République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université Ain Temouchent Belhadj Bouchaib
Faculté des Sciences et de la Technologie
Département de Mathématiques et Informatique

Projet de Fin d'études
Pour l'obtention du diplôme de MASTER en : Informatique
Spécialité : Cyber Sécurité & Intelligence Artificielle

Thème

Développement d'une application de gestion d'identité décentralisée (DID)

Présenté Par :

- M. KHETTAB Oussama
- M. FLIH Bilel

Soutenu Le : 25/06/2025

Devant le jury composé de :

Dr BOUCHAKOUR ERRAHMANI Hichem	UAT.B. B (Ain Témouchent) Président
Dr BENZERBADJ Ali	UAT.B. B (Ain Témouchent) Examinateur
Dr MEDEDJEL Mansour	UAT.B. B (Ain Témouchent) Encadrant

UAT.B. B

Année Universitaire 2024/2025

Dédicace

C'est avec un énorme plaisir, un cœur ouvert et une immense joie que ce travail est dédié à mes chers parents pour leur soutien inconditionnel,

De même je dédie ce travail pour l'ensemble des enseignants du département de Mathématique et informatique pour leur accompagnement.

Je dédie également ce travail à la mémoire de mon oncle maternel, que Dieu ait son âme, en reconnaissance pour l'amour, les inspirations et les valeurs qu'il m'a transmis.

Enfin, j'adresse une dédicace toute particulière à notre encadrant, Monsieur M. MEDEDJEL pour ses conseils avisés, sa disponibilité et son encadrement précieux.

FLIH BILEL

Dédicace

À ma chère Maman,

Source d'amour inépuisable, de patience et de sacrifices, celle qui a toujours cru en moi et m'a soutenu sans relâche.

À mon cher Papa,

Exemple de force, de sagesse et de persévérance, pour son appui constant et ses précieux conseils.

À mes deux sœurs,

Pour leur affection, leur encouragement, et les sourires qu'elles ont semés dans mes journées difficiles.

À mon frère,

Pour sa présence, son soutien discret mais essentiel, et sa confiance en mes capacités.

À notre encadreur,

Pour son accompagnement, sa rigueur et ses conseils précieux qui ont guidé la réalisation de ce travail.

À nos professeurs,

Pour leur dévouement, leur transmission du savoir et leur contribution à notre formation tout au long de ce parcours.

À toute ma famille,

Pour leur amour, leur fierté et leur présence bienveillante.

À tous mes camarades,

Avec qui j'ai partagé des efforts, des doutes, des réussites et de beaux souvenirs.

À tous ceux qui ont croisé ma route et ont contribué, d'une manière ou d'une autre, à ce parcours. À vous tous, je dédie ce travail avec gratitude, humilité et respect.

KHETTAB Oussama

Remerciements

Avant tout, je rends grâce à Dieu qui nous a accordé Sa guidance.

Un immense merci à mes parents pour leur soutien, leur réconfort et leurs précieuses prières.

Je tiens également à remercier mon professeur encadrant, Monsieur M. MEDEDJEL, qui s'est investi sans compter pour nous jusqu'aux derniers instants, et pour nous avoir orientés avec ses précieux conseils et remarques.

Nous tenons à exprimer notre profonde gratitude à l'ensemble du corps professoral – enseignants et encadrants pédagogiques – qui nous ont accompagnés tout au long de notre parcours académique.

À tous mes enseignants du Département de Mathématiques et Informatique.

Enfin, je tiens à remercier les jurés pour avoir accepté d'examiner et d'évaluer notre travail.

FLIH BILEL

Remerciements

Je souhaiterais tout d'abord rendre un hommage ému à Dieu qui m'a guidé tout au long de ce parcours.

Je remercie de tout cœur mes parents, pour leur amour inconditionnel, l'immuable soutien et les prières qui m'ont aidé durant les moments difficiles.

Je remercie notre encadrant, Monsieur M. MEDEDJEL pour son engagement, sa disponibilité et ses conseils enrichissants qui ont conforté et dirigé ce projet.

Je remercie les enseignants du Département de Mathématiques et Informatique,

Je remercie également les membres du jury pour le temps et l'attention qu'ils ont bien voulu accorder à l'évaluation de ce travail.

Un Grand Merci à Tous !

