République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Seddik Benyahia
Jijel
Faculté des Sciences Exactes et Informatique
Département d'Informatique

Mémoire de fin d'études pour l'obtention du diplôme
Master en Informatique
Option : Informatique Légale et Multimédia

Thème

Gestion des identités numériques sur Blockchain

Réalisé par :
Mlle.FERHAT Yamna

Encadré par :
Dr.MAHAMDIOUA Meriama

Promotion : 2022

Dédicace

Ce travail est dédié à la mémoire de mon chère pére, Allah Yerhmou, qui serait le plus heureux s'il pouvait attendre de voir ce que je suis devenue.

C'est avec un énorme plaisir, un cœur ouvert et une immense joie, que je dédie ce modeste travail à ma très chère mère, à qui je dois beaucoup pour ces sacrifices, son amour, son aide et son soutien afin de me voir parvenir à ce que je suis devenue aujourd'hui, Je T'aime du plus profond de mon cœur.

A mon frère, mon idole, qui a consenti beaucoup de sacrifices pour assurer mon éducation, qui a fait ma vie pleine de bonheur.

A mes yeux, la plus brillante de toutes, ma sœur et sa petite famille que Dieu les préserves et les procures la longue vie.

A toute ma famille.

A mon encadreur : Madame ***MAHAMDIOUA Meriama*** *qui m'a donnée le courage d'accomplir ce projet.*

À tous mes amies, et spécialement Redouane et mes deux copines Houria et Rania pour leur soutien.

À moi-même, pour ne pas avoir abandonné, je suis très fier de moi !

une sincérité si merveilleuse. . .jamais oubliable.

Remerciements

Je remercie d'abord ALLAH le tout-puissant qui m'a guidé et qui m'a donné la force et la volonté pour achever ce travail.

Je remercier infiniment ma chère maman et mon chèr frère qui ont toujours cru en moi. C'est grâce à leur soutient moral, matériel et prières que j'ai accomplie ce travail, ils savent déjà combien je leur dois.

Comme je remercie mon encadreur ***Mme. MAHAMADIOUA Meriama*** *de m'avoir orienté avec ces précieux conseils et remarques.*

J'en profite également pour remercier toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Mes remerciements vont aussi à tous mes professeurs, enseignants et toutes les personnes qui n'ont cessées de me donner des conseils très importants, qui m'ont fait comprendre et sentir ce que c'est l'informatique.

A toute l'équipe pédagogique qui a participé à ma formation depuis l'école primaire à ce jour, également à tous ceux qui m'ont aidées de près ou de loin lors de l'élaboration de ce travail..

A tous mes enseignants du Département d'informatique de Jijel qui m'ont initiés aux valeurs authentiques, en signe d'un profond respect.

Enfin, je tiens à remercier les jurés pour avoir accepter d'examiner et juger mon modeste travail.

Résumé

La blockchain est une technologie qui permet la transmission et le stockage des données d'une manière sécurisée et transparente sans organe central. Cette nouvelle technologie pourrait être appliquée dans plusieurs domaines tels que la gestion d'identité numérique. Dans ce travail, nous avons proposé un système de gestion d'identité auto-souveraine basé sur la blockchain. Le système permet à l'utilisateur de choisir ses propres données pour s'identifier et gérer la permission d'accès à ses données. Dans notre proposition, le système de fichiers interplanétaires IPFS est utilisé pour le stockage des données d'utilisateur. Un fournisseur d'identité valide ces données par un ajout des hashs à la blockcchain en utilisant un smart contrat. Le fournisseur de service à son tour vérifie l'existence des hashs dans la blockchain en répondant à la demande d'utilisateur pour que ce dernier puisse accéder au service fourni.

Mot clés : *Blockchain, Gestion d'identité, Contrat intelligent, Blockchain,auto-souveraine, fournisseu de service, fournisseur d'identité.*

Abstract

Blockchain is a technology that allows the transmission and storage of data in a secure and transparent way without a central body. This new technology could be applied in several areas such as digital identity management. In this work, we proposed a self-sovereign identity management system based on blockchain. The system allows the user to choose their own data to identify themselves and manage the permission to access their data. In our proposal, the IPFS interplanetary file system is used for storing user data. An identity provider validates this data by adding hashes to the blockchain using a smart contract. The service provider in turn verifies the existence of the hashes in the blockchain by responding to the user request so that the user can access the provided service.

Keywords : *Blockchain, Identity Management, Smart contracts, Self-Sovereign, Service Provider, Identity provider.*

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GÉNÉRALE

Contexte de travail

La blockchain (ou chaîne de blocs) est une technologie qui permet de stocker et de transmettre des informations d'une façon transparente et sécurisée. Elle ressemble à une grande base de données distribuée qui garde la trace de l'historique de tous les échanges entre ses utilisateurs depuis sa création, sans la nécessité d'un organe centrale. [4]

Cette technologie est apparue au début dans le domaine financier, pour effectuer des transactions en ligne sans se faire passer par des autorités centrales, tout en incluant des techniques de cryptographie, de hachage, de décentralisation et de consensus. Dans ces dernières années, la technologie blockchain est considérée comme un développement transformateur dans les systèmes distribués. Son principe de fonctionnement décentralisé a trouvé des applications dans des domaines variés, comme celui de la gestion des identités numériques.

L'identité numérique est une représentation numérique d'une entité réelle. C'est un moyen pour prouver électroniquement que l'on est bien la personne que l'on prétend être, d'une manière permettant la distinction des autres utilisateurs. [23]

Motivation

Dans la vie moderne, l'accès aux services électroniques via des fournisseurs de services en ligne augmente énormément. En effet, l'économie numérique en utilisant l'Internet offre de réels avantages aux personnes et organismes. Cependant, l'Internet a été construit sans moyen standard et explicite d'identifier les usagers. Cette identification simplement commencé par la création des comptes locaux avec des noms d'utilisateur et des mots de passe, et c'est la solution dominante depuis lors.

La gestion des identités a ensuite connu un intérêt accru à cause du besoin croissant d'identités numériques, et plusieurs modèles ont été proposés tels que centralisé, fédéré, etc. Dans ces modèles, l'identité est soit gérée par une autorité, où l'utilisateur doit avoir un identifiant d'identité numérique accordé par l'organisation dont il veut accéder à ses

services (et donc avoir un nouvel identifiant pour chaque organisation), ou elle est stockée dans plusieurs systèmes de gestion différents, et contrôlée par diverse autorité fédérées.

Malgré l'évolution de ces modèles de gestion d'identité numériques, ils souffrent des problèmes suivants :

- L'utilisateur n'a pas de contrôle sur la gestion de son identité numérique, et il peut être retiré à tout moment.
- La notion de la centralisation de ces modèles de gestion d'identité, qui est l'inconvénient majeur de la défaillance des données, existe toujours.

Pour répondre à ces exigences, un nouveau modèle est proposé dans la littérature, soit le modèle d'identité auto-souveraine, (en anglais : Self-Sovereign identity), qui décrit comment un individu peut avoir le droit de contrôler et de gérer son identité numérique sans l'intervention d'une autorité, tout en assurant un ensemble de critères tels que : la protection, existante, interopérabilité etc.

Ces dernières années, des tentatives ont été faites pour introduire des solutions de gestion d'identité basées sur la blockchain, qui permettent à l'utilisateur de prendre le contrôle de sa propre identité et de réaliser une identité auto-souveraine. Cependant, ces modèles sont encore à un stade précoce, et des recherches supplémentaires doivent être menées pour déterminer si les systèmes d'identité pourraient bénéficier ou non de l'utilisation de la blockchain.

Contribution

Dans ce travail, nous avons proposé un système d'identité auto-souveraine, basé sur l'utilisation de la blockchain, dont les tiers suivants ont participé à son architecture : utilisateur, fournisseur d'identité, fournisseur de service, système de fichiers interplanétaire IPFS (InterPlanetary File System) et blockchain. Le stockage des données est effectué sur le système de fichier distribué IPFS et la validité de ces données est assurée par un sauvegarde des hashs des données sur la blockchain.

Le modèle permet à l'utilisateur de choisir ses propres donnés pour s'identifier et de gérer la permission d'accès à ses données. L'utilisateur stocke ses données sur le système de fichier distribué IPFS puis il envoi leurs hashs à un fournisseur d'identité, dont son rôle ici est de vérifier et de prouver la validation des données fournies par l'utilisateur. Ce fournisseur d'identité prend alors les données de IPFS en les cherchant par les hashs envoyés par l'utilisateur, et après la vérification, soit il les valide, soit il les rejette et ça selon les preuves existantes. Si les données sont prouvées, le fournisseur d'identité crée un code composé de hashs des données plus des données supplémentaires, puis il l'envoi à l'utilisateur, et en même temps il applique une fonction de hachage sur les hashs des données et les enregistre sur la blockchain.

A son tour, et quand un utilisateur veut accéder à un service via un fournisseur de service, il présente juste le hash de la donnée nécessaire pour l'accès à ce service plus le code envoyé par le fournisseur d'identité. Le fournisseur de service va vérifier l'existence de hash

de ces données (le hash de hash de la donnée et code fournis par le fournisseur d'identité) sur la blockchain, et par suit, il permet l'accès à ses services.

Notre solution proposée est testée dans le cadre de l'accès aux documents universitaires afin de pouvoir accéder à un service comme postuler à un emploi ou se réinscrire dans une autre université. Cette proposition permet à l'utilisateur de garder le contrôle et la permission d'accéder et de sélectionner la donnée cherchée et pas tout l'ensemble des données, ainsi elle assure l'existence des données sur un système distribué IPFS, et assure aussi la validation des données en utilisant la blockchain.

Organisation de mémoire

Ce mémoire est structuré en trois chapitres. Le premier chapitre expose la technologie blockchain. Dans le deuxième chapitre, nous présentons la gestion d'identité numérique. Le dernier chapitre révèle notre proposition ainsi que la solution développée. Notre mémoire se termine par une conclusion générale et des perspectives futures.

CHAPITRE 1

TECHNOLOGIE BLOCKCHAIN

1.1 Introduction

Chaine de blocs ou Blockchain, une nouvelle technologie inspirante qui attire l'intention de plus en plus. Le principe de la blockchain existe depuis 1990, mais elle a connue une très grande popularité lors de l'apparition de la cryptomonnaie Bitcoin en 2009 qui a été conçue par une personne ou un groupe de personnes connue sous le pseudonyme de Satoshi Nakamoto. La blockchain est une technologie de transmission et de stockage des données sécurisées, elle est considérée comme la plus grande révolution après Internet. De nombreuses entreprises veulent l'intégrer à leurs systèmes pour bénéficier de ses avantages, mais malheureusement peu de gens seulement qui comprennent c'est quoi la blockchain et comment elle fonctionne. Dans ce chapitre nous allons bien expliquer cette technologie (son fonctionnement, ses composants, ses types, ainsi que son utilité).

1.2 Définition

Jusqu'à présent il n y a toujours pas de définition exacte pour la blockchain. Nous présentons en ce qui suit quelques définitions de la blockchain :

- **Déf 1.** "La blockchain est une base de données distribuée (grand livre) constituée de blocs de données interconnectés et protégés par des concepts cryptographiques contre la falsification". [2]

- **Déf 2.**"La blockchain est un registre numérique incorruptible des transactions économiques qui peut être programmé pour enregistrer non seulement les transactions financières mais aussi pratiquement tout ce qui a de la valeur." [4]

- **Déf 3.**"La Blockchain (ou chaîne de blocs) est une technologie de stockage et de transmission d'informations et qui fonctionne sans administrateur central qui

contrôle l'ensemble." [9]

En d'autre terme, on peut dire que la blockchain est une grande base de données, partagée entre plusieurs participants, que chacun d'entre eux a une copie d'elle. Son architecture qualifiée de distribuée et fonctionne en peer-to-peer. La blockchain n'est pas dirigée par un organe centralisé, mais elle peut être évoluée dans le temps par ses utilisateurs qui peuvent à tout moment ajouter ou vérifier des informations.

1.3 Fonctionnement de blockchain

La blockchain a été utilisée au début avec les cryptomonaies. Dans cette section, nous expliquons le fonctionnement de la blockchain en prenant la cryptomonnaie Bitcoin comme exemple.

1.3.1 Concepts de base pour le fonctionnement d'une blockchain

Avant d'évoquer le fonctionnement de la blockchain, nous présentons des notions générales et nécéssaire pour le fonctionnement de cette technologie.

- **Portefeuille (Wallet)**
  Un portefeuille est un logiciel, qui sauvegarde les clés privées et publiques d'un utilisateur de la blockchain. Ce logiciel permet d'initialiser ou de lancer des transactions et de vérifier le solde. Un même portefeuille peut avoir plusieurs clés privées (et clés publiques) qui sont générées de la clé privée originale.
- **Transaction**
  Une transaction, est définie comme une opération qui permet de changer l'état de la blockchain ou d'effectuer une lecture d'après de la blockchain. (pour plus de détails, voir la section 1.4)
- **Bloc**
  Un bloc est un rassemblement de transactions effectuées. Chaque bloc est lié avec un autre bloc qui le précéde, ce qui forme une chaine de blocs d'où l'appellation blockchain (pour plus de détails, voir la section 1.4)
- **Minage**
  Le minage est un concept de base pour une blockchain.

  Minage
  Le minage est le processus qui sert à valider les blocs par des mineurs, qui sont en compétition pour trouver des solutions pour des problèmes cryptographiques complexes. [13]

  Mineur
  Les mineurs sont des noeuds (pairs) dans un réseau blockchain, qui se chargent de créer et d'ajouter de nouveaux blocs (contenant des transactions des utilisateurs (voir la figure 1.1)) dans la blockchain en essayant de trouver une solution pour les problèmes cryptographiques demandés. Afin d'être récompensés pour

le temps et la puissance de calcul exploités, les mineurs reçoivent des frais de transactions et la récompense de minage associée à la création d'un nouveau bloc. [13]

FIGURE 1.1 – Ferme de minage
[42]

- **Réseau pair à pair**
  Un réseau pair à pair, appelé aussi réseau d'égal à égal, est une architecture de réseau, qui permet à deux ordinateurs de s'échanger des données sans la nécéssité de serveurs pour le stockage de données, ce qui permet d'avoir une certaine sécurité en cas de défaillance des données. Le réseau blockchain est basé sur un réseau pair à pair distribué (voir la figure 1.2). Les noeuds du réseau sont des ordinateurs éparpillés partout dans le monde et qui sont liés au réseau blockchain, ou chacun d'entre eux possède une copie complète ou partielle de la blockchain. Il en existe deux types de noeuds : noeuds complets et noeuds légers [17]

  Noeuds complets : Ce sont des noeuds qui contienent une copie complète des transactions, c'est à dire l'historique des transactions, il sont chargés de la vérification des transactions. ils peuvent être considérés comme des serveurs. [17]

  Noeuds légers : Ceux-ci contiennent une partie de la blockchain (seulement les en-tête des blocs). Ce genre de noeuds doit se trouver avec des noeuds complets. [17]

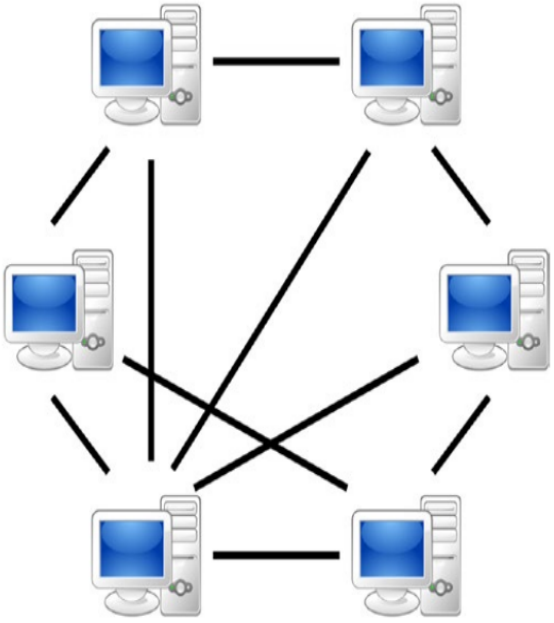

FIGURE 1.2 – Réseau pair à pair

1.3.2 Fonctionnement de la blockchain

Pour illustrer le fonctionnement de la blockchain, nous prenons l'exemple de Bitcoin :

- Lorsque un utilisateur A veut envoyer des bitcoins à un utilisateur B. La première étape consite à la demande d'une transaction (création de la transaction) de la part de l'utilisateur A, cette transaction sera signée avec sa clé privée qui est sauvegardée dans son wallet. Cette signature permet l'authentification. Ensuite cette transaction est diffusée pour tous les noeuds du réseau blockchain (voir la figure 1.3).

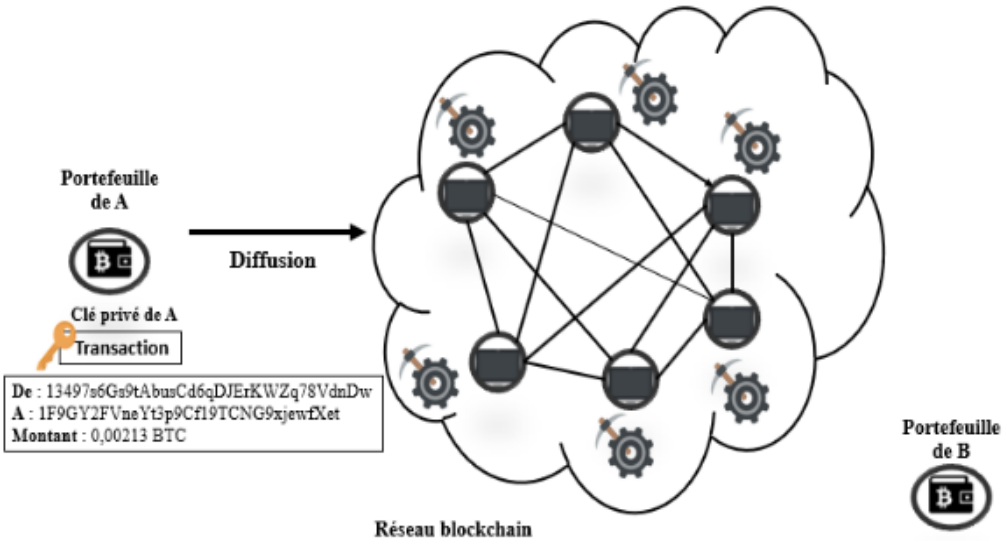

FIGURE 1.3 – Demande de transaction et distribution sur le réseau. [15]

- Une fois la transaction est diffusée, les noeuds du réseaux vont vérifier l'autheniticité de cette denrière ainsi que les moyens essentiels pour effectuer cette transaction, en utilisatant la clé publique de l'utilisateur A (voir la figure 1.4).

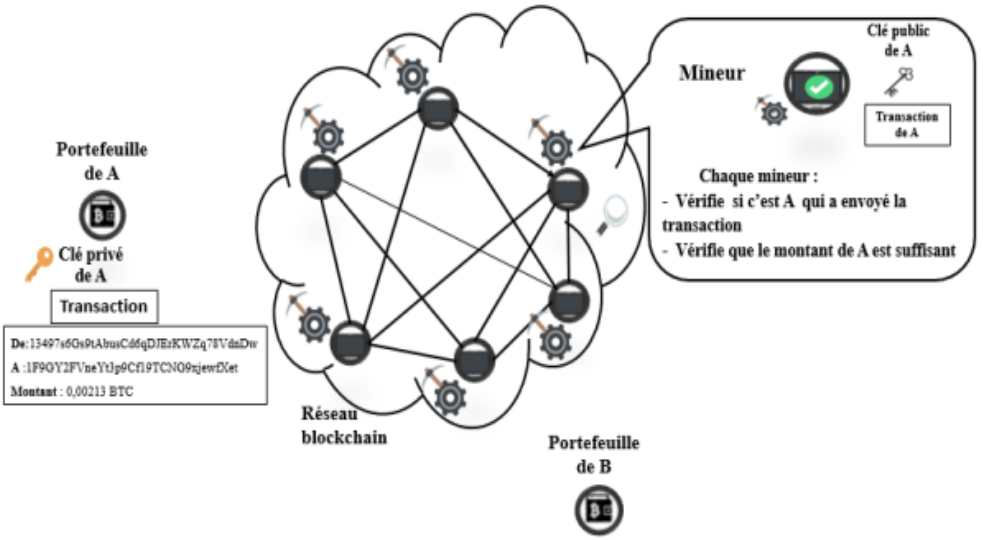

FIGURE 1.4 – Vérification de la transaction
[15]

- Lorsque la transaction est validée, un bloc qui contiendra cette dernière (avec d'autres transactions des autres utilisateurs) sera créé et la transaction sera ajoutée a ce bloc (voir la figure 1.5).

