre. Les clients soumettent des propositions de transaction à des peers endosseur pour exécution et validation, recueillent les réponses signées (validées) de ces pairs, puis regroupent les résultats et les validations dans une transaction qui est soumise à l'orderer. L'orderer ordonne et place les transactions dans un bloc qui est diffusé aux peers qui valident et écrivent les transactions dans le registre et mettent à jour l'état courant.

Les transactions peuvent être de deux types :

1. Les transactions de déploiement créent un nouveau chaincode et prennent un programme comme paramètre. Lorsqu'une transaction de déploiement s'exécute avec succès, le chaincode est installé "sur" la blockchain.
2. Les transactions d'invocation effectuent une opération dans le contexte d'un chaincode précédemment déployé. Une transaction *invoque* fait référence à un chaincode et à l'une de ses fonctions. En cas de succès, le chaincode exécute la fonction spécifiée, ce qui peut impliquer la modification de l'état correspondant et le renvoi d'une sortie.

- ***Nœuds***

Fabric est essentiellement constitué de nœuds qui sont les entités communicantes de la blockchain. Un "nœud" n'est qu'une fonction logique dans le sens où plusieurs nœuds de

types différents peuvent fonctionner sur le même serveur physique. Ce qui compte, c'est la manière dont les nœuds sont regroupés en "domaines de confiance" et associés aux entités logiques qui les contrôlent.

Il existe 3 types de nœuds à savoir :

3. Le client ;
4. Le peer (pair) ;
5. L'orderer ;

Le client : il représente l'entité qui agit au nom d'un utilisateur final. Il doit se connecter à un peer pour communiquer avec la blockchain. Le client peut se connecter à n'importe quel peer de son choix. Il a ainsi la possibilité de créer et invoquer des transactions.

Le peer : il reçoit des mises à jour de l'état des données de manière ordonnées sous forme de blocs provenant de l'orderer et maintient la base de données d'état (state database) et le registre (ledger). Un peer peut se voir attribuer un rôle spécial d'endosseur (endorser). La fonction d'un endorser est directement lié à une chaincode particulière et consiste à endosser une transaction avant que celle-ci ne soit validée. Chaque chaincode a une politique d'endossement bien définie qui peut faire référence à un ensemble de peer. Un peer peut aussi jouer le rôle d'ancre (anchor peer) ou de leader (leader peer).

L'orderer ou Ordering service : il fournit un canal de communication partagé aux clients et aux peers, offrant un service de diffusion pour les messages contenant des transactions. Il ordonne les transactions dans un bloc et distribue ensuite les blocs aux pairs connectés pour vérification et validation.