KHETTAB Oussama

Table des matières

Table Des Figures

Liste des Tableaux

Table des abréviations

2FA	Two-Factor Authentication (Authentification à Deux Facteurs)
AES	Advanced Encryption Standard (Standard de Chiffrement Avancé)
API	Application Programming Interface (Interface de Programmation d'Applications)
Bcrypt	Cryptographic Hashing Algorithm (Algorithme de Hachage Cryptographique)
CID	Content Identifier (identifiant de contenu lié à IPFS dans le stockage des fichiers)
CorDapp	Corda Distributed Application (Application Distribuée Corda)
DAO	Decentralized Autonomous Organization (Organisation Autonome Décentralisée)
DApp	Decentralized Application (Application Décentralisée)
DDOS	Distributed Denial of Service (Attaque par Déni de Service Distribué)
DHT	Distributed Hash Table (Table de Hachage Distribuée)
DID	Decentralized Identifiers (Identifiants Décentralisés)
DLT	Distributed Ledger Technology (Technologie de Registre Distribué)
DSA	Digital Signature Algorithm (Algorithme de Signature Numérique)
ECDSA	Elliptic Curve Digital Signature Algorithm (Algorithme de Signature Numérique à Courbe Elliptique)
ECC	Elliptic Curve Cryptography (Cryptographie à Courbe Elliptique)
EdDSA	Edwards-curve Digital Signature Algorithm (Algorithme de Signature Numérique à Courbe d'Edwards)
EVM	Ethereum Virtual Machine (Machine Virtuelle Ethereum)
FALCON	Fast Fourier Lattice-based Compact Signatures (signatures compactes basées sur les réseaux avec FFT)
GCI	Centralized Identity Management (Gestion Centralisée des Identités)
GCM	Galois/Counter Mode
GDPR	General Data Protection Regulation (Règlement Général sur la Protection des Données - version anglaise)
GUI	Graphical User Interface
HTLC	Hashed TimeLock Contract (Contrat à Horodatage et Verrouillage par Hachage)

HTTPS	HyperText Transfer Protocol Secure (Protocole de Transfert Hypertexte Sécurisé)
IdM	Identity Management (Gestion des Identités)
IdP	Identity Provider (Fournisseur d'Identité)
IPFS	InterPlanetary File System (Système de Fichiers Interplanétaire)
IOT	Internet of Things (Internet des Objets)
JSON	JavaScript Object Notation
MD5	Message Digest Algorithm 5 (Algorithme de Hachage MD5)
MRTD	Machine Readable Travel Document (Document de Voyage Lisible par Machine)
OACI	International Civil Aviation Organization (Organisation de l'Aviation Civile Internationale)
OAuth	Open Authorization (Autorisation Ouverte)
OpenID	Decentralized Authentication Protocol (Protocole d'Authentification Décentralisé)
P2P	Peer-to-Peer (Pair à Pair)
PBKDF2	Password-Based Key Derivation Function
PKI	Public Key Infrastructure (Infrastructure à Clé Publique)
POA	Proof of Authority (Preuve d'Autorité)
POS	Proof of Stake (Preuve d'Enjeu)
POW	Proof of Work (Preuve de Travail)
RPC	Remote Procedure Call
RSA	Rivest-Shamir-Adleman (Algorithme de Cryptographie Asymétrique)
SAML	Security Assertion Markup Language (Langage de Déclaration de Sécurité)
SAML 2.0	Security Assertion Markup Language 2.0 (Version 2.0 du Protocole SAML)
SDK	Software Development Kit (Kit de Développement Logiciel)
SHA-1	Secure Hash Algorithm 1 (Algorithme de Hachage Sécurisé 1)
SHA-256	Secure Hash Algorithm 256-bit (Algorithme de Hachage Sécurisé 256 bits)
SP	Service Provider (Fournisseur de Service)
SPOF	Single Point of Failure (Point de Défaillance Unique)
SSI	Self-Sovereign Identity (Identité auto-souveraine)
SSO	Single Sign-On (Authentification Unique)
UI	User Interface

URI	Uniform Resource Identifier (Identifiant de Ressource Uniforme)
UX	User Experience (Expérience Utilisateur)
VC	Verifiable Credentials (Justificatifs Vérifiables)
ZKP	Zero-Knowledge Proof (Preuve à Connaissance Nulle)