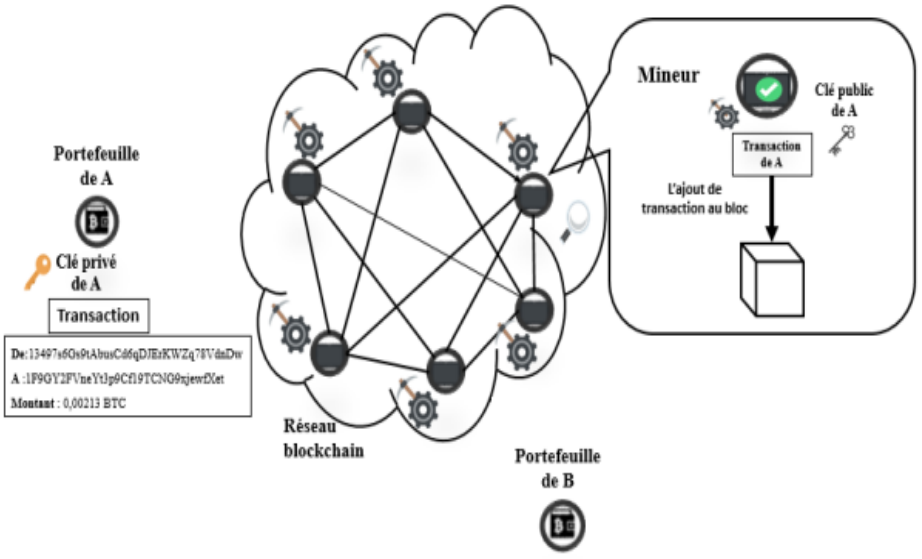

FIGURE 1.5 – Création de bloc pour la transaction
[15]

- Ensuite, Les mineurs vont se mettre en compétition pour la validation du bloc en appliquant un algorithme de consensus (voir la figure 1.6).

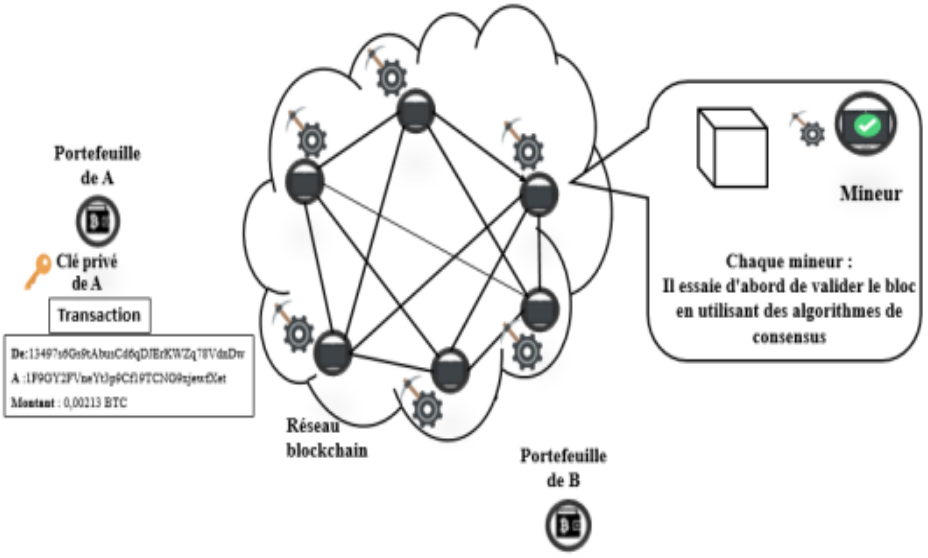

FIGURE 1.6 – Validation de la transaction
[15]

- Le noeud qui trouve la solution du problème cryptographique en premier, il va distirbuer le bloc sur le réseau blockchain pour qu'il soit vérifé, ce noeud va recevoire une récomponse lorsque le deuxième utilisateur reçoit la transaction et cela suite au travail qu'il a achevé (voir la figure 1.7).

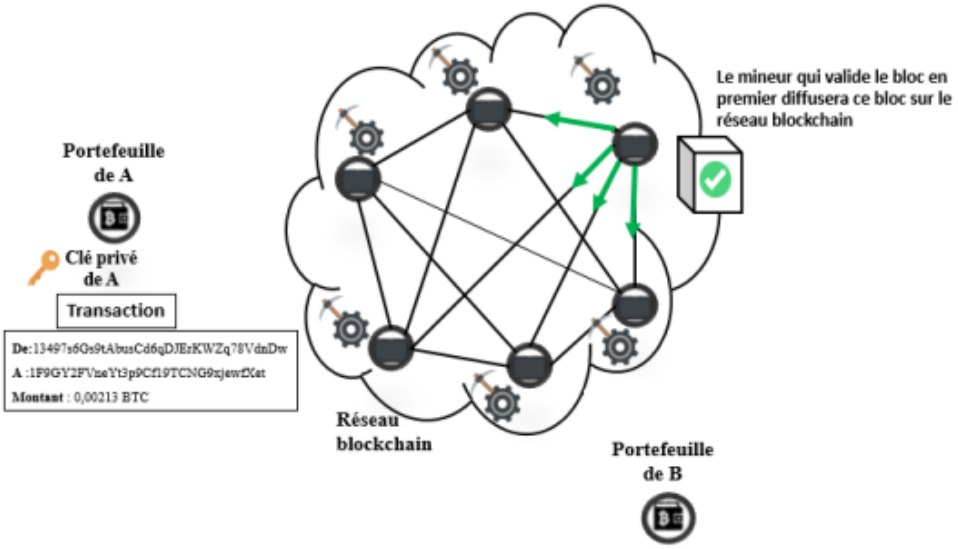

FIGURE 1.7 – Distribution de bloc
[15]

- Après la vérification du bloc, il sera daté et ajouté à la blockchain (voir la figure 1.8).

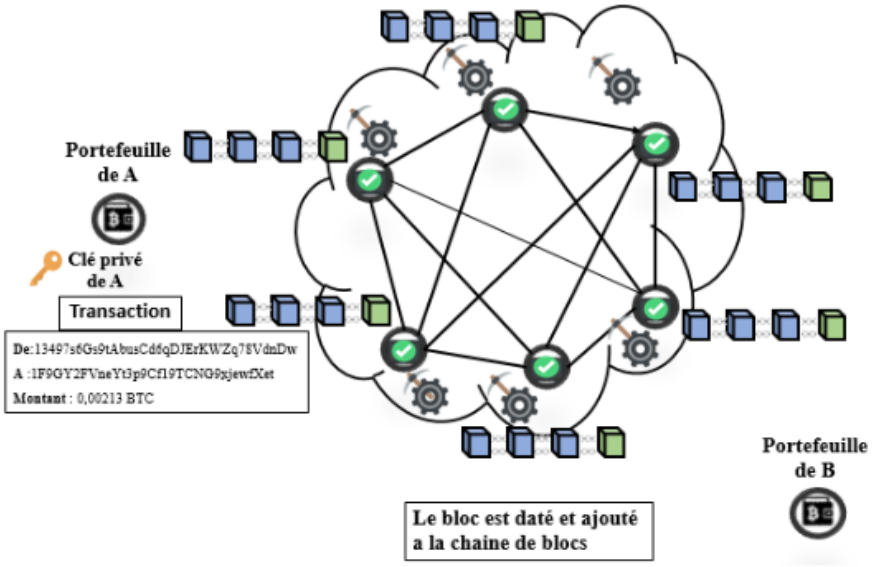

FIGURE 1.8 – Vérification du bloc
[15]

- Au final, la transaction est reçue par l'utilisateur B (voir la figure 1.9).

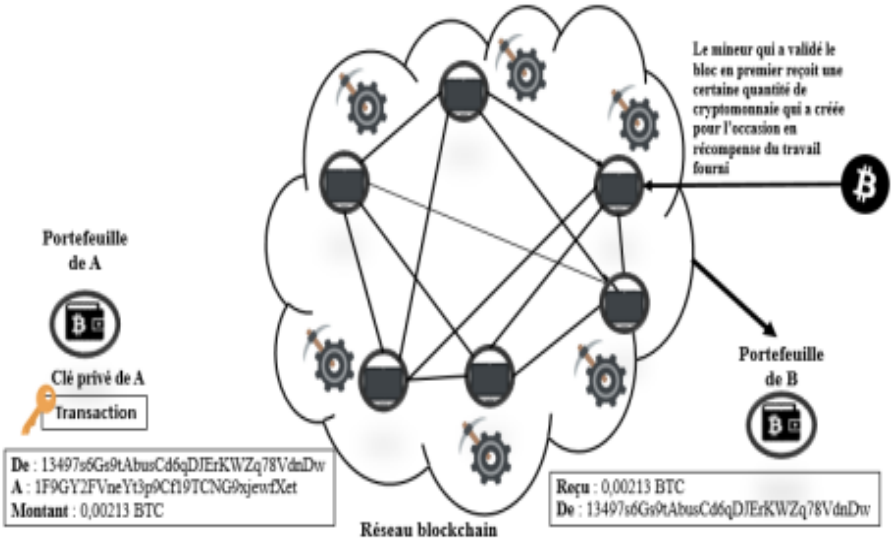

FIGURE 1.9 – Réception de la transaction
[15]

1.4 Composants principaux de la blockchain

Nous avons parlé brièvement de certains composants utilisés dans une blockchain dans la section précédente. Dans cette section, nous allons les détailler.

1.4.1 Transaction

La transaction est le petit composant de la blockchain. Une transaction est un transfert saigné et envoyé d'une addresse vers une autre, appelé entré (input) et sortie (output) rescpectivement. [1]

1.4.1.1 Structure d'une transaction

Une transaction est composée des entrées et des sorties. Elle a un certain nombre de champs. La table 1.1 présente la structure d'une transaction bitcoin.

Champ	**Dimension**	**Description**
Version	4 octets	La version du règlement (quelle règles suit)
Compteur d'entré	1-9 octets	Combien d'entrée sont inclues
Entrées	Variable	Un ou plusieurs entrées de transactions
Compteur de sortie	1-9 octets	Combien de sorites sont inclus
Sorties	Variable	Un ou plusieurs sortie de transactions
Temps de verrouillage (Locktime)	4 octets	Définie quand la transation peut être ajouté à la blockchain

TABLE 1.1 – Structure d'une transaction de bitcoin.
[44]

- **Entrée**
  Les entrèes sont ce qui va être transféré, donc l'entrée de la transaction peut être considérée la source de ce qui va être transféré, ou bien une référence des évenements passées. L'expéditeur doit signer la transaction pour qu'il prouve qu'il a un accès aux entrèes. [1]

- **Sortie**
  Les sorties, sont les sources destinataires ainsi que la quantitée de ce qu'ils recevront. [1]

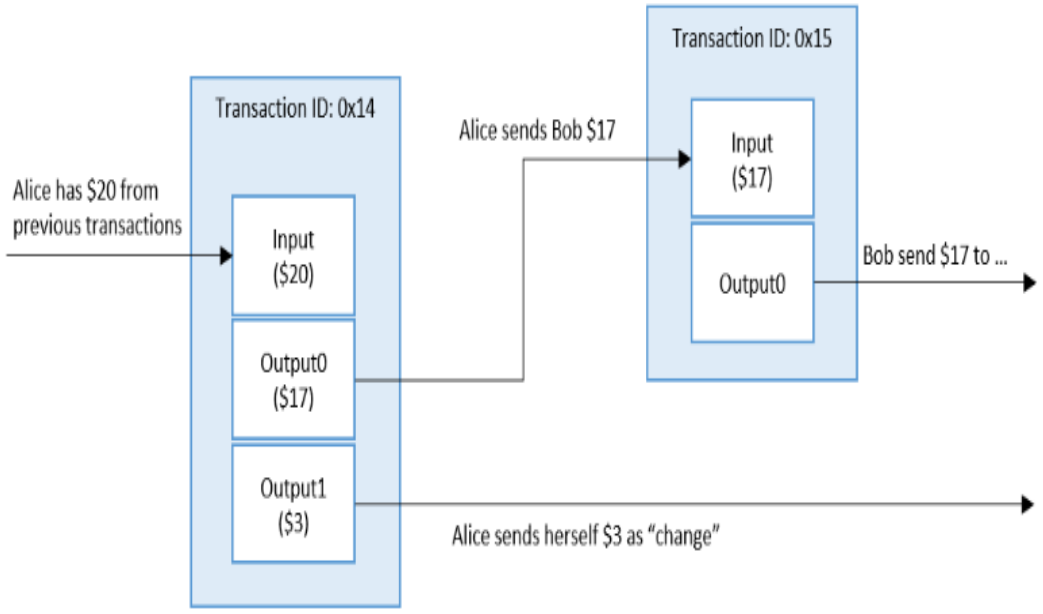

FIGURE 1.10 – Exemple d'une transaction d'une cryptomonnaie [1]

1.4.1.2 Formes de transaction

On peut distinguer trois formes des transactions : simple, avec plusieurs entrèes, avec plusieurs sorties.

- **Transaction simple**
  La transaction simple, est un traitement primitif d'une adresse vers une autre. Puisqu'il est difficile d'envoyer le montant exact spécifié des fonds au destinataire, le reste des fonds est renvoyé à l'expéditeur à l'aide d'un changement d'adresse, car l'utilisateur ne peut pas faire un paiement vers lui même, donc il change l'adresse du portefeuille afin de recevoir les fonds restants

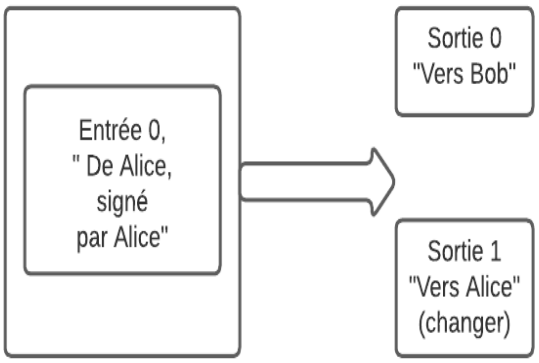

FIGURE 1.11 – Transaction simple

- **Transaction avec plusieurs entrées**
  Ce type de transaction est un agrégat de plusieurs entrées en une seule exportation unique. Ces transactions se produisent en générale lorsque le wallet de l'utilisateur n'a pas une seule entrée suffisamment grande pour remplir le montant du paiement. Donc le wallet choisit de retirer des cryptomonnaie de plusieurs adresses pour les utiliser

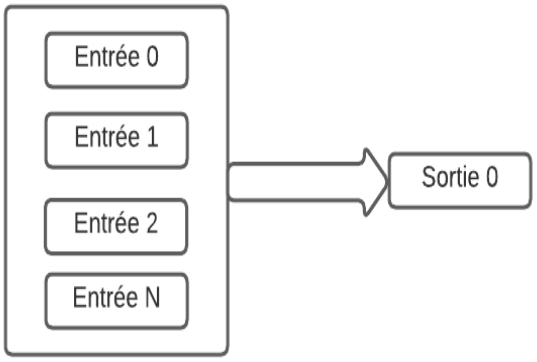

FIGURE 1.12 – Transaction avec plusieurs entrées

- **Transaction avec plusieurs sorties**
  La transaction avec plusieurs sorties, diffuse l'entrée à plusieurs sorties, qui au nom de plusieurs destinataires

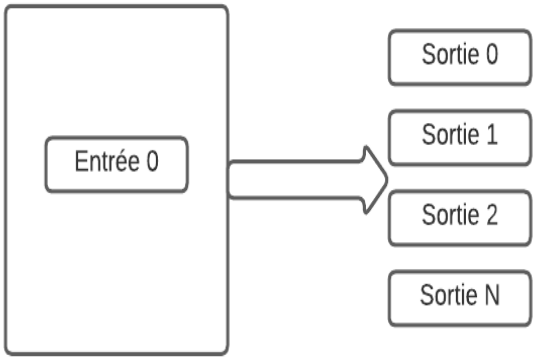

FIGURE 1.13 – Transaction avec plusieurs sorties

1.4.2 Bloc

Un bloc est un fichier(enregistrement) dans la blockchain, qui regroupe plusieurs transactions validées par des noeuds du réseau blockchain. Chaque bloc validé est ajouté à la blockchain. Le bloc est constitué de deux parties, l'entête du bloc et le corps du bloc. Comme la figure 1.14 montre. [1]

1.4.2.1 Entête du bloc

Qui contient les informations suivantes : [8]

- **La valeur de hachage du bloc précédent** (32 octets).
- **Horodatage :** Le temps de la création du bloc (4 octets).
- **nBits :** Le seuil cible d'un hachage de bloc valide.
- **Le nonce :** La valeur utilisée pour la preuve de travail, il commence la plus par du temps par 0 (4 octets).
- **Racine de Merkle :** La valeur de hachage de toutes les transactions dans le bloc.

1.4.2.2 Corp du bloc

Le corp contient un compteur et une liste de transaction, dont le nombre maximum de cette dernière qu'un bloc peut contenir, dépend de sa taille et de la taille de chaque transaction. [8]

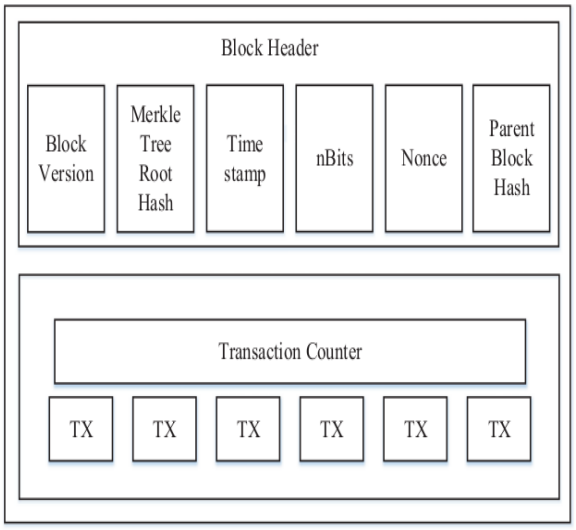

FIGURE 1.14 – la structure d'un bloc
[2]

1.4.3 Cryptographie dans la blockchain

Dans la technologie blockchain, la cryptographie (fonctions de hachage, cryptographie asymétrique, ..) est utilisée pour assurer la sécurité du réseau, l'integrité et l'authenticité des transactions.

- **Fonction de hachage**
  La fonction de hachage est une méthode qui permet de créer une empreinte unique d'une certaine donnée. S'il y a une modification de cette dernière, le hash sera totalement différent. La blockchain utilise un double SHA256 comme fonction de hachage. Ces fonctions sont utilisées dans la blockchain pour : la diffusion des addresses, la

sécurisation des données du bloc ainsi que pour la construction d'entête du bloc. [1]

- **Cryptographie asymétrique**
  La cryptographie à clé asymétrique permet de générer une paire de clé. En effet la clé privé sert à signer les transactions, alors que la clé publique est utilisée pour la dérivation des addresses et la vérification des signatures. [1]

1.4.4 consensus

Le système de la blockchain est un système décentralisé, qui est un problème pour tous les réseaux décentralisés, dont les noeuds ne se connaissent pas et ne se font pas confiance l'un à l'autre. Donc ils doivent se mettre d'accord sur une valeur identique de la blockchain. et pour cela un algorithme de consensus est utilisé. Ce dernier est un procèdé qui permet la validation, la sécurisation et la mise à jour de la blockchain. Son but est de se mettre sur une version unique du réseau. Il en existe plusieurs algorithmes de consensus, nous allons citer les plus connus et les plus utilisés. [4]

- **Preuve de travail (PoW)**
  Connu sous le nom de Proof of work, est parmi les algorithmes de consensus les plus connus. C'est l'algorithme utilisé par Bitcoin. Son principe est que les mineurs doivent résoudre un calcul cryptographique compliqué en dépensant de l'énergie qui est demandée par le protocole, dans le but de pouvoir publier un nouveau bloc. La résolution de ce problème est la preuve qu'il a fait un travail. [1]

- **Preuve d'enjeu (PoS)** C'est un autre protocole de consensus, qui ne se base pas sur la nécessité de puissance de calcul comme la preuve de travail. Dans la preuve d'enjeu, les mineurs sont obligés de risquer leur argent dans le réseau, afin de pouvoir créer de nouveaux blocs. Peercoin a été le premier qui a implémenté cet algorithme. [8]
  Il existe d'autres algorithmes de consensus comme : preuve tolérance aux pannes byzantines (BFT), Preuve d'activité (PoA), preuve d'enjeu déléguée (DPoS), preuve de publication (PoP), preuve de capacité (PoC), etc.. [2]

1.4.5 Contrat intelligent (Smart contracts)

Un smart contract ou contract intelligent, est un contrat écrit en un langage informatique, et stocké sur la blockchain, il s'exécute automatiquement lorsque des conditions sont remplies, sans l'intervention d'une tière partie. Ce genre de contrat est plus évloué que celui en version papier.
Selon Nick Szabo, " Un Smart contract est un protocole de transaction informatisé qui exécute les termes d'un contrat. Les objectifs généraux de la conception d'un contrat intelligent sont de satisfaire les conditions contractuelles courantes (telles que les conditions de paiement, les privilèges, la confidentialité et même l'exécution), de minimiser les exceptions, tant malveillantes qu'accidentelles, et de minimiser le besoin d'intermédiaires de confiance. Les objectifs économiques connexes comprennent

la réduction des pertes dues à la fraude, des coûts d'arbitrage et d'exécution et des autres coûts de transaction ". [20]

Les contrats intelligent passent par un cycle de vie qui est composé de 4 phases. La figure 1.15, montre ces phases.