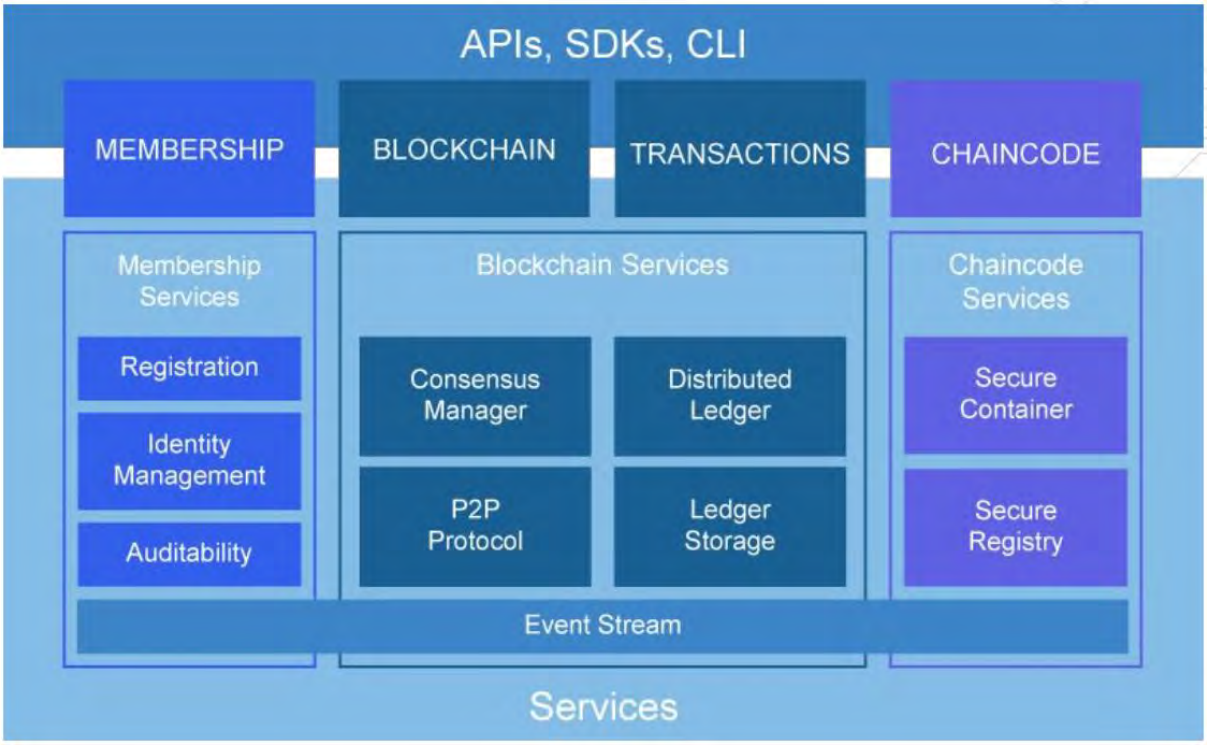

Figure 26 : Architecture d'Hyperledger Fabric

5.1.4. Gossip Protocol

Hyperledger Fabric optimise les performances, la sécurité et l'évolutivité du réseau blockchain en répartissant la charge de travail entre les pairs chargés de l'exécution des transactions (endossement et validation) et les nœuds de commande des transactions. Ce découplage des opérations réseau nécessite un protocole de diffusion des données sécurisé, fiable et évolutif pour garantir l'intégrité et la cohérence des données. Pour répondre à ces exigences, Fabric met en œuvre un protocole de diffusion des données appelé 'Gossip data dissemination protocol'.

Les pairs utilisent ce protocole pour diffuser les données du registre et du canal de manière évolutive. Le protocol Gossip est continu, et chaque pair sur un canal reçoit constamment des données de registre actuelles et cohérentes de plusieurs pairs. Chaque message diffusé est signé, ce qui permet d'identifier facilement les participants qui envoient des messages falsifiés et d'empêcher la distribution du ou des messages à des cibles non désirées. Les pairs affectés par des retards, des partitions de réseau ou d'autres causes entraînant des blocs manquants seront finalement synchronisés avec l'état actuel du registre en contactant les pairs en possession de ces blocs manquants.

Le protocole de diffusion de données Gossip remplit trois fonctions principales sur un réseau Fabric :

6. Gérer la découverte des pairs et l'adhésion au canal, en identifiant continuellement les pairs membres disponibles et en détectant éventuellement les pairs qui se sont déconnectés.
7. Diffusion des données du registre sur tous les pairs d'un canal. Tout pair dont les données ne sont pas synchronisées avec le reste du canal identifie les blocs manquants et se synchronise en copiant les données correctes.
8. Mettre à niveau les pairs nouvellement connectés en permettant le transfert d'état de pair à pair pour la mise à jour des données du registre.

Les pairs peuvent également exercer un pull plutôt que d'attendre la livraison d'un message. Ce cycle se répète, avec pour résultat que l'adhésion au canal, le registre et les informations d'état sont continuellement maintenus à jour et synchronisés. Pour la diffusion de nouveaux blocs, le pair leader du canal prend les données de l'Ordering service et lance la diffusion aux pairs de sa propre organisation.

5.1.5. Flux de transactions

En supposant que le canal est correctement configuré, que l'utilisateur de l'application est enregistré auprès d'un CA, que le chaincode est déployé sur le canal et que la politique d'endossement est définie ; décrivons les mécanismes qui se déroulent lors d'une transaction.

1- Le client qui souhaite effectuer une opération initie une transaction. La transaction est envoyée au peer endorser pour validation.

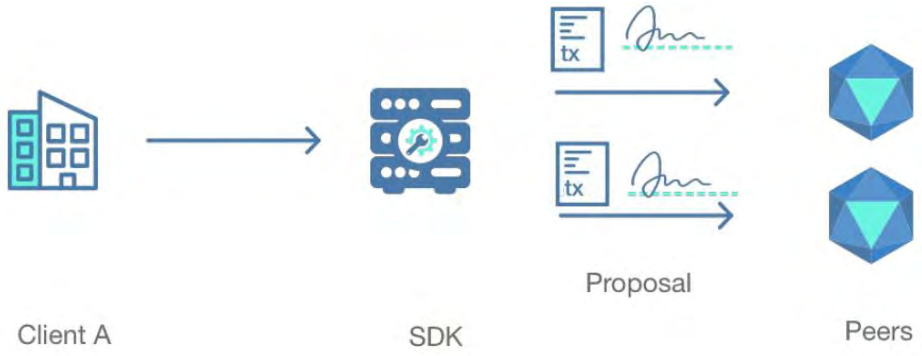

Figure 27 : Initiation d'une transaction

2- Les peers endorser vérifient que la transaction est bien formée, que la signature est valide et que le client est autorisé à effectuer cette transaction. Ensuite ils invoquent le chaincode et simule la transaction puis signe le résultat. On dit que le peer endosse la transaction. Aucune mise à jour du registre n'est effectuée à cette étape. A la fin du processus, le résultat est renvoyé en réponse au client.