Introduction Générale

Dans un monde de plus en plus connecté, l'identité numérique joue un rôle fondamental pour accéder aux divers services en ligne tels que les plateformes d'achat, les services bancaires ou médicaux. Cependant, la majorité des systèmes de gestion d'identité numérique repose sur des structures centralisées qui exposent les utilisateurs à de multiples risques : vol d'informations personnelles, usurpation d'identité, perte de données et violations de la vie privée. Ces architectures présentent également des vulnérabilités comme les points uniques de défaillance (SPoF - Single Points of Failure), le manque de transparence dans la consultation et la modification des données personnelles, ainsi que la centralisation d'informations sensibles côté serveur, augmentant considérablement la surface d'attaque.

L'apparition de la technologie blockchain, caractérisée par son immutabilité et son consensus distribué, a engendré un nouveau paradigme en matière d'identité numérique : l'identité décentralisée (DID : Decentralized Identifier). Cette approche novatrice permet aux individus de créer, gérer et contrôler leur propre identité sans dépendre d'une autorité centrale, incarnant ainsi le concept d'identité auto souveraine (SSI : Self-Sovereign Identity). Cette évolution fondamentale redéfinit la relation entre les utilisateurs et leurs données personnelles dans l'écosystème numérique.

Grâce aux techniques cryptographiques, la SSI permet à l'utilisateur un contrôle total sur ses données personnelles, tout en garantissant leur authenticité, intégrité, confidentialité et traçabilité. Dans ce mémoire, notre solution consiste à développer une application de gestion des identités décentralisées basée sur la technologie blockchain, permettant à l'utilisateur de :

- Accéder à une plateforme numérique sécurisée
- Créer différents types d'identités (nationale, bancaire, professionnelle, etc.)
- Modifier ou partager une identité existante
- Recevoir et vérifier des identités partagées par d'autres utilisateurs

Ce mémoire est structuré en trois chapitres :

Chapitre 1 : Présentation de l'identité numérique, description des systèmes de gestion d'identités et de leurs différents types, définition du concept SSI (Self-Sovereign Identity) et aperçu de l'identité décentralisée.

Chapitre 2 : Exploration de la technologie blockchain, ses typologies, ses avantages, ses limites et ses champs d'application.

Chapitre 3 : Analyse préalable de notre solution de gestion d'identité décentralisée, conception de l'application et implémentation détaillée de la solution proposée.

Chapitre I : Identité numérique

1. Introduction

L'évolution de la technologie nous a propulsé vers un nouveau monde appelé le monde digital, dans lequel l'identité numérique joue un rôle crucial dans la vie des utilisateurs de ce monde en façonnant la manière dont les individus interagissent avec les services en ligne, les réseaux sociaux et les transactions électroniques, Bien que l'identité numérique présente de grands avantages, elle nécessite d'être améliorée pour faire face à certains inconvénients, tels que le vol d'identité, la falsification des informations, la centralisation excessive des données personnelles et le manque de contrôle des utilisateurs sur leur propre identité.

Face à ces enjeux, plusieurs modèles de gestion de l'identité numérique ont vu le jour, évoluant des systèmes centralisés aux approches fédérées, jusqu'aux solutions décentralisées plus récentes.

Ce chapitre explore les fondements de l'identité numérique, son évolution au fil du temps, et les différents modèles qui ont marqué son développement. Une attention particulière sera portée à l'identité décentralisée et aux systèmes d'identités auto-souverains (ou SSI : *Self-Sovereign Identity*, en mettant en lumière les solutions les plus connues dans ce domaine.

2. Qu'est-ce que l'identité numérique ?

Nous pouvons considérer notre identité numérique comme une version en ligne unique de nous-mêmes. De la même manière que nous avons un nom, un visage et des informations sur nous-mêmes dans la vie réelle, nous avons un nom et des informations sur nous-mêmes sur Internet.

Cette version en ligne de nous permet aux sites Web et aux applications de reconnaître notre identité lorsque nous les utilisons. C'est similaire à présenter notre carte d'identité lorsque nous souhaitons entrer dans un édifice par exemple.

Notre identité numérique peut inclure notre adresse électronique, notre nom d'utilisateur et peut-être une photo de nous-mêmes. Cela permet aux sites Web de se souvenir de nous et de protéger nos informations lorsque nous achetons en ligne, communiquons avec des amis ou jouons à des jeux. Tout comme nous sommes prudents concernant notre identification physique, nous devons également être prudents concernant notre identité numérique, afin que seules les personnes autorisées puissent accéder à nos informations et les utiliser.