- **Création d'un contrat intelligent**
  La phase de création d'un contract intelligent, est un procèdé itératif où, les personnes concernées par ce contract vont négocier sur les droits et obligations qui vont être mis dans le contract. Dans des cas, les avocats aident à conclur un accord, qui va être ensuite converit par des ingénieurs d'un language naturel vers un contract intelligent écrit en un language informatique. La procédure de conversion comprends la conception et la mise en oeuvre. [18]

- **Déploiement de contrat intelligent**
  Dans cette phase, les contrats intelligent seront déployées sur des platforms de blockchain lorsqu'ils seront validées. Une fois le contract est déployée sur la blockchain, les personnes concernées peuvent accèder à leurs contract par l'intermidaire de la blockchain. Les contrats stockées dans le bloc , ne peuvent pas être modifiés, si un individu veut modifier cela l'oblige à créer un nouveau contrat. [18]

- **Éxecution de smart contract**
  Après avoir déployer le smart contract, une évaluation sur les conditions contractuelles est effectuée. Un smart contract est un ensemble de conditions contracutelles. Une fois la condition est déclenchée, l'instruction qui lui correspond s'exécute automatiquement, ce qui va résulter l'exécution d'une transaction qui sera validée par les mineurs et stockée dans la blockchain ainsi que les états mis à jour. [18]

- **Completion de smart contract**
  Lorsque les smart contracts sont exécutés, les transactions et les états mis à jour sont stockées dans la blockchain. Ce qui veut dire que les actifs numériques sont dévérouillées et transférées d'un compte à un autre. Dans ce cas, le contract intelligent a fini son cycle de vie et il faut noter que des transactions ont été effectuées c'est à dire que des données sont écrites dans la blockchain. [18]

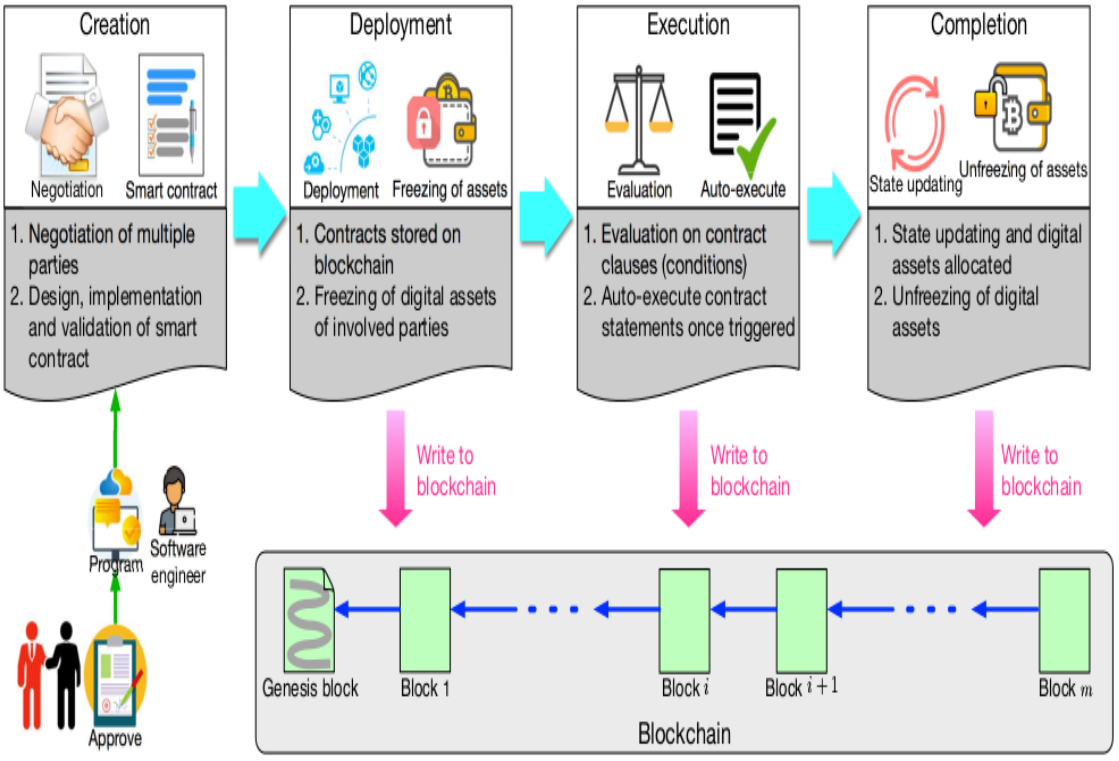

FIGURE 1.15 – Cycle de vie d'un smart contracts
[18]

1.5 Caractéristique de la blockchain

La technologie blockchain repose sur les caractéristiques suivantes :

- **Décentralisation**
  La principale caractéristique de la blockchain est la décentralisation, c'est à dire qu'aucune autorité centrale n'a le droit de contrôle, et les noeuds du réseau ont tous la même autorité, ces derniers se chargent de conserver les données de la blockchain. Cela a permis de mettre point aux risques liés aux systèmes centralisés notament les fraudes. Cependant il peut y avoir des blockchains privées sont purement ou partiellement centralisées. [2, 8]

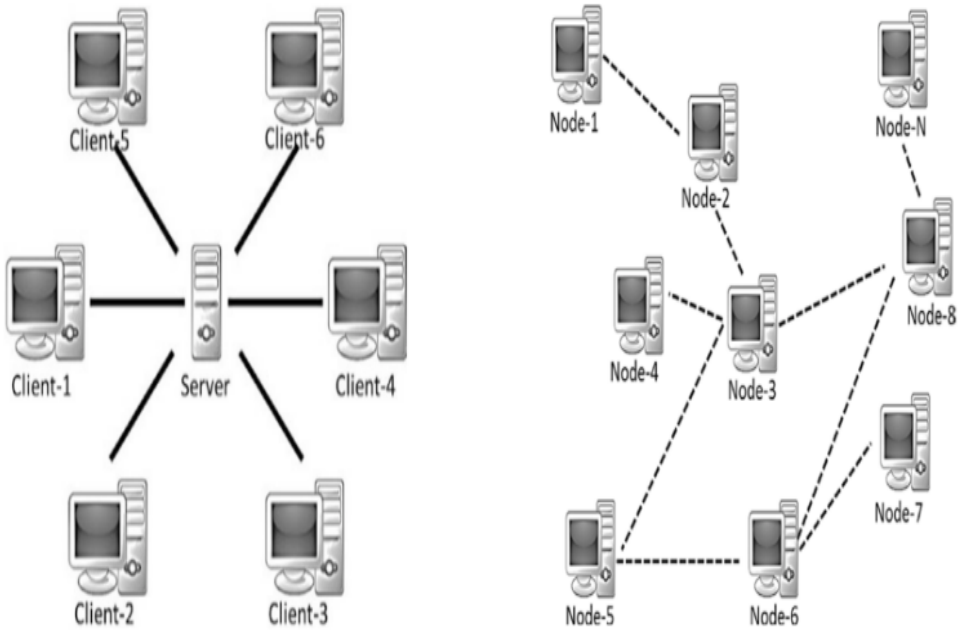

FIGURE 1.16 – Système centralisé vs système décentralisé
[5]

- **Sécurité**
  La blockchain est dite sécurisée, car il est difficile de détruire le réseau à cause des primitives cryptographiques. Pour détruire le réseau, il faut modifier tous les données de tous les blocs pour tous les noeuds du réseau. [2]

- **Transparence**
  La transparence de la technologie blockchain est dûe à la notion de l'horodatage. Vu que les blocs de la blockchains sont horodatés et stockés sur des noeuds dans le réseau, donc les transactions peuvent être vérifiées et vues par tout ce qui vont accèder au réseau, ce qui rend la blockchain transparente et traçable. [2]

- **Réduction des coûts**
  La plus part des entreprises veulent utiliser la blockcain, à cause de la réduction des coûts qu'elle offre, notamment celles liées aux systèmes centralisés. [2]

1.6 Types de blockchain

Il existe trois types de blockchain, qui sont au point commun d'un consensus, et utilisent le protocole de communication pair à pair.

1.6.1 Blockchain publique

Une blockchain publique, est une blockchain sans adhésion, c'est à dire que les utilisateurs n'ont pas besoin d'avoir une autorisation pour accèder au réseau, n'importe qui peut être un noeud sur le réseau, un mineur ou même un validateur. Donc ils peuvent lire,

écrire librement et particper à leurs consensus et même de voir quel est le dernier bloc qui a été ajouté à la chaine. Les blockchains publiques sont des réseaux décentralisés, à l'inconvénient qu'ils sont fragiles aux minage égoïste [1] ainsi qu'au problème de confidentialité. Parmi les blockchain publique connus nous pouvons citer : Bitcoin et Ethereum. [1, 2]

1.6.2 Blockchain privée

Une blockchain privée, est un réseau de blockchain avec permission, c'est à dire il y a une entité qui contrôle le réseau. Les enregistrements de ce réseau sont invisibles par le public, alors les utilisateurs sont obligés d'avoir l'autorisation d'accès à ce réseau pour qu'ils puissent manipuler la blockchain. Ce type de blockchain a la possibilité aussi de limiter l'accès de lecture ou écriture aux utilisateurs autorisés, c'est à dire quiconque ne peut pas être un mineur ou un validateur, à son tour ce type ne présente pas de problème de confidentialité ou de minage egoïste. Parmi les blockchains privées, il y en a Multichain et Blockstack. [1, 2]

1.6.3 Blockchain de consortium

Une blockchian consortium, appelée aussi blockchain hybride, c'est un assemblage des deux types de blockchain vu au préalable. Ce sont des réseaux ou seuls les membres du consortium peuvent participer. Dans les blockchains hybrides, des noeuds préséléctionnés créent les nouveaux blocs et valident les transactions, alors que les autres noeuds s'occupent seulement de l'envoie des transactions, la lecture et la vérification des nouveaux blocs. Nous citons Corda et Hyperledger comme exemples de blockchain de consortium. [2]

1.7 Application de blockchain

La blockchain a été utilisée dans diverse domaines, mais elle a connu un énorme succès dans le domaine financier. Dans cette section nous allons présenter quelques domaines ou la blockchain à été utilisée.

1.7.1 Bitcoin

Bitcoin a été la première cryptomonnaie inventée par Satoshi Nakamoto, où il a expliqué le principe de cette invention dans son livre blanc intitulé ' Bitcoin : A Peer-to-Peer Electronique Cash System '. Le nom Bitcoin est la concaténation des deux mots anglais, "bit", qui est une unité de mesure binaire et "coin", qui veut dire pièce de monnaie. Cette cryptomonnaie est basée sur une blockchain publique. [5]

1.7.2 Vote

L'utilisation de la blockchain dans le vote, permet d'assurer la transparence de ce dernier et d'organiser des éléctions libres et équitables, notamment dans les pays qui sont en développement. [2]

1. Une statégie dont son principe est qu'un mineur garde les blocs qu'il a découvert privée ce qui provoque la division intentionnelle de la chaine. [3]

1.7.3 Gestion de l'identité numérique

Malgré l'évolution des modèles de gestion d'identité numérique, ils provoquent beaucoup de vulnérablité comme la perte, le vol et le fraude d'identité. L'utilisation de la blockchain pour la gestion d'identité numérique est une nouvelle solution qui réponds à ces défis. Nous allons détailler le principe de cette application dans le chapitre qui suit [2].

1.7.4 Cybersécurité

L'utilisation de la blockchain peut renforcer la cybersécurité contre les attaquants. Le stockage basé sur la blockchain, permet la sécurisation des services contre les attaques comme le déni de serice distribuée (DDoS) [2] et d'autres. Il existe des entreprises et des startups qui utilisent la blockchain pour la cybersécurité, on peut citer les suivantes : openAVN, block armor, Cryptyk, Sentinel Protocol et Megahoo [2]

1.7.5 IoT

La combinaison de la blockchain avec l'Internet des Objet(IoT), a pour but d'intégrer des obejcts intelligents dans l'internet. L'utilisation de la blockchain dans l'IoT, peut offrir, le partage évolutif des ressources dans IoT, la mise en œuvre de la sécurité de l'IoT, et des véhicules aériens sans pilote. [19] Parmi les applications, qui ont introduit la blockchain dans l'IoT, citons : ADEPT, Filaments, GSF et ShareCharge d'IBM. [2]

1.8 Plateformes de blockchain

Il existe de nombreuses plateformes de blockhain pour développer des applications. Ces plateformes peuvent être divisées en publique et privées. Le choix de ces plateformes dépends des besoins et facteurs spécifiques de l'entreprise. Dans cette section nous allons ouvrir les portes sur certaines plateformes, nommées, Ethereum, Hyperledger Fabric, Corda, Stellar, Solana.

1.8.1 Ethereum

Ethereum est une plateforme de blockchain publique, open source, dévéloppée par Vitalik Buterin [3]. Cette plateforme permet l'exécution des smart contracts sur des machines virtuelles. Ethereum utilise la preuve de travail (PoW) comme algorithme de consensus, et il possède sa propre cryptommonaie Ether. L'Ether est une cryptomonnaie utilisée pour récomponser les mineurs lorsqu'ils valident un bloc, elle peut être transférée d'un portefeuille à un autre. Ethereum utilise Gas comme mécanisme de tarification conçu pour mesure combien de ressources une transaction peut consommer lors l'exécution d'un smart contract. Le coût du Gas représente le budget nécéssaire pour l'exécution du contrat. La limite du Gaz représente le prix maximum qu'un utilisateur est prêt à le payer pour exécuter un smart contract [18, 19, 22]. Ces dernières sont exécutées sur la machine virtuelle

2. DDoS, est une attaque dans laquelle plusieurs systèmes sont regroupés contre un système afin de le paralysé. [55]

3. Un programmeur russe, canadian

d'Ethereum. Cette plateforme supporte les languages de programmation suivant pour le développement des smart contracts :

- **Solidity**
  Solidity est un language orienté objet haut niveau, conçu pour le développement des smart contracts. Solidity est un language à accolades, et influencé par C++, Python et Javascript, il est élaboré pour cibler la machine virtuelle Ethereum.
- **Vyper**
  Vyper est un language de programmation pythonique conçu pour développer les smart contracts.

1.8.2 Hyperledger Fabric

Hyperledger Fabric est une plateforme privée élaborée pour qu'elle soit utiliée par les entreprises, elle permet l'exécution des smart contracts. Contrairement à Ethereum, Hyperledger n'utilise pas les machines virtuelles pour exécuter les smart contracts mais il utilise plutôt des conteneurs Docker. La plateforme Hyperledger utilise la tolérance aux fautes byzantine pratique (PBFT) comme algorithme de consensus. Elle supporte Java, NodeJs et GoLang comme language de programmation pour développer les smart contracts qui sont appelés ChainCode dans sa terminologie [18, 19]

1.8.3 Corda

Corda est une plateforme privée, elle permet de déployer des contrats. Corda utilise Raft comme algorithme de consensus. Cette platforme est utilisée dans de diverses applications et beaucoup plus utilisée dans le domaine financier. [18, 19]

1.8.4 Stellar

Stellar est une plateforme privée designée beaucoup plus pour les applications de monnaies numériques. Elle utilise sa prope cryptomonnaie appelée, Lumen. Cette plateforme est similaire à Hyperledger Fabric, car elle utilise des conteneurs Docker pour executer ses programmes, et elle utilise son propre algorithme de consensus, Stellar Consensus Protocol. Stellar supporte plusieurs languages de programmation telque : Python, PHP, JavaScript, GoLang [18, 19]

1.9 Avantages et inconvéninents de blockchain

Dans cette section, nous allons cibler les avantages et inconvéninents de la blockchain.

1.9.1 Avantages de la blockchain

Dans ce qui suit, nous allons citer les avantages de la blockchain.

- **Décentralisation**
  L'avantage principale de la blockchain est qu'elle est basée sur la décentralisation, c'est à dire la non-nécssité d'une autorité centrale pour le contrôle de réseau. Les noeuds complets du réseau ont tous la même autorité, et qui se chargent de conservés les données de la blockchain. Ce qui permet de limiter les risques liés aux systèmes centralisés notament les fraudes. Cependant il peut y avoir des blockchains privées qui sont purement ou partiellement centralisées. [4]

- **Rapidité de traitement**
  L'utilisation de la technologie blockchain offre une rapidité dans le traitement et l'initialisation des transactions qui prennent beaucoup de temps dans les organisations bancaires. [4]

1.9.2 Inconvéninets de la blockchain

La blockchain présente des avantages et des inconvénients, dans cette partie nous allons citer quelques inconvénients propres à cette technologie.

- **Consommation de l'energie**
  Le principale inconvéninet de la blockchain est la forte consommation d'énergie. En effet cette consommation énergitique dépend du type de consensus utilisé, comme l'algorithme de concensus preuve de travail dont lequel les mineurs cherchent à valider des transactions, ce qui nécéssite une importance puissance informatique, et du coup une très grande consommation d'énergie. [4]

- **Division de blockchain**
  Lorsque les noeuds valident un même bloc, en même temps, cela peut engendrer une bifurcation de la chaîne en deux branches, longue qui va être considérée comme la chaîne principale, et la courte sera abandonnée, comme montre la figure 1.17 [4]

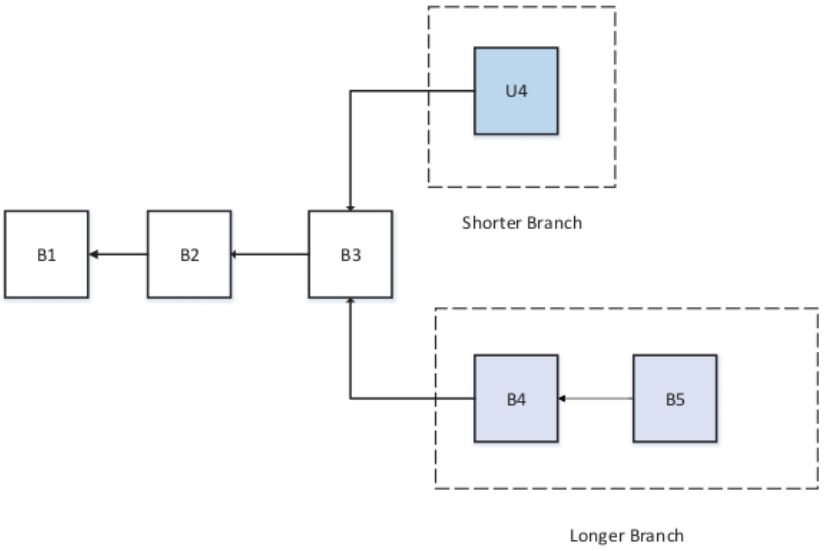

FIGURE 1.17 – Bifurcation de la chaîne.
[8]

1.10 Blockchain et stockage des données volumineuses : Blockchain et IPFS

Il y a un grand intérêt dans de nombreuses approches vers la blockchain pour fournir une solution afin d'enregistrer les transactions de manière décentralisée. Cependant, il existe certaines limitations lors du stockage de fichiers ou de documents volumineux sur la blockchain. Afin de répondre aux exigences de stockage de telles données, il est nécessaire de recourir à un support de stockage décentralisé.. Une des nouvelles solutions qu'on peut citer dans ce contexte est IPFS (**InterPlanetary File System**). [58]

1.10.1 Système de fichiers interplanétaires (IPFS : InterPlanetary File System)

IPFS est un système de fichiers distribué adressable par le contenu [58], présenté pour la première fois par le programmeur Juan Benet en 2015. IPFS permet à ceux qui en font partie de stocker et de diffuser des informations de façon décentralisée sur le réseau d'Internet. La version actuelle d'IPFS, bien qu'il s'agisse d'une version de développement, elle permet de déployer un ensemble de ses fonctions finales de manière stable.