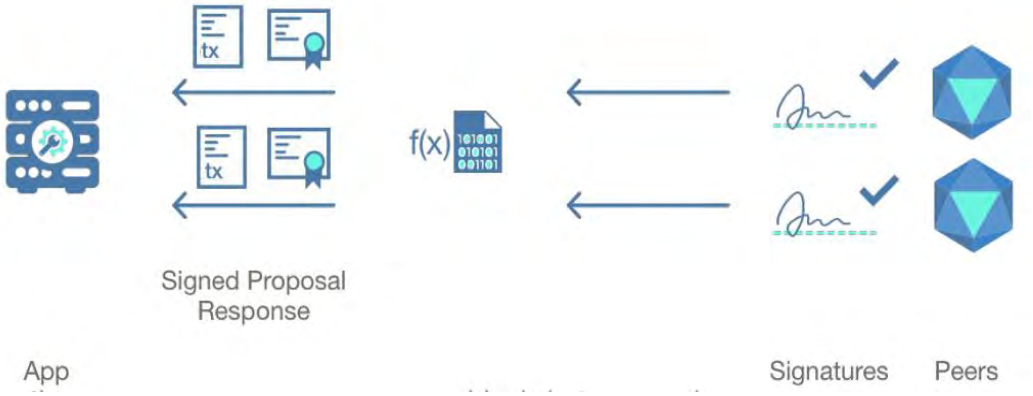

Figure 28 : Réponse à l'initiation de la transaction

3- Le client inspecte les réponses qui lui sont envoyées afin de déterminer si la politique d'endossement a été respectée, si par exemple le nombre minimum de peer endorser ont approuvés la transaction.

4- Le client assemble ensuite la proposition de transaction et les signatures des peers endorser qu'il envoie cette fois ci à l'orderer (ou Ordering service). L'Orderer n'a pas besoin d'inspecter le contenu de la transaction. Il reçoit simplement les transactions, les ordonne chronologiquement et crée des blocs.

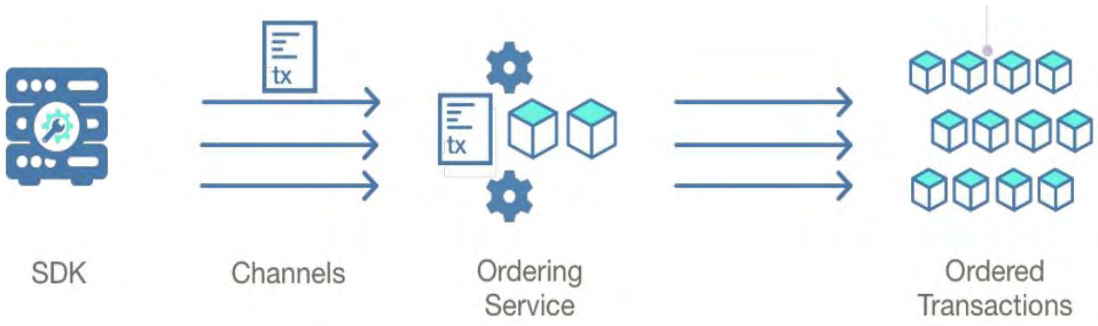

Figure 29 : Envoie de la transaction à l'OS

5- L'orderer diffuse ensuite le nouveau bloc formé à tous les peers du canal. Ces derniers se chargent de s'assurer que la politique d'endossement a été respectée pour chacune des transactions du blocs. Les transactions sont ensuite marquées comme valide ou invalide suivant le résultat.