Une identité numérique, d'un point de vue institutionnel, est une combinaison d'informations et de références électroniques qui identifient de manière unique une personne ou une entité dans le monde numérique. Les particuliers, les organisations et les gouvernements développent souvent des identités numériques, qui comprennent des informations telles qu'un identifiant de connexion, un mot de passe, une adresse électronique et d'autres caractéristiques personnelles. Ces identités sont utilisées pour accéder à des services en ligne, effectuer des transactions et participer à de nombreuses activités sur Internet. [1]

Des normes sur la gestion de l'identité sont en cours dans de nombreuses organisations. Certaines de ces normes incluent la définition de l'identité. Par exemple, la Recommandation UIT-T Y.2720 définit l'identité comme « des informations sur une entité qui sont suffisantes pour identifier cette entité dans un contexte particulier ».

Selon la norme Y.2720, une identité se compose de trois types de données différents : l'identifiant, Informations où attestation d'identification (credentials) et les attributs.

• Identificateurs : Une série de chiffres, de caractères et de symboles ou toute autre forme de données utilisée pour identifier un sujet. Les identificateurs peuvent être définis dans le temps et/ou dans l'espace. Par exemple, une URI est globalement unique au fil du temps. Les pseudonymes ne peuvent être temporels et efficaces que pour un service spécifique. Quelques exemples de Gestion de l'identité : concepts, technologies et systèmes sont les noms de compte utilisateur, les numéros de passeport, les numéros de téléphone portable, les numéros d'employé, les pseudonymes et les URI.

• Informations d'identification (*Credentials)* : un ensemble de données fournissant des preuves pour des affirmations sur des parties ou des identités entières. Une référence peut être générée sur la base d'une ou de plusieurs références. Par exemple, les mots de passe, les certificats numériques, les empreintes digitales, les tickets Kerberos et les assertions SAML.

• Attributs : ensemble de données qui décrit les caractéristiques d'un sujet. Les données comprennent les informations fondamentales permettant d'identifier un sujet (par exemple, son nom complet, son domicile et sa date de naissance), ses préférences et les informations générées à la suite de ses activités. Il s'agit, par exemple, des prénoms, des domiciles, des âges, des sexes, des rôles, des titres, des affiliations, des registres d'activité et des réputations [2].

La figure I.1 présente les composants de l'identité numérique, ou elle présente deux identités numériques d'une même personne, une professionnelle (employée avec certificat et affiliation) et l'autre personnelle (réseau social avec préférences et localisation). Elle illustre la dissociation des rôles et attributs selon les contextes.