- Le système IPFS est basé sur une architecture pair-à-pair (P2P), sur laquelle chaque nœud joue le rôle de client et de serveur. Chaque nœud peut accéder et distribuer des données.
- L'accès à une ressource sur IPFS ne se base pas sur sa localisation, mais sur son contenu. Chaque ressource est représentée par un identifiant unique, défini par son contenu (CID ou Content Identifier).

- Si personne ne connaît l'identifiant d'un document, personne ne pourra y accéder. Un tel document ne peut pas être supprimé d'IPFS s'il y a au moins un nœud du réseau qui le rend disponible
- IPFS utilise une structure de données basée sur des graphes orientés acycliques (DAG), plus précisément des arbres de hachage (arbres de Merkel), utilisé aussi par Blockchain. Ce concept est utilisé pour identifier les fichiers et les répertoires. Si un fichier contenu dans un répertoire est modifié, l'empreinte (donc l'identifiant) de celui-ci ainsi que celle du répertoire changent. Toute modification est alors facilement détectée.

1.10.2 Adressage de contenu

Dans ce type d'adressage utilisé par IPFS, Chaque fichier est identifié par une empreinte cryptographique unique, appelée content identifier (CID). Cette empreinte est calculée par une fonction de hachage à partir des données contenues dans le fichier, ce qui permet d'obtenir une signature numérique d'un fichier et vérifie son intégrité.

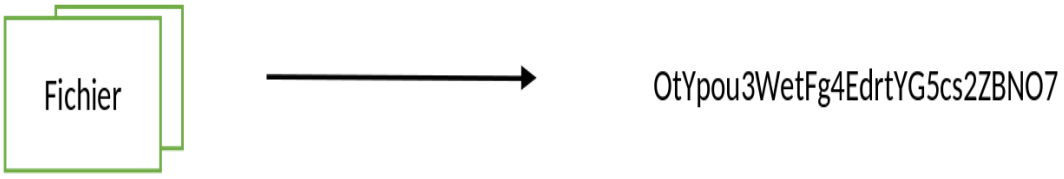

FIGURE 1.18 – Exemple d'un fichier et son CID.

1.10.3 Avantages et inconvénients d'IPFS

Quelques avantages de l'IPFS :

1. Le système de stockage est distribué.
2. Le système est entièrement décentralisé, et il résiste aux attaques par déni de service.
3. L'accès rapide aux informations est garanti à tout moment.
4. Le code source est disponible sous licence de logiciel libre, et l'utilisation de système est gratuite.
5. Le réseau est évolutif et extensible, et des nouvelles fonctions peuvent être adaptées.

Par contre on peut citer les inconvénients suivants :

1. Il est encore à la phase de développement et d'évolution, son utilisation est encore limitée.
2. Il est complexe à utiliser pour les utilisateurs qui n'ont pas d'expériences dans ce type de systèmes.
3. Il n'a pas d'extensions de confidentialité par défaut.

1.11 Conclusion

La technologie blockchain est une technologie moderne, qui ne nécéssite pas de système de contrôle central mais elle assure la transmission des données d'une façon sécurisée. La blockchain est une suite de blocs qui sont chainés entre eux avec des concepts cryptographiques, ces derniers permettent d'assurer la sécurité de cette technologie. Les blocs stockent l'historique des différentes transactions effectuées et validées depuis la création de la blockchain. Dans ce chapitre nous avons présenté la techonologie blockchain, ses composants, son fonctionnement par rapport à une cryptomonnaie, on a aussi cité les différents types de blockchains qui existent. Nous avons aussi expliqués les caractéristiques de la blockchain, ses avantages et inconvéninets, et quelques applications de la blockchain comme la gestion d'identité numérique. Une présentation détaillée de principe et modèles de cette dernière est l'objectif du prochain chapitre.

CHAPITRE 2

GESTION D'IDENTITÉ NUMÉRIQUE

2.1 Introduction

L'évolution de la technologie de l'information et de la communication permet à notre société de s'envoler vers un monde numérique ou tout est informatisé. Cette numérisation permet aux gens d'accéder à des services et des applications à partir d'une identité numérique où ils fournissent leurs propres donnés personnels à un fournisseur d'identité qui est chargé de leur autoriser l'accès à ces services. Dans ce chapitre, nous allons expliquer c'est quoi la gestion d'identité numérique, quelles sont ses modèles depuis l'apparition d'Internet. Nous allons parler aussi des systèmes de gestion d'identité numérique auto-souveraine, en présentant ceux les plus connus.

2.2 Définitions

Dans cette section, nous allons présenter quelques définitions.

- **Identité**
  Selon le dictionnaire larousse, une identité est un "Caractère permanent et fondamental de quelqu'un, d'un groupe, qui fait son individualité, sa singularité".

- **Identité numérique**
  L'identité numérique est l'ensemble des données confidentiels et personnel qui permet d'identifier un individu sur Internet et qui peuvent être gérées par un système. [23]

- **Gestion d'identité**
  La gestion des identités, appelée également, gestion des identités et d'accès, est un mécanisme qui touche l'identification, l'authentification ainsi que l'autorisation des individus pour accéder aux systèmes et réseaux. [23]

- **Fournisseurs d'identité**
  Un fournisseur d'identité, est un partenaire qui certifie l'identité des utilisateurs. Il

authentifie un utilisateur et diffuse son jeton d'authentification à un founisseur de service. [56]

- **Fournisseurs de service**
  Un fournisseur de services, est un partenaire qui offre des services à des utilisateurs,dont leur identité est certifiée par un fournisseur d'identité. [56]

2.3 Architecture typique d'un système de gestion d'identité

Dans cette section, nous présentons une architecture basique d'un système de gestion d'identité, dont elle est composée de trois tiers : l'utilisateur, fournisseur d'identité et fournisseur de service. Pour simplifier, considérons un scénario où un utilisateur demande une preuve d'identité au près d'un fournisseur d'identité (voir la figure 2.1) [56]

- **Utilisateur**
  L'utilisateur est l'un des principaux acteurs du système et qui bénéfice des différents services offerts par le fournisseur de services et le fournisseur d'identité. Les utilisateurs n'ont pas tous le même privilège. [56]
- **Fournisseur d'identité**
  Le fournisseur d'identité, est le cœur du système. Il a pour mission de fournir aux utilisateurs des services d'identité (par exemple, l'enregistrement, l'authentification et la gestion). Cet acteur assure également l'authentification des utilisateurs. [56]
- **Fournisseur de service**
  Le fournisseur de services est une partie importante du système, et est principalement chargé de fournir des services aux utilisateurs (une fois qu'ils sont authentifiés avec succès). [56]

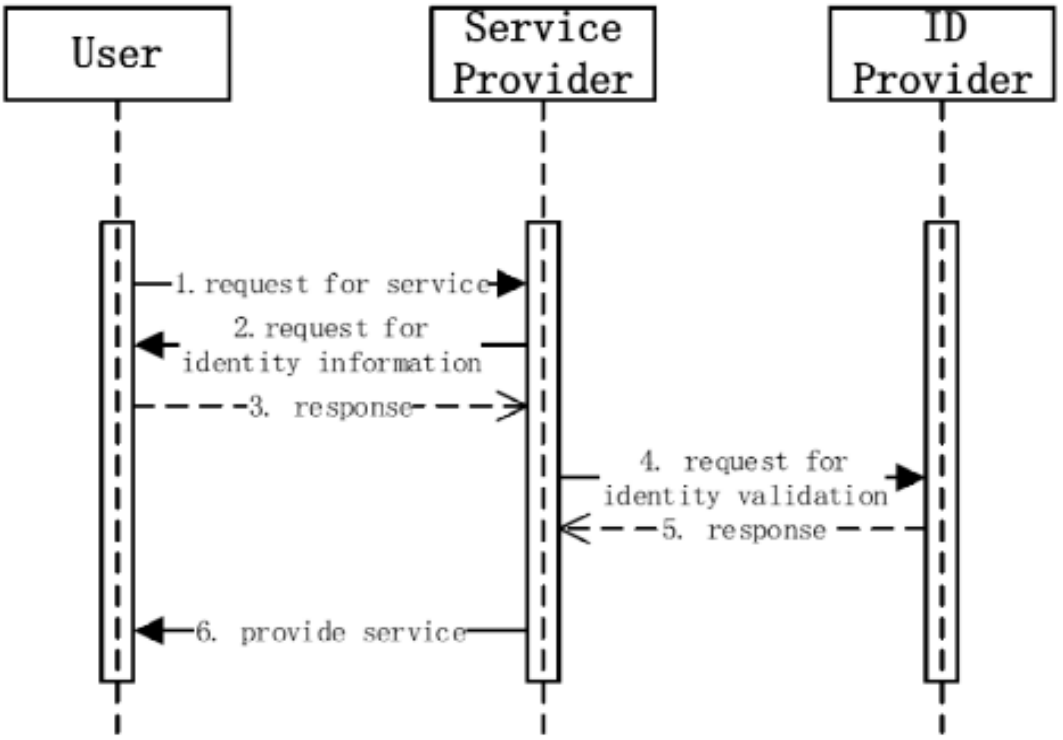

FIGURE 2.1 – Fonctionnement typique d'un système de gestion d'identité [56]

2.4 Modèles d'identité numérique

Avec l'apparition de l'Internet et le monde numérique, différentes identités sont progressées avec le temps. Dans cette section, nous allons voir les modèles d'identité numérique, et détailler le fonctionnement du modèle d'identité auto-souveraine.

2.4.1 Identité centralisée

L'identité centralisée est la première forme d'identité qui a été apparue avec la naissance de l'Internet. C'est une identité largement utilisée dans le monde. Cette forme est gérée par une autorité, où l'utilisateur doit avoir un identifiant d'identité numérique accordé par l'organisation dont il veut accéder à ses services. Donc l'utilisateur est obligé d'avoir un nouvel identifiant pour chaque organisation avec laquelle il s'engage, ce qui rend difficile de suivre le grand nombre d'identité. [26] La figure 2.2, montre l'architecture de cette identité.

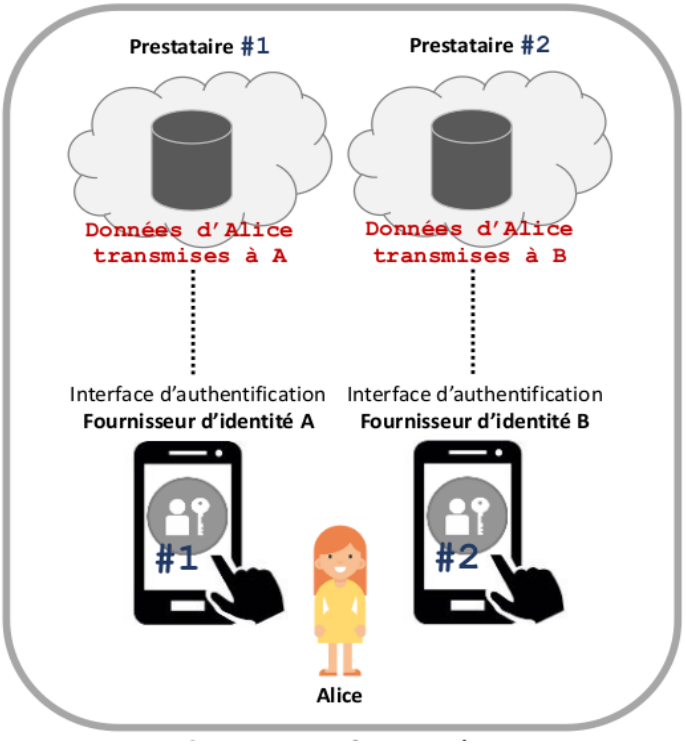

FIGURE 2.2 – Architecture d'identité centralisée [29]

2.4.2 Identité fédérée

A cause de la difficulté de suivre les identités de la première forme, les organisations ont pensé à une nouvelle forme d'identité appelée l'identité fédérée. C'est une méthode qui permet de stocker l'identité numérique d'un individu dans plusieurs systèmes de gestion différents. Elle est contrôlée par diverse autoritées fédérées. Microsoft Passport est le premier exemple qui permet aux utilisateurs, grâce à une authentification unique, d'accéder aux services de l'ensemble des sites web affiliés à Microsoft. [23, 26]

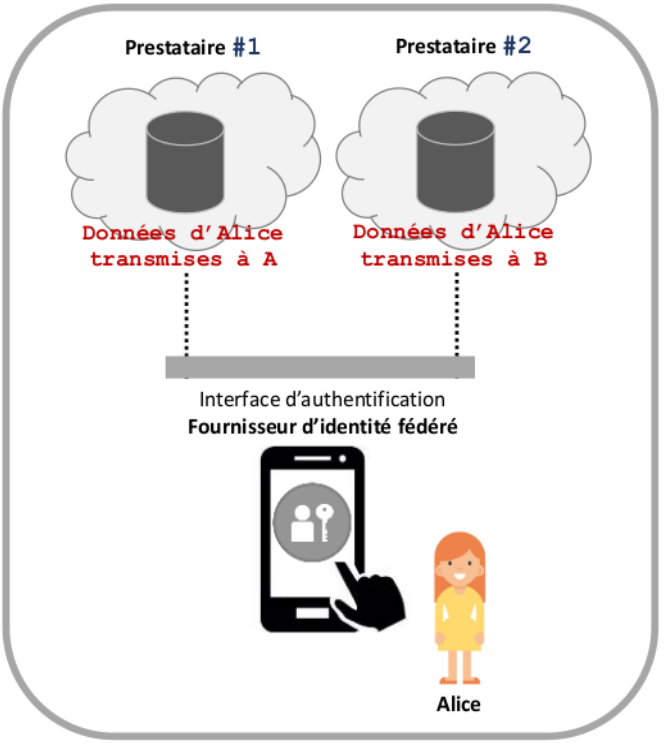

FIGURE 2.3 – Architecture d'identité fédérée [29]

2.4.3 Identité centré sur l'utilisateur

La troisième forme de l'identité est l'identité centré sur l'utilisateur. C'est une identité contrôlée par un individu, à travers plusieurs autorités, sans nécessiter de fédération. Le modèle d'identité repose sur l'hypothèse que chacun a le droit de contrôler son identité en ligne. Un utilisateur peut théoriquement enregistrer son propre Open ID, qu'il peut utiliser indépendamment. [23]

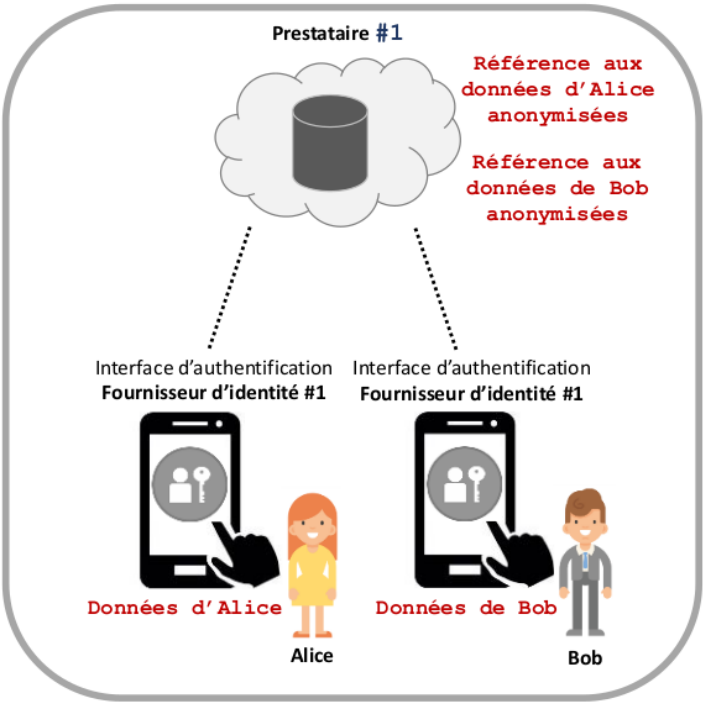

FIGURE 2.4 – Architecture d'identité centrée sur l'utilisateur [29]

2.4.4 Identité auto-souveraine

Malgré les améliorations apportées par les modèles de gestions d'identité précédents, ils restent toujours des modèles centralisés, ainsi que l'utilisateur n'a pas un vrai contrôle sur ses données d'identification. Ces caractéristiques sont assurées par le nouveau modèle de gestion d'identité auto-souveraine. L'identité auto-souveraine (en anglais : Self-Sovereign Identity), est une démarche qui décrit comment un individu peut avoir le droit de contrôler et gérer son identité numérique sans l'intervention d'une autorité [30]. La blockcchain est un outil très promoteur pour la réalisation des modèles de gestion d'identité auto-souveraine. [24, 25]

Selon [43], une identité, est dite auto-souveraine si elle répond a dix principes proposés par Christopher Allen, un développeur et spécialiste des normes et de l'identité.

Dans ce qui suit nous allons résumer ces dix principes :

- **Existence**
  L'identité de l'utilisateur doit être connexe a ses caractéristiques, il ne doit pas y avoir des collisions d'identifiants.
- **Contrôle**
  Seul l'utilisateur qui peut contrôler chaque attribut de son identité.
- **Accès**
  L'utilisateur doit pouvoir accèder et récupérer ses données facilement.
- **Transparence**
  Les systèmes et algorithmes utilisés doivent être transparents, gratuits et a code source ouvert.

- **Persistance**
  L'identité doit être pérennisé, ou au moment que l'utilisateur le souhaite.
- **Portabilité**
  La capacité de l'identité s'adapte plus ou moins facilement pour fonctionner dans différentes plateformes.
- **Interopérabilité**
  L'utilisation de l'identité ne doit pas être limitée, mais plutôt utilisée largement que possible.
- **Consentement**
  L'utilisateur doit accepter de partager ses données.
- **Minimalisation**
  La minimalisation des données signifie que seules les données nécéssaires pour accéder a un services doivent être exposées.
- **protection**
  Les données de l'identité doivent être protégées en cas de danger.

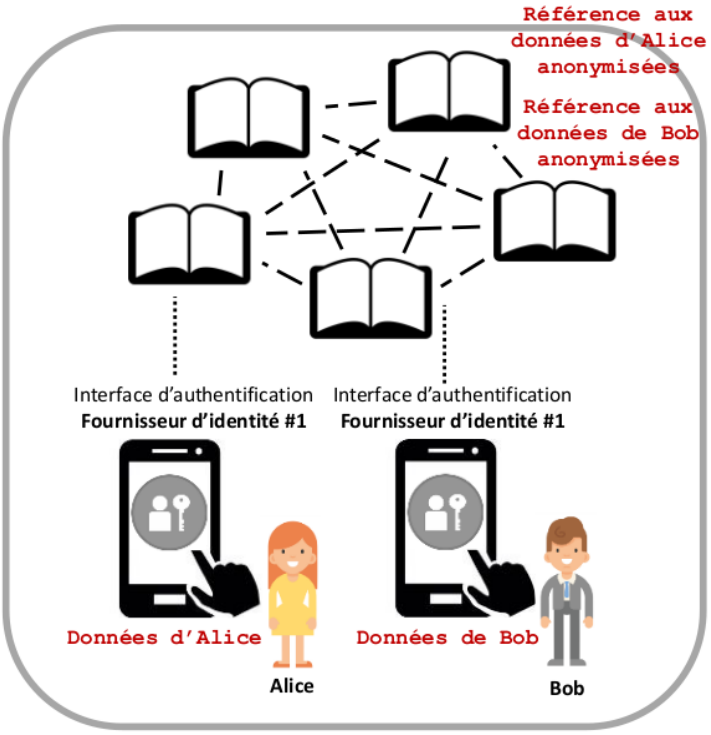

FIGURE 2.5 – Gestion d'identité auto-souveraine [29]

2.5 Etat de l'art

Dans cette section, nous allons exposer des travaux présentés dans des articles scientifiques comme tentatives de propositions des systèmes de gestion d'identité en utilisant la blockchain. Puis, nous présentons des exemples des systèmes réels, déjà réalisés et utilisés.