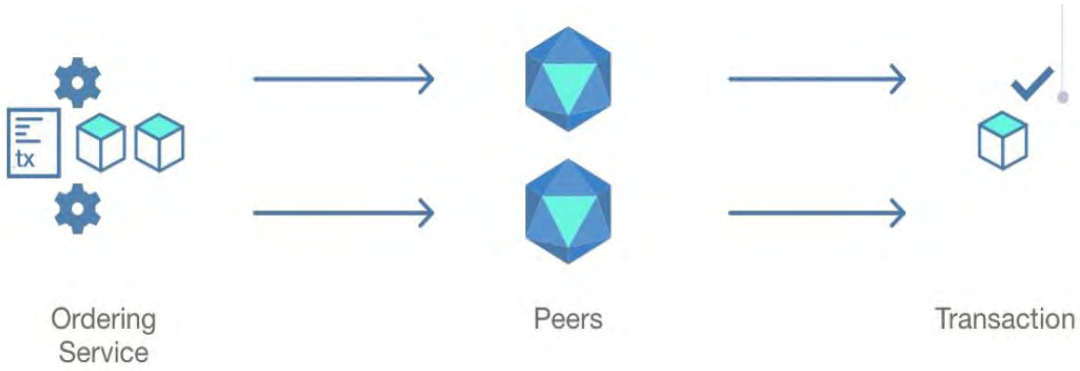

Figure 30 : Diffusion du bloc aux peers

6- Chaque peer ajoute le bloc à son registre et pour chaque transaction valide du bloc, la base de données d'état est mise à jour. Une notification est émise pour signaler au client que la transaction a été immuablement ajoutée et si cette dernière a été validée ou pas.

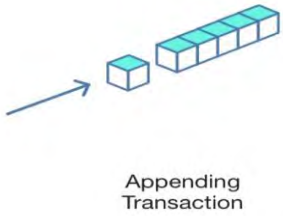

Figure 31 : Mise à jour du registre

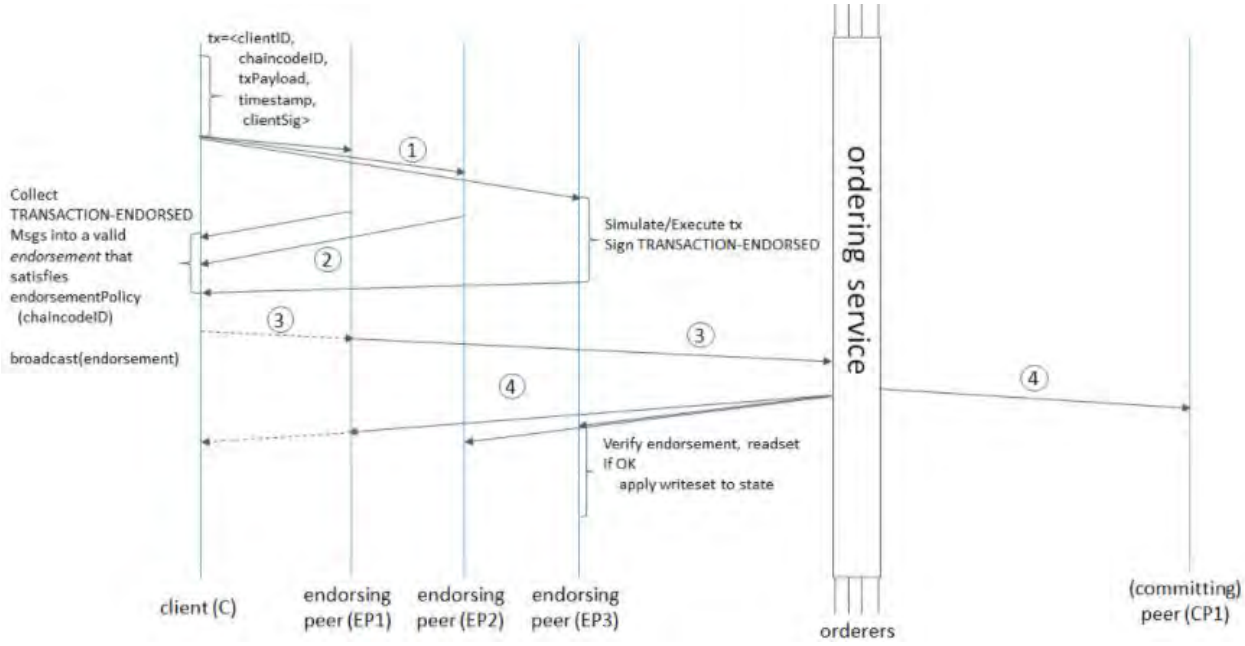

Figure 32 : Exemple récapitulatif du flux d'une transaction

5.1. Virtual Box

VirtualBox est un hyperviseur permettant de virtualiser des systèmes d'exploitation (OS) invités sur une machine hôte. Oracle VM VirtualBox, de son nouveau nom, peut créer des machines virtuelles (VM), en exécuter une ou plusieurs en même temps, en mettre en pause,