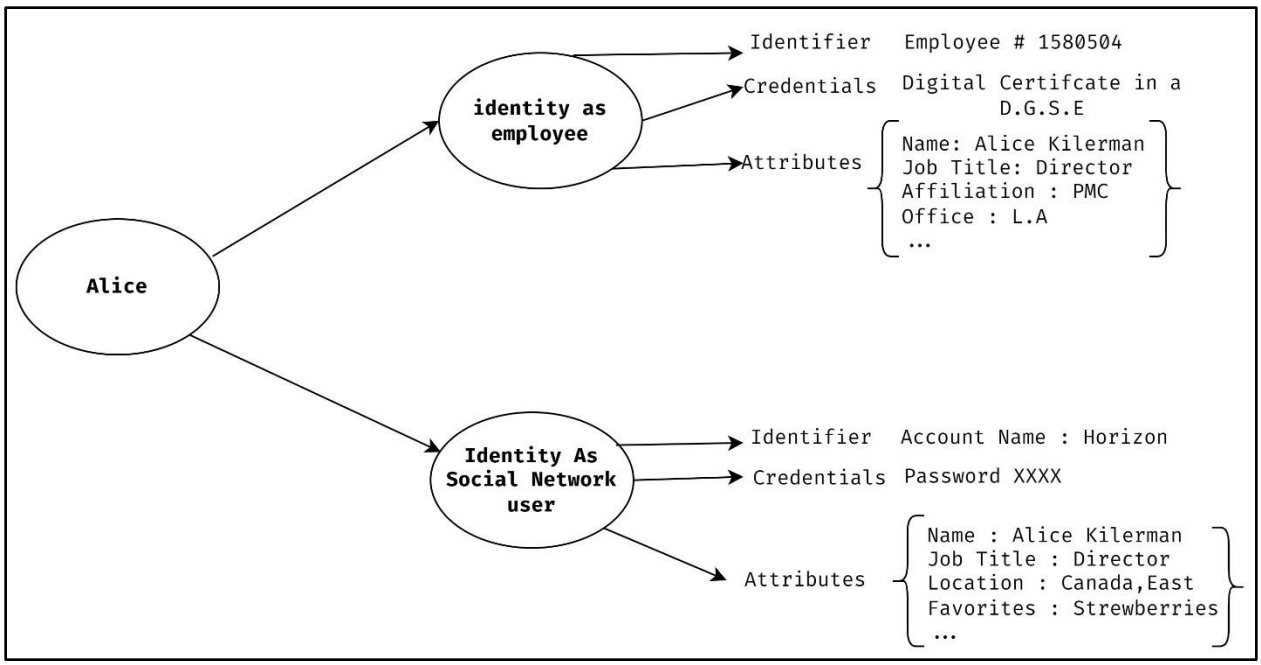

Figure I-1 Les identités numériques se composent d'identifiants, de références et d'attributs [1]

3. L'évolution des identités numériques

L'évolution des identités numériques a connu un développement significatif au cours du temps, influencée par divers facteurs [3]:

- Sécurité : Les combinaisons traditionnelles de nom d'utilisateur et de mot de passe étaient vulnérables au vol d'identité et au piratage. La cybercriminalité devenant un problème majeur, des méthodes plus sûres de vérification et d'authentification de l'identité étaient nécessaires.
- Commodité : Les utilisateurs avaient besoin d'un moyen plus fluide et plus simple d'accéder à de nombreuses plateformes sans avoir à se souvenir de plusieurs noms d'utilisateur et mots de passe à mesure que les services en ligne et le e-commerce proliféraient.
- Personnalisation : En adaptant les informations et les services aux goûts de chacun, les fournisseurs de services ont tenté de personnaliser l'expérience des utilisateurs. Pour ce faire, ils avaient besoin d'une méthode permettant d'identifier de manière unique les consommateurs sur plusieurs plateformes et services.

- Confiance et responsabilité : Pour instaurer la confiance et la responsabilité dans les interactions en ligne, l'identité numérique est requise. Elle tient les particuliers et les entreprises responsables de leurs activités et soutient la conformité légale et réglementaire dans le domaine numérique.
- Interopérabilité : à mesure que le nombre de services en ligne augmentait, une méthode standardisée de vérification et d'authentification des identités numériques sur plusieurs plateformes et applications est devenue nécessaire.

Divers systèmes d'identité numérique ont été développés afin de répondre aux exigences croissantes en matière de sécurité, de confidentialité et de vérifiabilité. Parmi les approches couramment utilisées, on trouve la biométrie, l'authentification à deux facteurs (2FA), les certificats numériques, ainsi que l'infrastructure à clé publique (PKI). Plus récemment, les systèmes d'identité décentralisés ont émergé comme une alternative innovante, notamment grâce à l'intégration de technologies telles que la blockchain, qui joue un rôle central dans certaines de ces solutions.

Le sujet de l'identification numérique évolue à mesure que de nouvelles technologies émergent, telles que l'intelligence artificielle et l'apprentissage automatique, qui sont utilisées pour améliorer les procédures de vérification d'identité et de détection des fraudes tout en protégeant la vie privée et la sécurité des utilisateurs. Néanmoins, des difficultés telles que la confidentialité des données, l'autorisation de l'utilisateur et l'équilibre entre la commodité et la sécurité dans la gestion de l'identité numérique, persistent.

Le concept d'identité numérique a évolué au cours des dernières décennies à mesure que la technologie progressait et qu'Internet devenait plus omniprésent. Voici un bref historique des identités numériques :

- Identités numériques : (années 1960) Au début d'Internet, les identités numériques se limitaient souvent aux noms d'utilisateur et aux mots de passe que les utilisateurs créaient pour accéder aux services en ligne.