2.5.1 Quelques travaux de la littérature

- Dans [27]. Les auteurs proposent le système **DNS-IdM**, qui est une nouvelle architecture du système de gestion d'identité basé sur les smart contracts. Cette architecture permet aux utilisateurs d'avoir le contrôle de leur identités numérique. Son objectif est d'assurer la sécurité et la confidentialité ainsi que de permettre a une tièrce partie d'identifier les utilisateurs. Le concept de ce système peut être mis en oeuvre par les utilisateurs qui créent de nouveaux comptes sur un réseau Ethereum privé. Chaque utilisateur introduit un ensemble d'attributs formant son identité qui sont stockés dans le système de fichier IPFS(InterPlanetary File System). Ces attributs sont validés par des contracts spécialisés stockés sur un autre réseau blockchain sans permission. L'objectif de cette validation est d'éviter que l'utilisateur ajoute de fausses données ou de négliger de faux attributs. L'utilisateur doit permettre à l'autre partie d'accèder à un attribut en fonction du type de service qu'elle l'offre, cela peut se réaliser lorsque l'autorité récupère les données du validateur et vérifie l'authenticité de ces dernières en validant la signature.

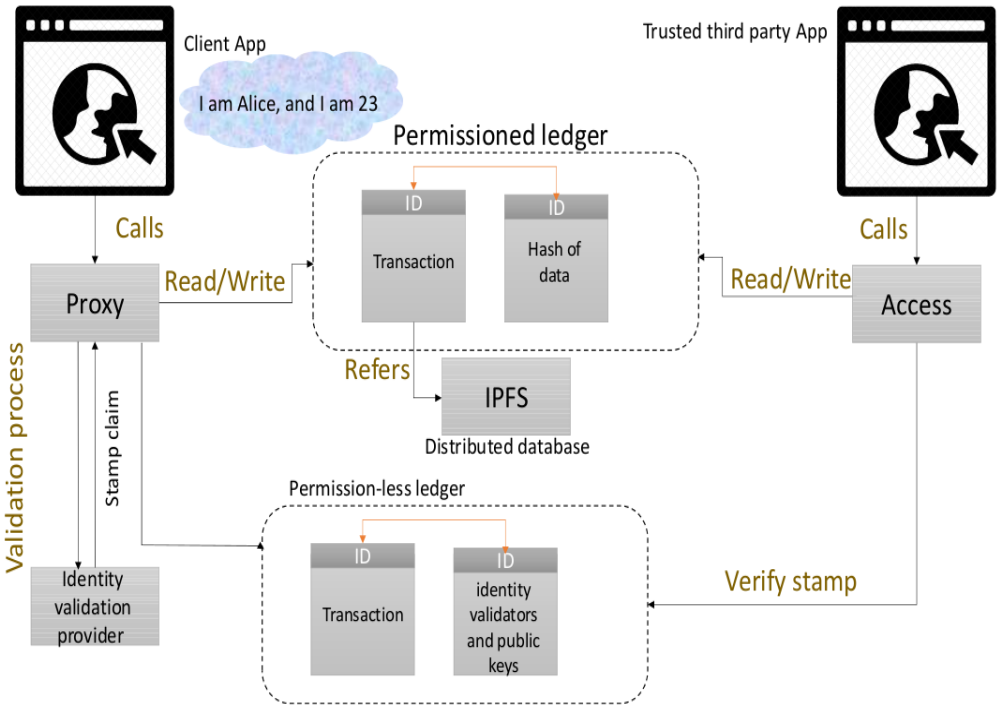

FIGURE 2.6 – Architecture DNS-IdM
[27]

- Dans [34], les auteurs ont proposés de faire des **Self-Sovereign Biometric IDs** (SelfIs), une idée, qui consiste à fusionner les concepts de décentralisation avec ceux de la biométrie afin de développer une solution qui permet aux utilisateurs de contrôler la manière dont leurs données biométriques sont utilisées. Le concept de cette proposition se déroule selon 10 étapes selon un exemple d'un utilisateur qui veut voyager à l'étranger, dans ce qui suit on va expliquer ces étapes.

  - L'utilisateur crée son SelfIs qui va l'enregistrer à un backend de confiance. Il

peut utiliser une application mobile pour le faire. Dans ce cas l'utilisateur doit enregistrer son identitée auprès de la compagnie aérienne (Cloud1).

- L'utilisateur enregistre son SelfIs, ainsi que d'autres métadonnées, auprès de la compagine aérienne.
- Les SelfIs sont ajoutées à la blockchain et à un backend d'apprentissage automatique(Cloud2), pour que l'utilisateur puisse être authentifier plutard.
- Un utilisateur qui veut réserver un vol, la compagnie aérienne lui crée des informations de réservation nécéssaire.
- Ensuite, l'utilisateur se rend au point d'accès pour l'enregistrement.
- Le point d'accès génére le Selfis correspondant, à partir des métadonnées qui sont dans le Cloud 1.
- Dans cette étape, le kiosque automatique, lance un appel au backend qui se trouve au Cloud2, pour effectuer une correspondance des SelfIs.
- La logique de correspondance est déclenchée.
- Une réponse d'autorisation est envoyé au point d'accès.
- Dans ce cas le processus d'enregistrement est poursuivie.

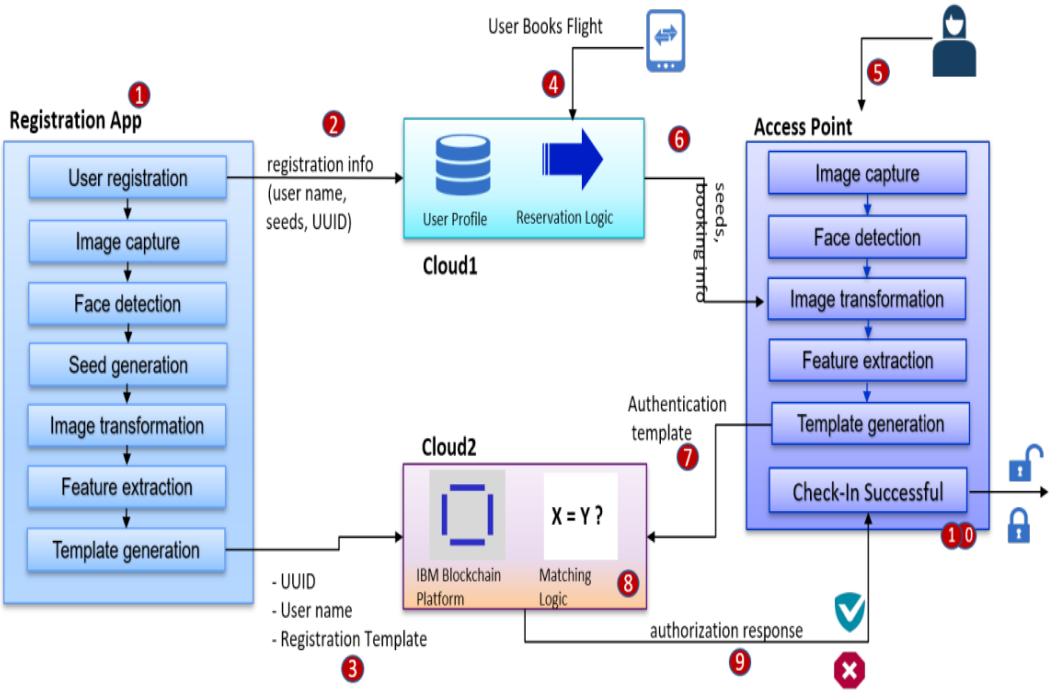

FIGURE 2.7 – Architecture SelfIs
[34]

- Dans [35], un framework basé sur la blockchain publique et autorisé Hyperledger Indy est proposé. Il permet d'identifier les appels de services bancaires frauduleux et éviter la perte éventuelle de comptes personnels. Ce framework permet aussi d'améliorer la confiance entre les parties concernés et de renforcer le processus de

connaissance du client.
Le concept de cette solution peut être résumé comme suit :

- Lors de l'ouverture d'un compte bancaire pour un client, la banque emet une carte de paiement et la relie a son compte financier.
- La banque installera une applet d'identité comme un portefeuille séparée.
- Une authentification par second facteur peut être appliquée pour une sécurité optimale, par exemple, la banque peut envoyer un mot de passe à usage unique (OTP).
- Le client procédera à la première configuration de l'applet d'identité de la carte, y compris la définition du code PIN et l'insertion de la graine de clé maîtresse, de la même manière que pour le processus d'activation de la carte de paiement et l'installation de l'application logicielle appropriée.
- La banque installera l'applet d'identité sur la carte à puce de l'agence (il ne s'agit pas d'une carte de paiement, mais uniquement de l'applet d'identité).
- L'agence effectuera la première configuration de l'applet d'identité de la carte, y compris la définition du code PIN et l'insertion de la graine de clé maîtresse pour installer l'application appropriée.
- La banque émettra une VC (Verifiable Credentials) signée par sa clé privée pour le client afin de certifier ses informations de base et son compte bancaire.
- Le client pourra divulguer de manière sélective une partie de ses données et les signer avant de les présenter à l'agence comme preuve de la propriété du compte bancaire.
- L'agence sera en mesure de vérifier et d'authentifier le client.
- La banque émet un VC pour l'agence contenant des informations de base, un numéro de téléphone autorisé et un service de promotion.
- La banque stocke les métadonnées des justificatifs d'identité sur la blockchain.
- L'agence reçoit les informations d'identification et les stocke dans son portefeuille.
- L'agence appelle le client en utilisant le numéro de téléphone autorisé.
- Le client demande à l'agence d'effectuer des contrôles d'identité mutuels pour s'assurer que les deux parties ont une identité valide.
- L'agence génère un VC et utilise la carte pour le signer puis l'envoie au client.
- Le client vérifie le VC de l'agence par validation cryptographique. validation cryptographique.
- Le client génère une VC incluant des attributs sélectifs à partager avec l'agence et utilise la carte pour la signer puis la transmettre à l'agence.
- L'agence vérifie la VC du client par validation cryptographique. En cas d'échec du contrôle d'identité, le flux s'arrête et le client abandonne l'appel et supprime le DID par paire de l'agence - éventuellement pour déclencher le mécanisme de fraude bancaire et mettre à jour une liste noire.

- Si le contrôle d'identité est réussi, le flux humain se poursuit, l'agence propose le service qu'elle souhaite vendre.
- Si le client accepte l'offre, l'agence envoie le dossier à la banque.
- [Optionnellement], la banque appelle le client pour confirmer et demander des preuves supplémentaires.
- La banque transfère le service convenu, tel qu'un contrat immobilier, au client.

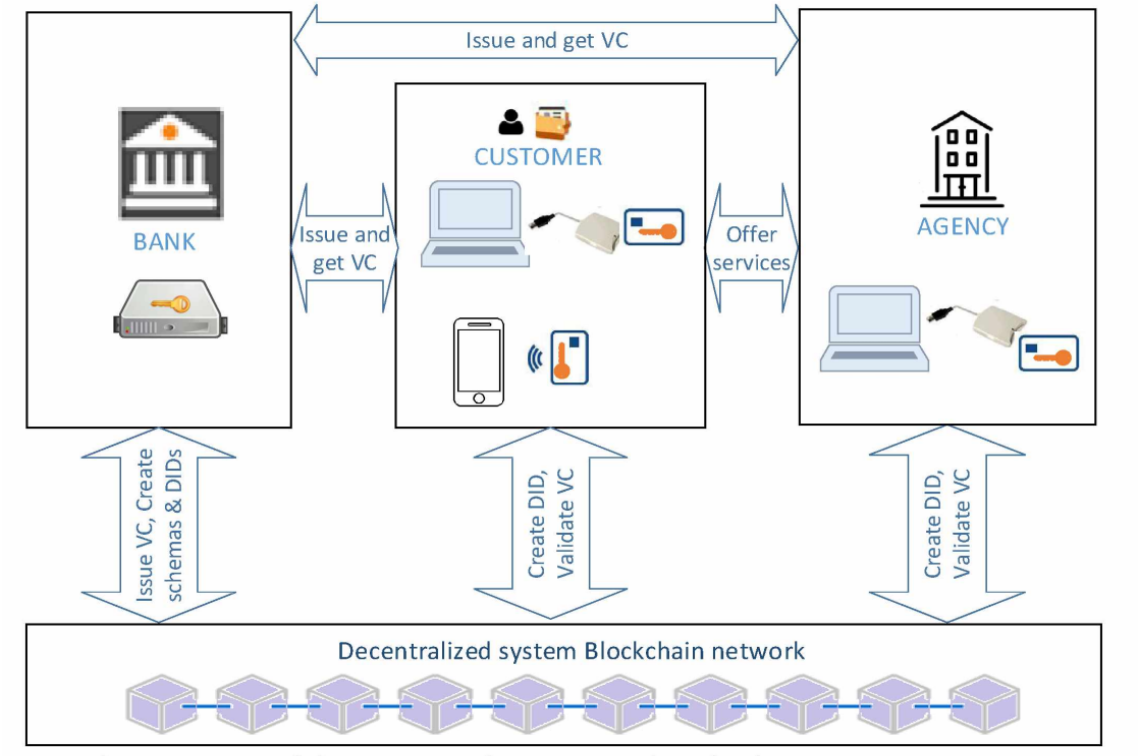

FIGURE 2.8 – Architecture Banking cards

- Un autre système est proposé dans [41], **Healthcare IdM**. C'est un système de gestion d'identité décentralisée pour les soins de santé à distance basé sur une blockchain consortium Ethereum. Le consortium est géré par les régulateurs de santé, ou chaque membre de ce consortium est un noeud . Les fournisseurs de soins de santé et les patients sont les utilisateurs du système, qui sont identifiés par un identifiant de santé unique (healthID), utilisé pour l'authentification et l'identification. Tout les acteurs du système utilisent une application pour s'inscrire sur la blockchain. Cette application contient un portefeuille sécurisé, qui permet de stocker des pairs de clés, publique et privée. La clé privée est utilisée pour signer les transactions, alors que la clé publique, est utilisée pour générer un compte sur la blockchain. Ce compte sera utilisé pour déployer des contrats intelligents sur cette dernière. Les régulateurs de santé, effectuent une vérification d'identité pour enregistrer le HealthID. Pour les fournisseurs de santé, la vérification de leurs identités se fait grâce à leur licence d'exercice, cependant que les utilisateurs peuvent prouver leurs identités à l'aide de document publique, telque, passeport, carte d'identité nationale ou des permis de conduires. La figure 2.9, montre le processus de ce système.

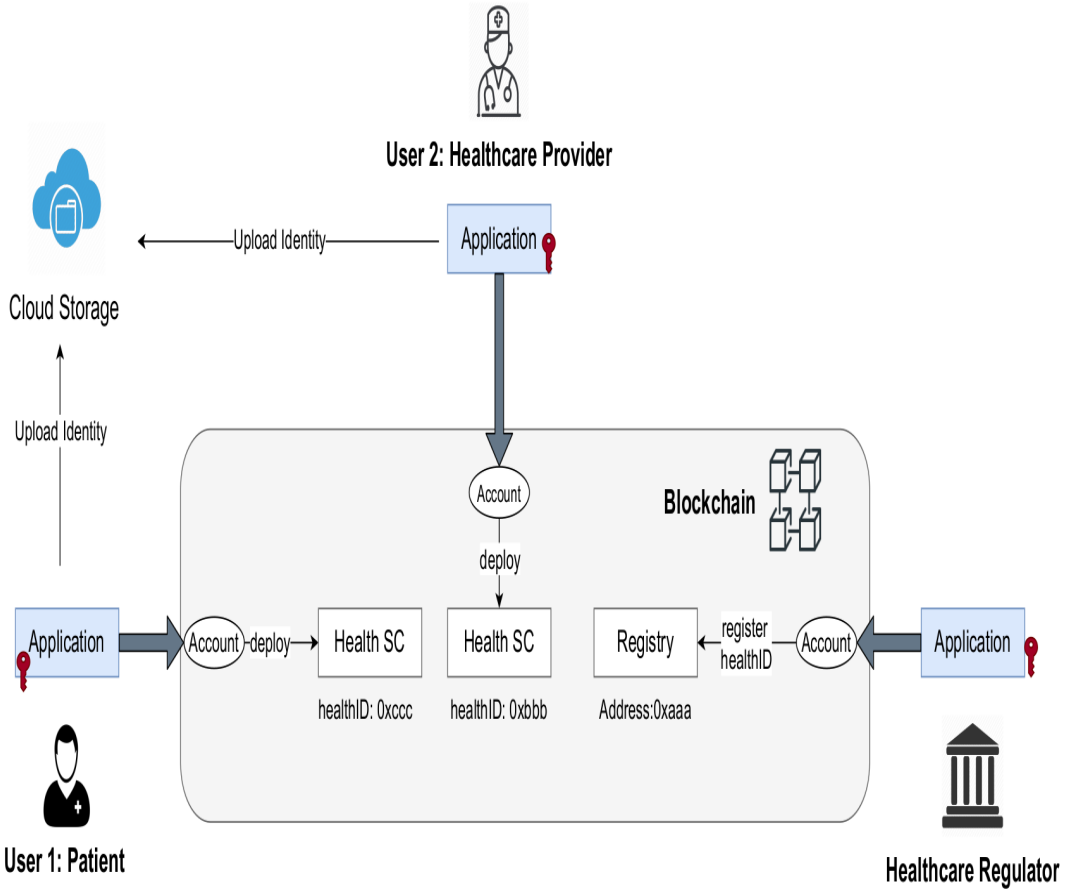

FIGURE 2.9 – Architecture Health IdM

2.5.2 Systèmes de gestion d'identité existant

On peut citer les systèmes suivants :

2.5.2.1 uPort

uPort est un système de gestion d'identité numérique open-source [32]. Il offre une identité auto-souveraine. Cette plateforme d'identité auto-souveraine est enregistrée sur une blockchain Ethereum publique sans autorisation. Accessible à l'aide d'une application mobile. Une identité uPort peut être créée pour des utilisateurs, organisations ou autres entités. uPort fournit des mécanismes pour récupérer la clé privée qui est stockée sur le mobile de l'utilisateur : soit en récupérant l'identité à l'aide d'un mot de base (dont la clé privée est dérivée), soit en utilisant un groupe de mandataire préalablement choisis [39].

Selon [32], uPort est composé de differents composants. Dans ce qui suit, nous allons expliquer brievement ceux qui sont dans la figure 2.10 :

- **Composants du contrats intelligents**
  - **Contrat de contrôleur(Controller Contract)**
    il s'agit de la logique de contrôle globale avec la fonctionnalité de contrôle de l'accès au contrat du mandataire. En outre, il permet à l'utilisateur de récupérer son identité s'il perd son téléphone portable et sa clé privée. Il tient à jour une liste de délégués de récupération (par exemple, des membres de la famille,

des amis ou des institutions sélectionnés) qui peuvent aider l'utilisateur à retrouver son identité uPort.

- **Contrat de proxy(Proxy Contract)**
  Il s'agit de l'identifiant permanent d'un utilisateur lié à sa clé privée. Il permet donc à l'utilisateur de remplacer sa clé privée sans affecter son identité permanente.

- **Contrat de registre (Registry contract)**
  il offre un lien cryptographique entre un identifiant uPort et ses attributs de données ou données de profil stockées hors blockchain (par exemple, InterPlanetary File System (IPFS)). IPFS est un protocole peer-to-peer pour le stockage et la récupération de données sur un système de fichiers distribué. Le contrat de proxy ne peut mettre à jour que le contrat de registre.

- **Composants du serveur**
  - **Chasqui**
    Le serveur de messages gère tous les aspects des communications avec n'importe quelle application décentralisée et application mobile.
  - **Sensui**
    Le serveur d'approvisionnement en gaz évite à un nouvel utilisateur d'Ethereum d'avoir à acheter de l'Ether et à payer des frais pour utiliser le réseau. Il paie les frais de gaz pour le nouvel nouvel utilisateur, ce qui lui permet de créer un nouveau compte uPort instantanément.
  - **Infura Ethereum RPC**
    Cette API Infura fournit une interface RPC standard pour permettre à uPort de communiquer avec le réseau Ethereum.
  - **Infura IPFS**
    Cette API Infura fournit une interface standard pour permettre à uPort de communiquer avec le réseau IPFS.

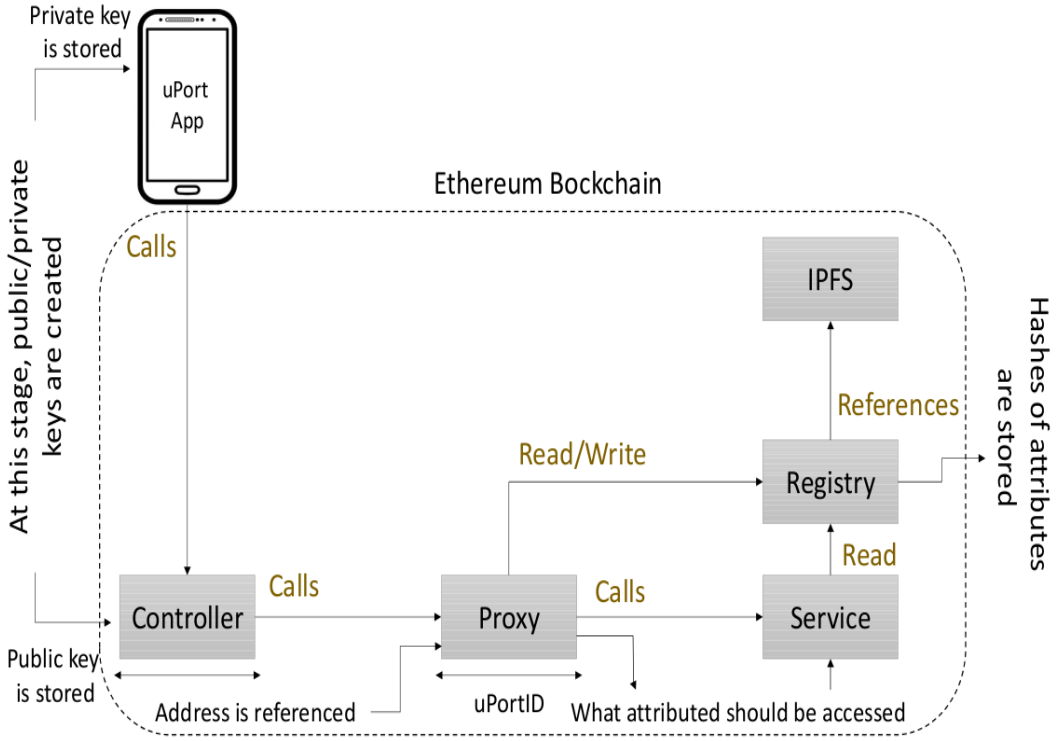

FIGURE 2.10 – Architecture uPort
[27]

2.5.2.2 Sovrin

Sovrin est un réseau open source [31], qui permet de fournir une identité auto-souveraine. Ce réseau est basé sur la blockchain publique avec autorisation Hyperledger Indy. [32] Sovrin est géré par des noeuds de confiances appelés steward. Ces institutions de confiances sont les seules qui peuvent exploiter les nœuds du réseau [33]. Dans ce qui suit, nous allons expliquer rapidement les composants mentionnés dans la figure 2.11 selon [32] .

- **Agents Sovrin (Sovrin Agents)**
  Sont des programmes demandés aux entités participantes, et qui permettent d'interagir entre eux. Il existe deux types d'agents : Edge Agent et Cloud agent.
- **Noeuds Sovrin (Sovrin Nodes)**
  Les noeuds, sont des serveurs qui exécutent des instances de code par le grand livre. Ces noeuds peuvent être des validateurs, ou des observateurs, mais ils ne peuvent agir que dans un sens à la fois. Il y a deux type de nœud : nœud observateur et nœud validateur.
- **Sovrin Ledgers**
  Il s'agit d'un grand livre distribué qui conserve les enregistrements de différents types de transactions. Sovrin possède deux types de ledgers : ledgers de configuration et ledgers des nœuds.

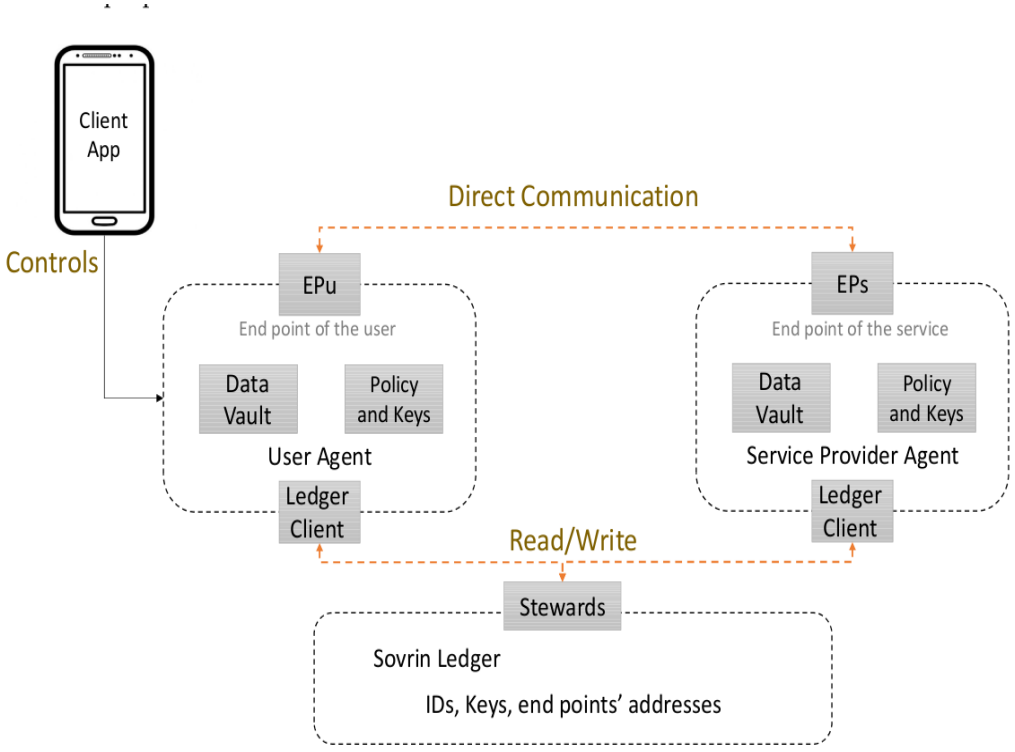

FIGURE 2.11 – Architecture Sovrin
[27]

2.5.2.3 LifeID

LifeID [36] est une plateforme open source, basé sur la blockchain, conçu pour offrir une identité numérique auto-souveraine. Ce réseau fonctionne avec les smartphones et les applications biométriques. LifeID permet aux utilisateurs de créer leurs identités d'une manière autonome, c'est a dire que les utilisateurs peuvent gérer facilement et en toute sécurité toutes les transactions en ligne et dans le monde réel qui nécessitent une authentification sans dépendre de la surveillance ou du contrôle par des tiers parties [36]. Elle permet également de stocker les données sur l'appareil de l'utilisateur et les informations nécessaires ne sont divulguées que lorsqu'une vérification d'identité est requise. De plus les identités LifeID peuvent être sauvegardées et récupérées via trois options différentes : sauvegardées dans un stockage froid, sauvegarde auprès de la famille ou des amis proches et auprès d'une organisation de confiance. Par conséquent, les utilisateurs peuvent lutter contre le vol en désactivant temporairement leurs identités et en les restaurant à l'aide des trois options citée au préalable. [36]

2.5.2.4 SelfKey

SelfKey [37], est une plateforme d'identité numérique auto-souveraine. SelfKey fonctionne sur la blockchain publique Ethereuem [37]. L'utilisateur de cette plateforme a tout le contrôle sur ses données qui sont stockées sur son appareil. Lorsqu'un tiers souhaite collecter des données spécifiques stockées sur la blockchain, les utilisateurs peuvent choisir de les afficher. Sur SelfKey, une identité peut être authentifiée en utilisant des algorithmes coercitifs et résistants à la censure. Ces algorithmes indépendants sont décentralisés. Les

revendications d'identité des utilisateurs ne peuvent être vérifiées que par des entités de confiance, ce qui garantit que les attributs d'attestation sont respectés. [38]

2.6 Conclusion

Les inconvéninets de la gestion d'identité numérique traditionnelle, ont donné naissance à la nouvelle gestion d'identité numérique auto-souvoraine. C'est une identité, qui permet aux utilisateurs de gérer leurs identités numérique, et avoir la liberté de controler leurs données personelles. Dans ce chapitre nous avons présenté la gestion d'identité numérique, ses modèles et nous avons détaillé les principes d'une identité auto-souveraine. Nous avons également cité les systèmes de gestion d'identité numérique auto-souveraine qui sont basés sur la blockchain. Dans le prochain chapitre, nous exposons notre proposition d'un système de gestion d'identité auto-sauveraine en utilisant la blockchain.

CHAPITRE 3

SYSTÈME DE GESTION D'IDENTITÉ NUMÉRIQUE EN UTILISANT BLOCKCHAIN

3.1 Introduction

Avec l'évolution continue des applications et des services offerts par Internet, la gestion d'identité numérique est devenue une nécessité incontournable. Bien que les modèles de gestion des identités numériques proposés aient évolué ces dernières années, ils ne répondent pas au besoin de décentralisation et d'autocontrôle des utilisateurs de leurs identités.
L'idée de modèle d'identité auto-souveraine est proposée dans la littérature pour répondre à ces exigences. Cependant, les propositions visant à réaliser de tels modèles en sont encore à leurs débuts. Notre travail dans ce chapitre s'article sur la proposition d'un modèle de gestion d'identité auto-souveraine. Pour se faire, nous avons adopté une architecture qui se compose de : l'utilisateur, un fournisseur d'identité, un fournisseur de service tout en utilisant la blockchain pour assurer la validation des identités numériques.
Dans ce chapitre, nous présentons le détail de notre proposition pour répondre à la problématique signalée, l'architecture de notre système, les outils utilisés, et nous terminons par des fenêtres d'illustration de notre application.

3.2 Motivation

Auparavant, la plus part des organisations utilisent l'identité centralisée, avec tous les défaillances connues d'un tel modèle. Les identités sont gardées et contrôlées par des points centraux, qui peuvent causer des défaillances des données, et qui retirent le contrôle des identités des mains des utilisateurs. De plus un utilisateur doit s'identifier et s'authentifier séparément pour chaque application, dont il est obligé de souvenir de tous ces mots de passes

Cependant et malgré l'évolution des modèles de gestion d'identité numériques, ils souffrent des deux problèmes majeurs suivants :

- **Contrôle d'identité :** La gestion des identités est réservée aux organismes chargés de fournir les identités, et l'utilisateur n'a aucun contrôle sur la gestion de son identité numérique, et celle-ci peut être retirée à tout moment. De plus, les processus d'identification et d'authentification ne sont pas séparés.
- **Centralisation :** La centralisation des systèmes est un inconvénient majeur de la défaillance des données. Cette centralisation existe toujours dans les modèles de gestion des identités encore en usage sur Internet

D'autre côté, les modèles auto-souveraines sont encore à leurs balbutiements. Les propositions actuelles répondent plus ou moins aux attentes de tels systèmes, et nécessitent beaucoup de travail. Dans la vie numérique, on ne trouve pas un système parfaitement fiable qui peut être considéré comme un modèle de gestion d'identité auto-souveraine de référence.

3.3 Idée et proposition

Un modèle de gestion d'identité auto-souveraine doit assurer plusieurs critères, tels que l'existence, le contrôle, l'accès, la persistance, la portabilité, etc. Une architecture qui utilise la blockchain est une architecture prometteuse pour assurer une grande partie de ces critères.

Notre idée est basée sur l'utilisation de blockchain et le système IPFS, plus les acteurs habituels qui sont l'utilisateur, le fournisseur d'identité et le fournisseur de service.

Le système des fichiers distribués IPFS est utilisé pour le stockage des données, tant que la validité de ces données est assurée par un sauvegarde des hashs des données sur la blockchain. Dans notre proposition, l'utilisateur peut choisir ses propres données à présenter pour s'identifier et gérer la permission d'accès à ses données.

- L'utilisateur stocke ses données sur le système de fichier distribué IPFS et il récupère leur hashs.
- Puis il envoi ces hashs à un fournisseur d'identité, dont son rôle ici est de vérifier et de prouver la validation des données fournies par l'utilisateur.
- Ce fournisseur d'identité utilise les hashs envoyé par l'utilisateur, et récupère les données de IPFS,
- Ensuite, il vérifie les données, et soit il les valide, soit il les rejette et ça selon les preuves existantes.
- Si les données sont prouvées, le fournisseur d'identité crée un code composé de hashs des données plus des données supplémentaires,
- Le fournisseur d'identité ensuite envoie le code à l'utilisateur, et en même temps il applique une fonction de hachage sur les hashs des données et les enregistre sur la blockchain en utilisant un smart contract.
- Quand un utilisateur veut accéder à un service via un fournisseur de service, il présente juste les données nécessaires pour l'accès à ce service, en donnant le hash de ces données plus le code envoyé de fournisseur d'identité.

- Le fournisseur de service va vérifier l'existence des hashs de ces données (les hashs des hashs des données et code fournis par le fournisseur d'identité) sur la blockchain,
- Si les données sont existent, il permet à l'utilisateur d'accéder à ses services.

3.4 Architecture du sysème proposé

3.4.1 Acteurs

Notre système est composé de trois acteurs : **Utilisateur (user), Fournisseur de service (FS), Fournisseur d'identité(FD) :**

- L'utilisateur, est la personne qui a besoin d'un service.
- Le fournisseur d'identité, qui est une autorité tel qu'une université, qui permet de certifier (valider) les données déclarées par l'utilisateur.
- Le fournisseur de service, est une organisation qui offre un service pareil aux utilisateurs.
- De plus, nous utilisons le IPFS pour sauvegarder les données des utilisateurs, et la blockchain pour sauvegarder le code fournis par le fournisseur.

3.4.2 Scénario d'utilisation

Notre solution est basée sur celle exposée dans [54]. Elle peut être utilisée avec n'importe quelle application ou service nécessite des documents des utilisateurs à vérifier. Cependant, elle est testée dans le cadre de l'accès aux documents universitaires afin de pouvoir accéder à un service comme postuler à un emploi ou se réinscrire dans une autre université. Cette proposition permet à l'utilisateur de conserver le contrôle et la permission d'accéder et de sélectionner la donnée ciblée par un service et pas tout l'ensemble des données. Ainsi elle assure l'existence permanent des données sur un système distribué IPFS, et garantit aussi la validation des données en utilisant la blockchain.

Notre application passe par trois phases, une phase d'enregistrement, une phase de validation et une phase de vérification. La figure 3.1 montre l'architecture de notre proposition.

Il est à noter que l'utilisateur est un étudiant universitaire ou X-étudiantt qui s'est préalablement inscrit dans son système universitaire. Le fournisseur d'identité est l'université, et le fornisseur de service est tout service compatible avec notre application tels que une autre université ou entreprise qui offre des postes d'emploi.

- **Enregistrement**
  - En premier lieu l'utilisateur accède à une interface, où il insère ses documents (diplôme, carte d'identité et relevé de note) sur le système des fichiers distribués (IPFS)
  - Le IPFS, renvoie trois hashs h1 (le hash du diplôme), h2 (le hash de la carte) ,h3 (le hash du relevé de note)
  - Ensuite, l'utilisateur envoie les trois hashs, h1, h2, h3 à l'université (fournisseur d'identité) pour la validation des documents.

- **Validation**
  - Le fournisseur d'identité (FD) qui est présenté dans notre scénario par l'université et via son propre interface vérifie les documents et les valides.
  - Ensuite, il crée un code, **Code=H(h1.h2.h3.date)**, tel que **H** est une fonction de hachage
  - Une fois le hash créé, le FD, insère dans la blockchain les informations suivantes : **H(h1.code), H(h2.code), H(h3.code), en utilisant un smart contract.**
- **Vérification**
  - Lorsqu'un utilisateur veut accéder à un service, il choisit la donnée qui veut présenter.
  - S'il choisit la première donnée, il envoie **h1.code** au FS.
  - Le FS, calcul le **H(h1.code)** et cherche son existance dans la blockchain en utilisant un smart contract.
  - Si le hash existe dans la blockchain, il prend h1 et cherche la donnée correspondate dans le IPFS, pour la récupérer.
  - A la fin, le fournisseur de service permet ou refuse l'accès de l'utilisateur à son service selon la vérification faite.

La figure 3.1 résume les différentes étapes.

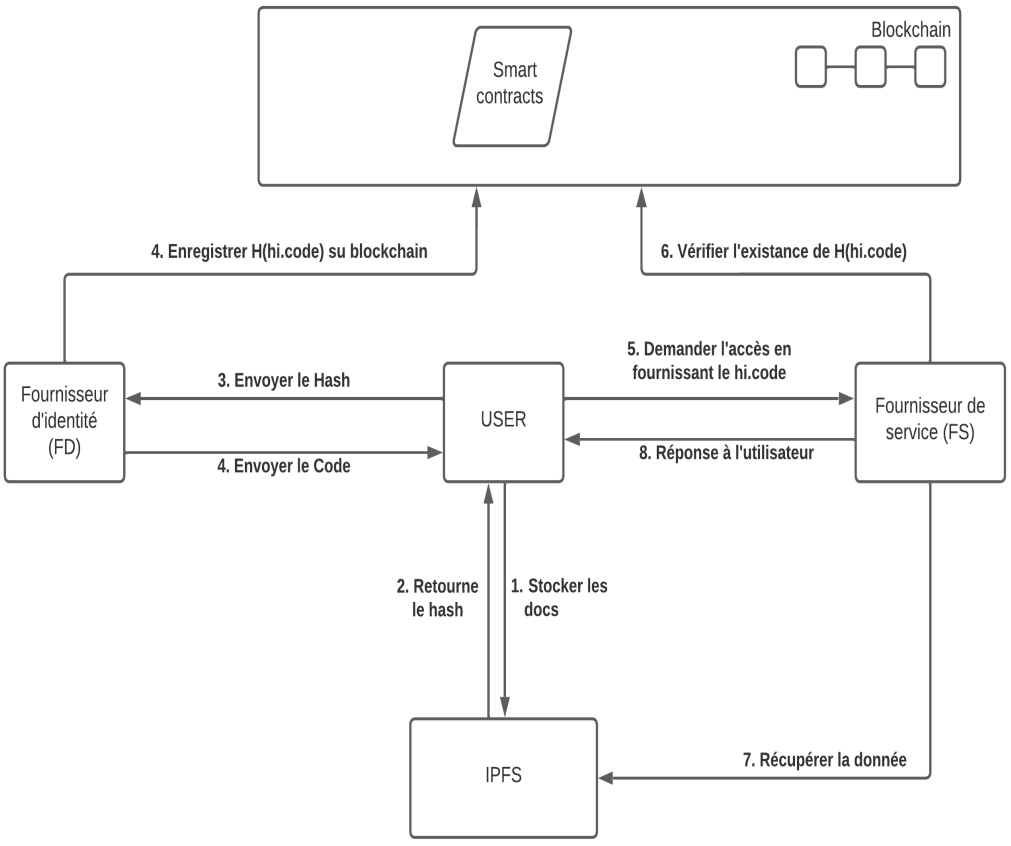

FIGURE 3.1 – Architecture de notre proposition

Remarque :
Pour une meilleure sécurisation des données, nous avons ajouté des hashs des données à la blockchain (fonction de hachage) au lieu de mettre des données en claire.

3.5 Implémentation et outils de développement

Pour le développement de notre application, nous avons utilisé les outils présentés dans ce qui suit.

3.5.1 Outils de développement

Dans cette application nous avons utilisé les langages de programmation et les outils qui sont présentés dans les lignes suivantes :

- **HTML5**
  HTML5, est une version du HTML (format de données conçu pour représenter les pages web). Cette version a été finalisée le 28 octobre 2014

- **CSS3**
  CSS3, signifie Feuilles de style en cascade, utilisé pour augmenter la fonctionnalité et la polyvalence, et une performance efficace du contenu du site. Il permet la création des sites Web riches en contenu qui ne nécessitent pas beaucoup de poids ou de codes, cela se traduit par des graphiques et des animations plus interactifs, une interface utilisateur supérieure, une organisation beaucoup plus importante et un temps de téléchargement plus rapide.

- **JavaScript**
  JavaScipt (abrégé en JS), est un langage de programmation léger, interprété, orienté objet et doté de fonctions de premier ordre. JavaScript n'est pas Java. Il est surtout connu comme langage de script pour les pages Web, mais il est également utilisé dans de nombreux environnements autres que les navigateurs. Il s'agit d'un langage de script multiparadigme basé sur des prototypes, dynamique et prenant en charge les styles de programmation orientés objet, impératifs et fonctionnels. JavaScript s'exécute sur le côté client du Web, ce qui permet de concevoir et de programmer le comportement des pages Web lorsqu'un événement se produit. JavaScript est un langage de script facile à apprendre et puissant, largement utilisé pour contrôler le comportement des pages Web. [45]

- **Solidity**
  Solidity est un langage de programmation orienté objet, à typage statique et à accolades conçu pour développer et mis en œuvre de contrats intelligents sur diverse plateformes. Les programmes dans Solidity s'executent sur la machine virtuelle Ethereum. Il a été développé par Christian Reitwiessner, Alex Beregszaszi et plusieurs anciens contributeurs du noyau d'Ethereum. [46]

- **Truffle Suite**
  Trufflesuite, est un framework pour le développement des applications décétralisées (dApp). Il se compose essentiellement de trois parties différentes, ces trois parties sont Truffle, Ganache et Drizzle. Son objectif est de garantir un processus de développement plus accessible. Truffle offrent de différents fonctionnalité comme : la gestion des contrats intelligent, des tests automatisés des contrat intelligent, migration et déploiement scriptables, gestion du réseau, console interactive. [48]

- **Ganache**
  Ganache est une blockchain personnelle pour le développement rapide d'applications distribuées Ethereum et Corda. Elle permet de développer, déployer et tester des applications distribuées dans un environnement sûr et déterministe en fournissant 10 compte Ethereum avec une balance de 100 ETH (ce sont des faux ether) . Ganache existe en deux versions : interface utilisateur et une interface CLI [47]. La figure 3.2 montre la version interface de la blockchain ganache.

- **Metamask**
  Metamask, est un portefeuille de crypto-monnaies, développé par ConsenSys Software Inc, une société de logiciels de blockchain. Metamask permet aux utilisateurs d'accéder à leur portefeuille Ethereum pour interagir avec la blockchain Ethereum. L'accès à ce logiciel se fait via une extension de navigateur ou d'une application mobile. [49]

- **IPFS**
  IPFS (InterPlanetary File System), est un protocole et un réseau pair à pair avec un système de fichier conçu pour stocker et partager des données. Il a été créer par Juan Benet, le fondateur de Protocol Labs. IPFS vise rendre le web plus rapide, plus sûr et plus ouvert. IPFS fournit un hachage cryptographique, qui permet d'identifier le contenu des données au lieu d'utiliser des adresses machines sur le réseau. [50]

- **Remix**
  Remix IDE, est un un envirennement de développement open source, qui permet de développer, déployer et administrer des contrats intelligents pour les blockchains de type Ethereum. [53]

- **Web3.js**
  Web3.js, est une collection de bibliothèques, qui permet d'intéragir avec la blockchain Ethereum et d'autre fonctionnalitées comme, l'envoie des Ether d'un compte vers un autre, de créer des SC et de lire et écrire des données a partir de ces dernières. [51]

- **Infura IPFS**
  Infura IPFS, est un API qui permet à notre application de communiquer avec le réseau IPFS. [57]

- **Visual Studio Code**
  Visual Studio Code (Version 1.66.2), qui un éditeur de code développée par Microsoft. [52]

3.5.2 Installation des outils et configuration d'environnement

3.5.2.1 Installation des outils

Dans cette partie, nous allons expliquer comment installer les outils mentionnés auparavant.

- Sous Linux et avant de commancer l'installation de truflle suite, ganache et web3.js, il faut tout d'abord installer NodeJs via la commande suivante :
  - **sudo apt-get install nodejs.**
- Pour installer ganache, nous utilisons les commandes suivantes :
  - Pour la version CLI : **npm install -g ganache-cli**
  - Pour la version interface : **wget le lien de téléchargement de ganche interface.**
- L'installation de Truffle Suite se fait par la commande :
  - **npm install -g truffle**
- Pour installer web3, la commande suivante est utilisée :
  - **npm install -g web3.**
- Pour installer ipfs, il suffit de taper la commande suivante :
  - **npm install -g ipfs-http-client**

3.5.2.2 Configuration d'environnement

Nous allons d'abord créer un répértoire qui va inclure les fichiers de notre projet de cette façon :
mkdir projetFinEtude
cd projetFinEtude

- Pour initialiser **NodeJs**, nous utilisons la commande suivante :
  npm init
- Pour initialiser truffle, la commande suivante est utilisé
  truffle init
- Pour initialiser **Ganche**, soit on lance la fenêtre ganche celle que montre la figure 3.2, soit on lance la commande **ganache-cli**.

FIGURE 3.2 – Blockchain personelle ganache

- Pour initialiser **MetaMask**, on utilise l'extension qui se trouve dans le navigateur, et on importe un compte en utilisant une clé privée d'un faut compte de la blockchain ganache. Les figures 3.4 et 3.3

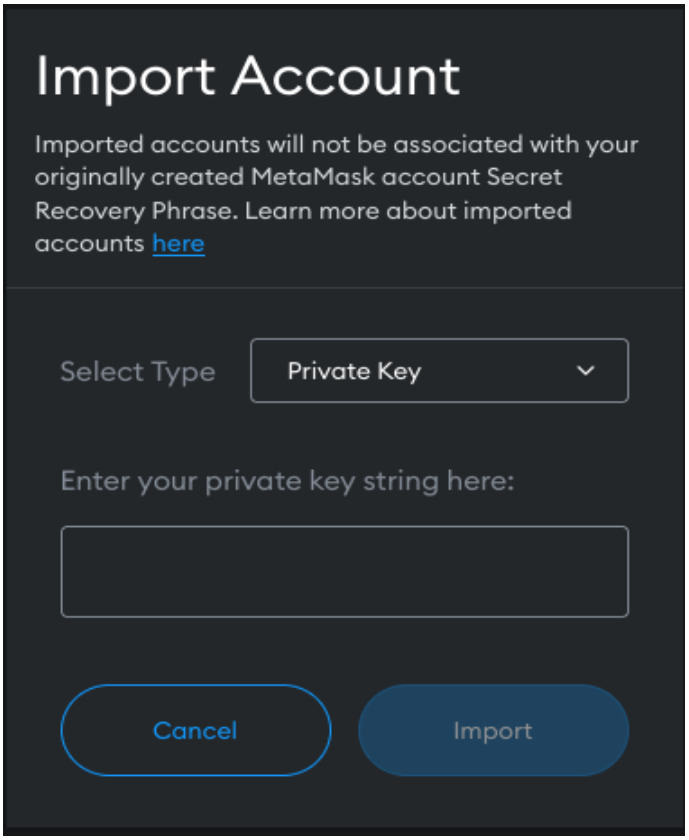

FIGURE 3.3 – Importer le compte

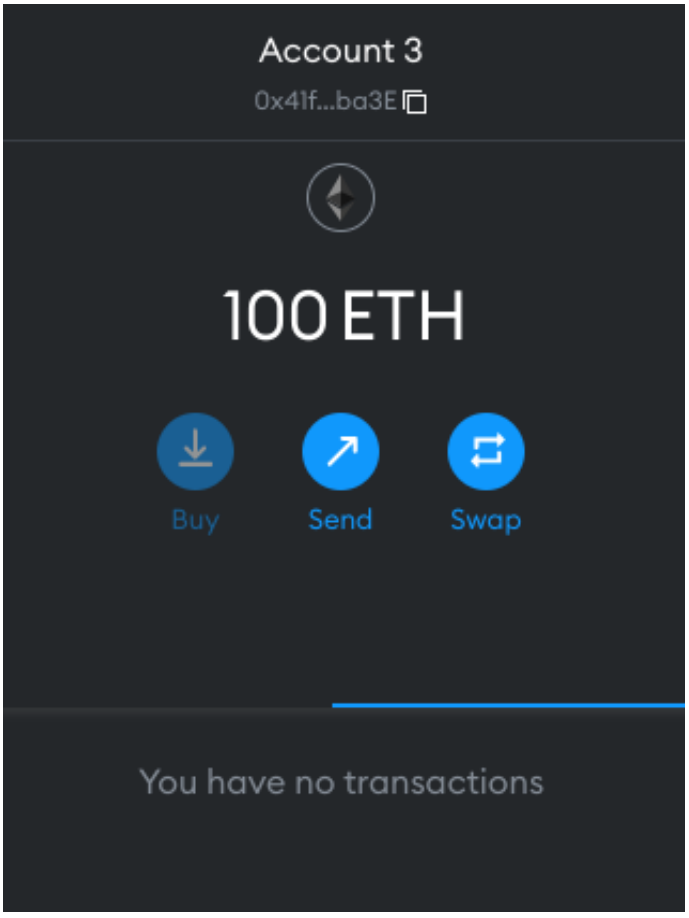

FIGURE 3.4 – Compte MetaMask

3.6 Quelque fenêtre de notre application

Dans ce qui suit nous allons illustrer quelques interfaces de notre application :

3.6.1 Fenêtres de l'utilisateur

Au début, l'utilisateur fournit les données suivantes :

- Sa carte d'identité scannée.
- Son dilpôme scanné.
- Son relevé de note version numérique ou scanné.

Ces documents sont sauvegardés sur le système de fichier distribué IPFS (voir la figure 3.5).

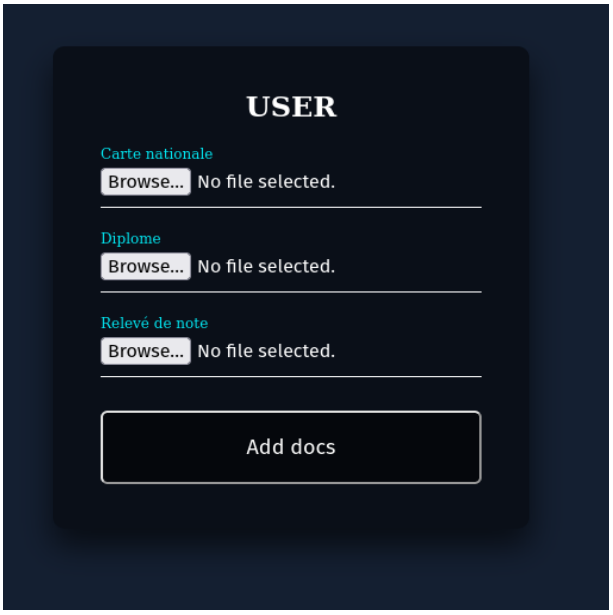

FIGURE 3.5 – Stockage des documents

En cliquant sur le button **Add docs**, le IPFS retourne un hash pour chaque documents ajouté. Ce hash est envoyé à l'université, et l'utilisateur sera dirigé vers une autre fenêtre comme montre la figure 3.6

Cette fenêtre contient les éléments suivants :

- Un button **Doc's link to IPFS**, qui permet d'accèder aux documents stockés sur le réseau IPFS.
- Un button **Get code**, qui permet de récupérer le code généré par l'université.
- Un button **Send Data**, qui permet d'envoyer la donné séléctionnée dans le champ au fournisseur de service.

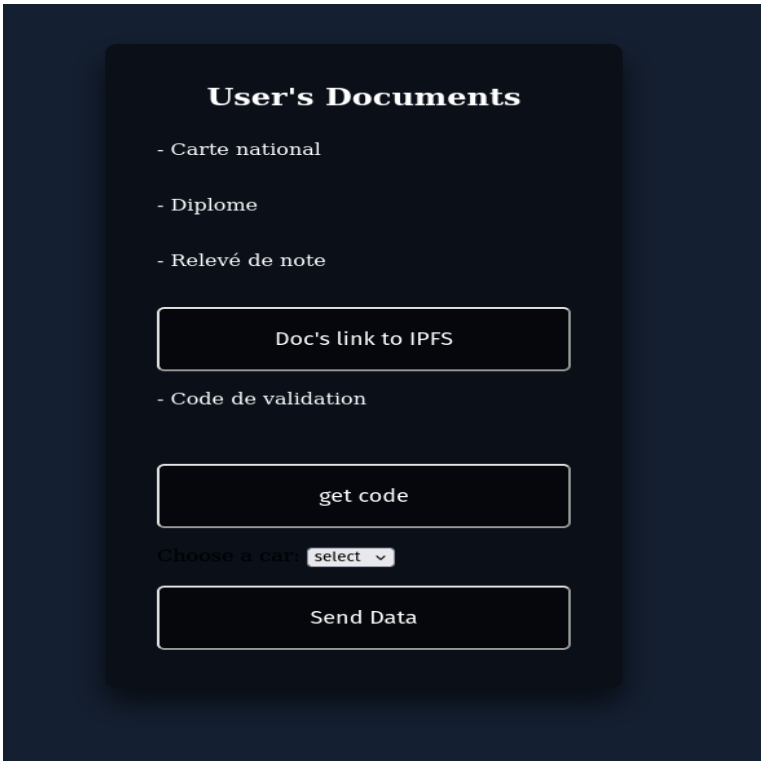

FIGURE 3.6 – Affichage des données

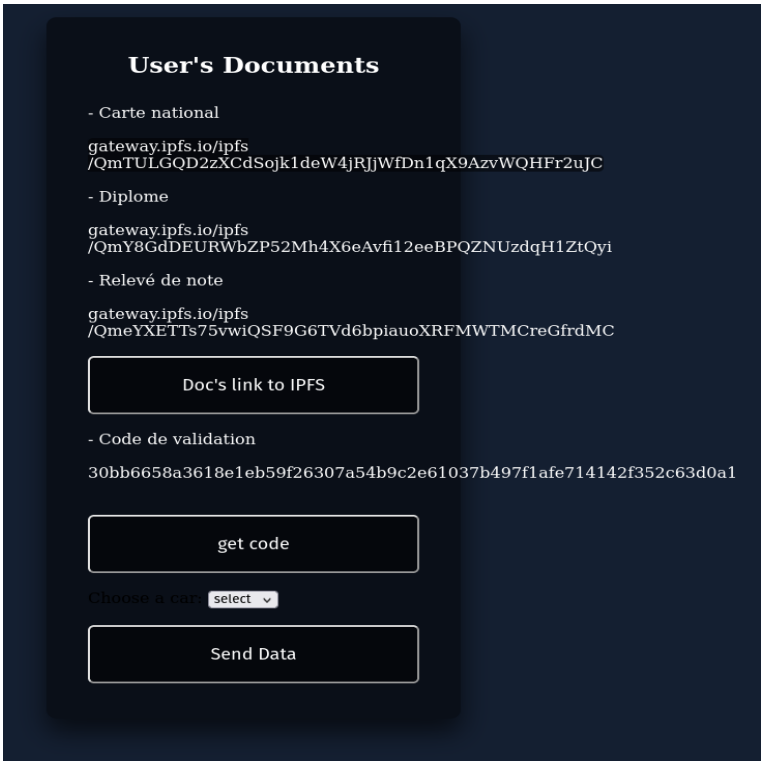

FIGURE 3.7 – Interface principle de l'utilisateur

3.6.2 Fenêtre de l'université

- Sur cette fenêtre 3.8, un administrateur de l'université accéde aux documents stockés

sur IPFS, et cela en cliquant sur le button **Show Docs** pour vérifier les documents de l'utilisateur.

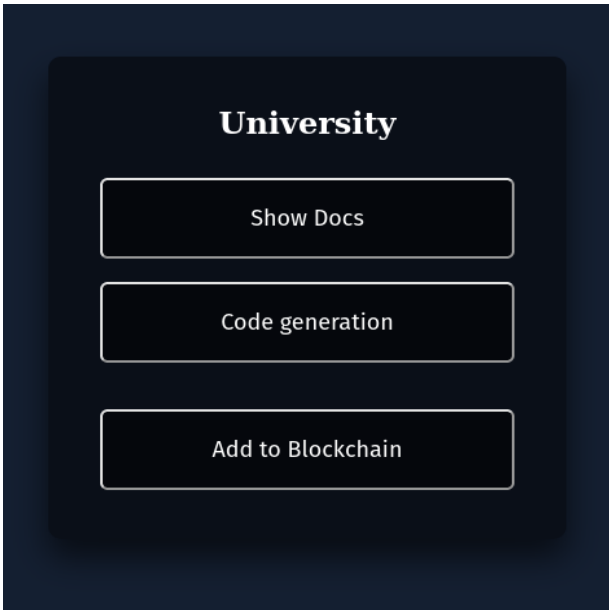

FIGURE 3.8 – Interface de l'université

- Une fois les documents vérifiés, le système génere un code qui permet à l'utilisateur de s'identifier dans l'organisation.
- Après avoir envoyer le code à l'utilisateur, il applique une fonction de hacahage sur le hash du document concaténé avec le code généré, et se connecte à la blockchain via **MetaMask** (voir la figure 3.9), afin de pouvoir ajouter les hashs à la blockchain.
- Chaque hash est sous forme d'une transaction. La figure 3.11 montre toutes les transactions qui sont enregistrées dans la blockchain.

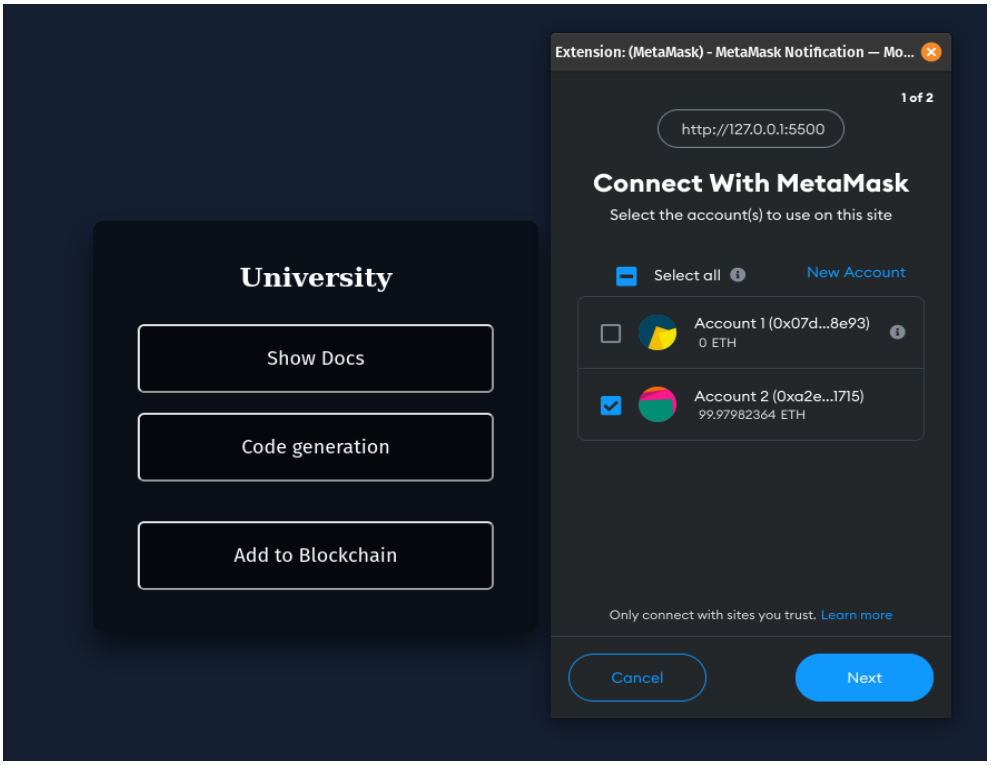

FIGURE 3.9 – Connection à la blockchain via MetaMask

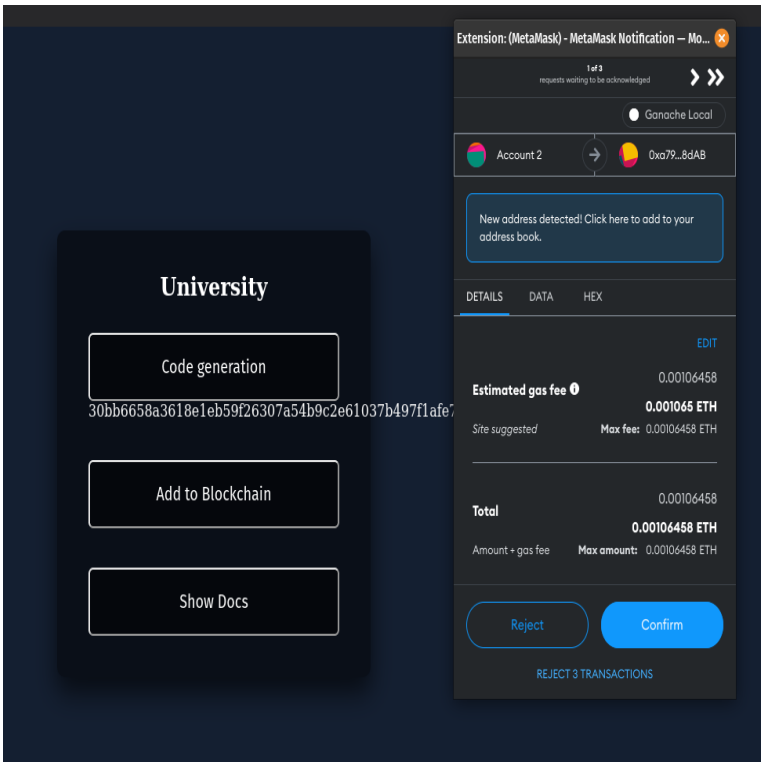

FIGURE 3.10 – Confirmation de transaction

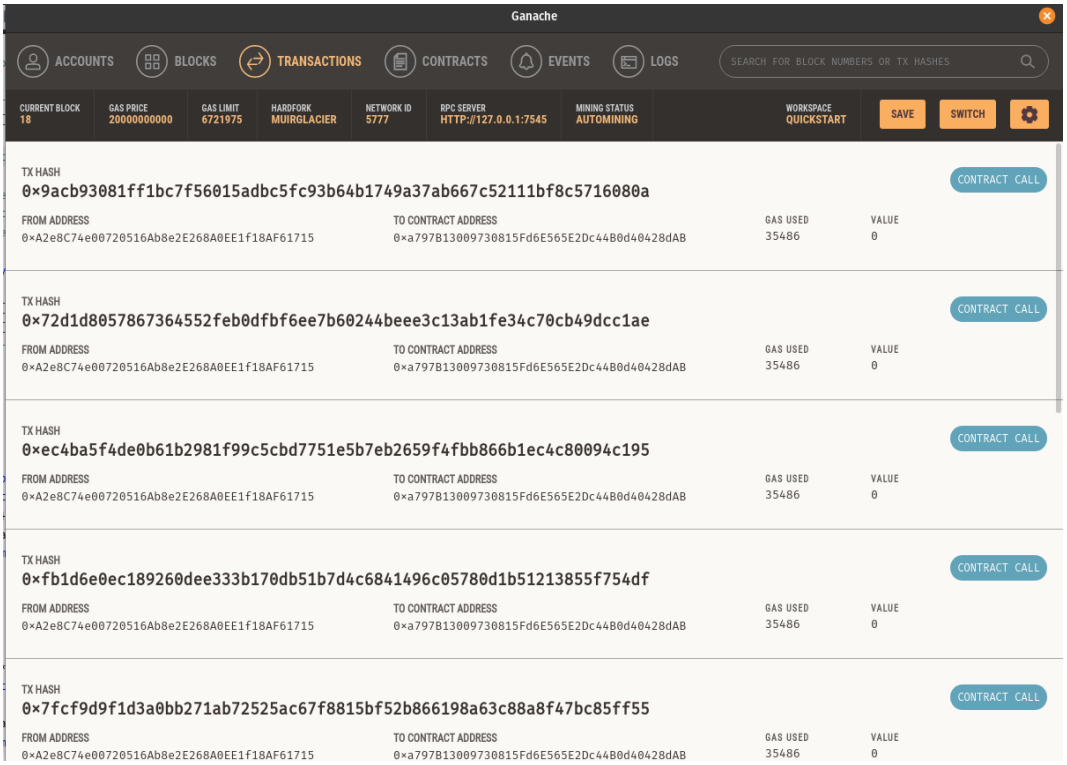

FIGURE 3.11 – Transactions dans la blockchain Ganache

3.6.3 Fenêtre de fournisseur de service

Dans cette fenêtre, le button **Verification**, permet de vérifier l'existance du hashs des hashs des documents dans la blockchain et le compare avec celui envoyé par l'utilisateur. Si le hash existe dans la blockchain, il se dirige vers le réseau IPFS et récupère la donnée demandée pour offrir le service.

FIGURE 3.12 – Interface de fournisseur de service

3.7 Analyse et discution

Dans ce travail, nous avons développé un système de gestion d'identité auto-souveraine, en utilisant la blockchain et IPFS. Une telle solution permet d'assurer plusieurs critères d'un modèle auto-souveraine, qui sont analysées dans les lignes qui suivent.

Propriétés de système proposé

Nous discutons ici si notre système assure les dix critères proposés par Christopher Allen [43].

- **Existence :** l'identité d'un utilisateur est définie en appliquant une fonction de hachage, sur ses données, donc il n'aura pas de collusion d'identité.
- **Contrôle :** seul l'utilisateur peut contrôler ses données sur IPFS. Donc, il est le seul qui peut contrôler chaque attribut de son identité.
- **Accès :** Les données sont sauvegardées sur IPFS et ses codes sur la blockchain, ce qui permet à l'utilisateur d'accéder et récupérer ses données facilement.
- **Transparence :** la transparence des algorithmes et systèmes est assurée, car les systèmes utilisés sont connus et accessibles, et les algorithmes proposés peuvent être accessibles aussi.
- **Persistance :** les attributs d'identité sont gardés au niveau de compte d'utilisateur (hash et code), et leurs hashs sont toujours existent sur la blockchain, ainsi que les documents utilisés peuvent être toujours sur IPFS. Donc, la persistance est assurée.
- **Portabilité :** L'identité est présentée par un hash, qui peut être transportée, seulement il sera mieux de développer une application mobile pour une meilleure portabilité.
- **Interopérabilité :** L'identité proposée peut être utilisée avec des applications compatibles avec notre système.
- **Consentement :** les données sont gardées sur IPFS, qui est un système distribué et accessible par tout le monde, et ce qui a le CID (l'identifiant), peut les accéder.
- **Minimalisation :** l'utilisateur peut choisir qu'elle donnée présente à un service en utilisant seulement son hash et son code, ce qui permet la minimisation des données exposées.
- **Protection :** Dans notre système, la vérification et la validation d'identité par blockchain est très intéressante, car elle permet de stocker et de gérer en toute sécurité les identifiants numériques de l'utilisateur (identifiants non modifiables et conservés). Cependant, à ce jour, il n'existe pas de méthode standardisée pour stocker et coder des données cryptées sur IPFS. Ce dernier utilise le chiffrement de transport mais pas le chiffrement de contenu. Cela signifie que les données sont sécurisées lorsqu'elles sont envoyées d'un nœud IPFS à un autre. Cependant, n'importe qui peut télécharger et afficher ces données s'il possède le CID.

Cette analyse est résumée dans la table 3.1

Proriété d'identité auto-souveraine	**Vérification**
Existence	Oui
Contrôle	Oui
Accès	Oui
Transparence	Oui, peut être totalement assurée
Persistance	Oui
Portabilité	Partielle. Elle peut être améliorer
Interopérabilité	Oui
Consentement	Oui
Minimalisation	Oui
Protection	Oui pour l'identifiant sur la blockchain Non pour les données sur IPFS

TABLE 3.1 – Résumé des propriétés de notre système

De plus, et puisque les documents de l'utilisateur sont gardés sur l'IPFS et les codes sur blockchain, qui sont des systèmes décentralisés, notre proposition assure d'autres propriétés de plus tels que :

- **Indépendance d'identité des administrateurs ou des fournisseurs d'identité :** le code créé à partir d'identité est inséré dans la blockchain, qui est un système distribué sans organe central. De plus et comme il est connu, les transactions exécutées dans l'environnement de la blockccahin et les données stockées dans son espace de stockage sont non modifiables et permanents.
- **Disponibilité :** les données étant présentées sur plusieurs nœuds (IPFS et blockchain), alors leur disponibilité est assurée. De plus les données sont dupliquées sur plusieurs nœuds indépendants, ce qui rend la censure plus compliquée.
- **Performance :** l'accès aux données est plus efficace, car les ressources sont accessibles en pair-à-pair.

3.8 Conclusion

La vérification d'identité par blockchain, basée sur la technologie du registre distribué, est très prometteuse pour le marché de la gestion des identités, car elle permet de stocker et de gérer en toute sécurité les identifiants numériques des entreprises et des personnes. En plus, la technologie pourrait permettre aux individus de garder le contrôle de leurs identités numériques et d'assurer une identité auto-souveraine.

Dans ce chapitre, nous avons proposé un système de gestion d'identité auto-souveraine, en utilisant la blockchain pour stocker et sécuriser les identifiants, et IPFS pour stocker les documents des utilisateurs. Une telle solution permet de vérifier un ensemble de propriétés de modèle auto-souveraine tels que l'existence, le contrôle, l'accès, la persistance, etc. Pour pouvoir implémenter notre solution, nous avons utilisé plusieurs technologies et plates formes tels que Ganache, Remix et autres. Ainsi, nous avons utilisé un smart contact afin de pouvoir insérer les identifiants dans la blockchain.

Néanmoins, notre application nécessite plus de travail, afin de pouvoir l'améliorer surtout l'aspect de protection des données sur IPFS qui est un nouveau système encore au stade de développement. D'autre côté, la portabilité d'identité peut être aussi améliorée

CONCLUSION GÉNÉRALE

La gestion des identités est un processus complexe engendrant l'identification, l'authentification et l'autorisation des utilisateurs à accéder aux différents services et applications. Dans la littérature, on peut trouver des solutions d'identité numérique qui sont mises en œuvre pour réduire le risque de fraude, d'usurpation d'identité et de violation des données. Des approches décentralisées de la gestion des identités à base de blockchain sont adoptés par certaine entreprises, surtout pour la nature inviolable de la technologie.

L'utilisation de blockchain pour la gestion d'identité peut assurer un modèle d'identité auto-souveraine. Cette dernière est une identité numérique contrôlée et possédée par l'utilisateur. Elle préserve le droit des individus de contrôler leur identité.

Dans ce mémoire, nous avons abordé la technologie de la blockchain et contexte de la gestion d'identité numérique. Nous avons donné une présentation détaillée sur la blockchain, ses composants et fonctionnement en exposant les différentes plateformes existent et quelques cas d'application. Pour l'identité numérique et après avoir expliqué son principe, nous avons parlé de différents modèles et présenté un aperçu des nouveaux travaux sur la gestion de l'identité basée sur la blockchain dans la littérature.

Notre proposition dans ce modeste travail est s'articulé autour d'un système de gestion d'identité numérique auto-souveraine. Nous avons utilisé la blockchain pour assurer la validation d'identité de l'utilisateur et assurer sa sécurisation fournie par la nature de fonctionnement de la blockchain. Les documents d'utilisateur sont stockés sur IPFS, qui est un système de fichiers distribué adressable par le contenu, et qui assure la disponibilité de ces documents. L'utilisateur stocke ses données sur le système de fichier distribué IPFS et il récupère leur hashs, et les envois au fournisseur d'identité, dont son rôle ici est de vérifier et prouver la validation des données fournies par l'utilisateur. Ce fournisseur utilise les hashs envoyé par l'utilisateur, pour les récupère de l'IPFS et les vérifier. Après la vérification il envoie un code calculé de cette identité à l'utilisateur, et en même temps il applique une fonction de hachage sur les hashs des données et les enregistre sur la blockchain. Quand l'utilisateur se présente à un service, il présente la donnée nécessaire seulement. Le fournisseur de service va vérifier l'existence des hashs de ces identifiants

(les hashs des identifiants et code fournis par le fournisseur d'identité) sur la blockchain, Après une vérification positive, il permet à l'utilisateur d'accéder à ses services.

Comme un cas d'utilisation de notre proposition, nous avons pris l'université comme fournisseur d'identité, et une organisation compatible (autre université, ou organisation d'offre d'emplois).
Notre proposition assure presque la totalité des propriétés d'un modèle d'identités auto-souverain telles que le contrôle, l'accès, la persistance, etc. elle assure aussi la disponibilité et l'indépendance d'identité des administrateurs ou des fournisseurs d'identité. Mais, grâce à la nature de fonctionnement d'IPFS, les données ne sont pas cryptées et peuvent être visibles par ceux qui ont le CID (identifiant) des données.

Comme perspectifs et pour améliorer notre application, nous proposons :

- Appliquer un cryptage sur les données avant de les sauvegarder sur l'IPFS pour ne pas les laisser en claire pour ceux qui ont leur CID.
- Développer une application mobile pour améliorer la portabilité d'identité numérique.

BIBLIOGRAPHIE

[1] Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. Blockchain technology overview. *arXiv preprint arXiv :1906.11078*, 2019.

[2] Abdurrashid Ibrahim Sanka, Muhammad Irfan, Ian Huang, and Ray CC Cheung. A survey of breakthrough in blockchain technology : Adoptions, applications, challenges and future research. *Computer Communications*, 2021.

[3] Ittay Eyal and Emin Gün Sirer. Majority is not enough : Bitcoin mining is vulnerable. In *International conference on financial cryptography and data security*. Springer, 2014.

[4] Julija Golosova and Andrejs Romanovs. The advantages and disadvantages of the blockchain technology. In *2018 IEEE 6th workshop on advances in information, electronic and electrical engineering (AIEEE)*. IEEE, 2018.

[5] Bikramaditya Singhal, Gautam Dhameja, and Priyansu Sekhar Panda. *Beginning Blockchain : A Beginner's guide to building Blockchain solutions*. Springer, 2018.

[6] JAF Schlumberger, P Geoffin, S Voison, and P Campsavoir. Livre blanc «blockchains & développement durable». *Vie & Sciences de l'Entreprise*, (211/212), 2021.

[7] Janvi Dattani and Harsh Sheth. Overview of blockchain technology. *Asian Journal of Convergence in Technology*, 5(1), 2019.

[8] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology : Architecture, consensus, and future trends. In *2017 IEEE international congress on big data (BigData congress)*. IEEE, 2017.

[9] Éric A Caprioli. La blockchain ou la confiance dans une technologie. *la semaine juridique, édition générale*, (23), 2016.

[10] S Velliangiri and P Karthikeyan. Blockchain technology : Challenges and security issues in consensus algorithm. In *2020 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 2020.

[11] Tatiana Gayvoronskaya and Christoph Meinel. *Blockchain : Hype Or Innovation*. Springer Nature, 2020.

[12] Gavin Zheng, Longxiang Gao, Liqun Huang, and Jian Guan. *Ethereum Smart Contract Development in Solidity*. Springer, 2021.

[13] Olivier Desplebin, Gulliver Lux, and Nicolas Petit. Comprendre la blockchain : quels impacts pour la comptabilité et ses métiers ? *ACCRA*, (2), 2019.

[14] Shilpa Karkeraa. Aspects of blockchain transactions. In *Unlocking Blockchain on Azure*. Springer, 2020.

[15] CHELAGHMA Abdessamad. Primitives cryptographiques dans la blockchain. 2021.

[16] Satoshi Nakamoto. Bitcoin whitepaper. *https://bitcoin.org/bitcoin.pdf-(: 17.07.2019)*, 2008. Consultée le 8 mars 2022.

[17] Imran Bashir. *Mastering blockchain*. Packt Publishing Ltd, 2017.

[18] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Weili Chen, Xiangping Chen, Jian Weng, and Muhammad Imran. An overview on smart contracts : Challenges, advances and platforms. *Future Generation Computer Systems*, 105, 2020.

[19] Tharaka Hewa, Mika Ylianttila, and Madhusanka Liyanage. Survey on blockchain based smart contracts : Applications, opportunities and challenges. *Journal of Network and Computer Applications*, 177, 2021.

[20] Nick Szabo. Smart contract. `https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html`. Consultée 28 février 2022.

[21] Ruhi Taş and Ömer Özgür Tanrıöver. Building a decentralized application on the ethereum blockchain. In *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. IEEE, 2019.

[22] Weiqin Zou, David Lo, Pavneet Singh Kochhar, Xuan-Bach Dinh Le, Xin Xia, Yang Feng, Zhenyu Chen, and Baowen Xu. Smart contract development : Challenges and opportunities. *IEEE Transactions on Software Engineering*, 47(10), 2019.

[23] Yang Liu, Debiao He, Mohammad S Obaidat, Neeraj Kumar, Muhammad Khurram Khan, and Kim-Kwang Raymond Choo. Blockchain-based identity management systems : A review. *Journal of network and computer applications*, 166, 2020.

[24] Shu Yun Lim, Pascal Tankam Fotsing, Abdullah Almasri, Omar Musa, Miss Laiha Mat Kiah, Tan Fong Ang, and Reza Ismail. Blockchain technology the identity management and authentication service disruptor : a survey. *International Journal on Advanced Science, Engineering and Information Technology*, 8(4-2), 2018.

[25] Sesaria Kikitamara, MCJD van Eekelen, and Dipl Ing Jan-Peter Doomernik. Digital identity management on blockchain for open model energy system. *Unpublished Masters thesis–Information Science*, 2017.

[26] KA Nyante. Secure identity management on the blockchain. Master's thesis, University of Twente, 2018.

[27] Jamila Alsayed Kassem, Sarwar Sayeed, Hector Marco-Gisbert, Zeeshan Pervez, and Keshav Dahal. Dns-idm : A blockchain identity management system to secure personal data sharing in a network. *Applied Sciences*, 9(15), 2019.

[28] Andreas Grüner, Alexander Mühle, and Christoph Meinel. An integration architecture to enable service providers for self-sovereign identity. In *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*. IEEE, 2019.

[29] Sophie Coutor, Christine Hennebert, and Mourad Faher. Blockchain et identification numérique. 2021.

[30] Paul Dunphy and Fabien AP Petitcolas. A first look at identity management schemes on the blockchain. *IEEE security & privacy*, 16(4), 2018.

[31] Jayana Kaneriya and Hiren Patel. A comparative survey on blockchain based self sovereign identity system. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*. IEEE, 2020.

[32] Nitin Naik and Paul Jenkins. Self-sovereign identity specifications : Govern your identity through your digital wallet using blockchain technology. In *2020 8th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*. IEEE, 2020.

[33] What is a steward. `https://sovrin.org/`. Consultée le 5 mai 2022.

[34] Luis Bathen, German H Flores, Gabor Madl, Divyesh Jadav, Andreas Arvanitis, Krishna Santhanam, Connie Zeng, and Alan Gordon. Selfis : Self-sovereign biometric ids. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[35] Khaled AM Ahmed, Sabry F Saraya, John F Wanis, and Amr MT Ali-Eldin. A self-sovereign identity architecture based on blockchain and the utilization of customer's banking cards : The case of bank scam calls prevention. In *2020 15th International Conference on Computer Engineering and Systems (ICCES)*. IEEE, 2020.

[36] Lifeid. `https://lifeid.io/whitepaper.pdf`. Consultée le 18 mai 2022.

[37] Selfkey. `https://selfkey.org/`. Consultée le 18 mai 2022.

[38] Dirk Van Bokkem, Rico Hageman, Gijs Koning, Luat Nguyen, and Naqib Zarin. Self-sovereign identity solutions : The necessity of blockchain technology. *arXiv preprint arXiv :1904.12816*, 2019.

[39] Andreea-Elena Panait, Ruxandra F Olimid, and Alin Stefanescu. Identity management on blockchain–privacy and security aspects. *arXiv preprint arXiv :2004.13107*, 2020.

[40] Lifeid. `https://www.shocard.com/en.html`. Consultée le 18 mai 2022.

[41] Ibrahim Tariq Javed, Fares Alharbi, Badr Bellaj, Tiziana Margaria, Noel Crespi, and Kashif Naseer Qureshi. Health-id : a blockchain-based decentralized identity management for remote healthcare. In *Healthcare*, volume 9. Multidisciplinary Digital Publishing Institute, 2021.

[42] L'iran légalise officiellement le minage de cryptomonnaies. `https://theblockchainland.com/fr/2019/07/29/liran-legalise-officiellement-le-minage-de-cryptomonnaies/`. Consultée le 18 mai 2022.

[43] The path to self-sovereign identity. `https://www.coindesk.com/markets/2016/04/27/the-path-to-self-sovereign-identity/`. Consultée le 18 mai 2022.

[44] Andreas M Antonopoulos. *Mastering Bitcoin : unlocking digital cryptocurrencies*. " O'Reilly Media, Inc.", 2014.

[45] About javascript. https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript. Consulté le 4 juin 2022.

[46] Solidity. https://github.com/ethereum/solidity. Consulté le 4 juin 2022.

[47] Ganache. https://trufflesuite.com/docs/ganache/. Consulté le 5 juin 2022.

[48] Truffle suite. https://trufflesuite.com/docs/truffle/. Consulté le 5 juin 2022.

[49] Metamask. https://metamask.io/. Consulté le 5 juin 2022.

[50] Ipfs. https://ipfs.io/. Consulté le 5 juin 2022.

[51] Web3.js. https://web3js.readthedocs.io/en/v1.7.3/. Consulté le 5 juin 2022.

[52] Visual studio code. https://code.visualstudio.com/. Consulté le 5 juin 2022.

[53] Remix. https://remix-project.org/. Consulté le 5 juin 2022.

[54] Kevin Atighehchi, Loubna Ghammam, and Morgan Barbier. Id-blockchain. In *Conference w/o proceedings*, 2018.

[55] Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak, and Ali A Ghorbani. Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2019.

[56] Yang Liu, Debiao He, Mohammad S Obaidat, Neeraj Kumar, Muhammad Khurram Khan, and Kim-Kwang Raymond Choo. Blockchain-based identity management systems : A review. *Journal of network and computer applications*, 166, 2020.

[57] Infura ipfs. https://infura.io/product/ipfs. Consulté le 28 juin 2022.

[58] Hsiao-Shan Huang, Tian-Sheuan Chang, and Jhih-Yi Wu. A secure file sharing system based on ipfs and blockchain. In *Proceedings of the 2020 2nd International Electronics Communication Conference*, 2020.