- Réseaux sociaux : Avec l'essor des plateformes de réseaux sociaux telles que MySpace et Facebook au milieu des années 2000, les identités numériques ont commencé à prendre une dimension plus sociale. Les utilisateurs pouvaient créer des profils, partager des informations personnelles et se connecter avec d'autres personnes d'une manière qui n'était pas possible auparavant.
- Appareils mobiles : L'adoption généralisée des smartphones et autres appareils mobiles à la fin des années 2000 et au début des années 2010 a encore élargi l'utilisation des identités numériques. Les utilisateurs pouvaient accéder à leurs comptes de n'importe où, et les applications mobiles facilitaient plus que jamais la création et la gestion des identités numériques.
- Authentification numérique : (1969 jusqu'à présent) à mesure que les services et les transactions en ligne sont devenus plus courants, le besoin d'une authentification numérique sécurisée s'est accru. L'authentification à deux facteurs (2FA : 2000s), l'authentification biométrique et d'autres mesures de sécurité avancées se sont répandues.

Blockchain : Ces dernières années, la technologie blockchain est apparue comme une nouvelle façon de gérer les identités numériques. Avec la blockchain, les utilisateurs peuvent créer une identité numérique décentralisée qui n'est pas contrôlée par une seule entité, ce qui peut offrir une plus grande confidentialité et sécurité. [2]

4. La gestion des identités

La gestion des identités regroupe les processus, politiques et technologique assurant que seules les entités autorisées accèdent aux ressources numériques. Les systèmes de gestion des identités et des accès évoluent pour renforcer la sécurité tout en optimisant l'expérience utilisateur [4].

4.1. Les défis de la gestion d'identité

La gestion de l'identité numérique doit concilier sécurité, respect de la vie privée et facilité d'utilisation. Les solutions actuelles, souvent incompatibles entre elles, ne répondent pas pleinement à ces défis. Cette problématique soulève plusieurs questions fondamentales pour les organisations et la société [2] :

- Comment garantir un accès sélectif et contextuel aux identités numériques, en s'assurant qu'elles ne soient disponibles qu'aux entités légitimes dans des circonstances appropriées ?
- Par quels mécanismes établir des relations de confiance solides entre les différents acteurs impliqués dans les échanges d'informations d'identité ?
- Quelles mesures mettre en place pour prévenir efficacement l'usurpation et l'utilisation frauduleuse des identités ?
- Comment concevoir des systèmes d'identité qui soient à la fois évolutifs, conviviaux et économiquement viables ?

4.2. Les modèles de gestion d'identité numérique

Les modèles d'identité numérique sont souvent classés en trois grandes catégories, en fonction du niveau de contrôle et de l'utilisation des données d'identité.

4.2.1. Modèle centralisé

La gestion centralisée des identités repose sur la collecte et le stockage des données d'identité des utilisateurs. Grâce à la gestion centralisée des identités, les utilisateurs peuvent accéder à toutes leurs applications, sites web ou autres systèmes avec le même ensemble d'identifiants. Cela améliore l'expérience utilisateur car il suffit de saisir un seul nom d'utilisateur et un seul mot de passe, mais cela peut entraîner une vulnérabilité accrue si vos identifiants sont compromis [5].

La figure I.2 représente un schéma simplifié d'un système de gestion d'identité numérique centralisé.

➢ Fonctionnement :

Le fonctionnement d'un modèle centralisé de gestion des identités peut être décrit à travers les étapes suivantes :

- Inscription : Les utilisateurs créent un compte unique lors de leur inscription, comprenant des informations telles que nom d'utilisateur, mot de passe et adresse e-mail. Ces données sont stockées dans une base de données centrale [5] [6].

- Authentification et autorisation : L'authentification vérifie l'identité de l 'utilisateur via des méthodes telles que le mot de passe ou l'authentification multi facteur.

- Contrôle d'accès : Une fois authentifié, des droits d'accès sont attribués selon les rôles ou attributs des utilisateurs. Ce modèle permet d'appliquer des politiques de sécurité cohérentes à l'ensemble des ressources, facilitant la gouvernance, l'audit et la conformité.

- Audit et l'analyse : La gestion centralisée des identités comprend des fonctionnalités d'audit qui permettent de suivre les accès et les modifications effectuées sur les identités. Cela facilite la détection d'activités suspectes ou non conformes, renforçant ainsi la sécurité et la conformité des systèmes d'information [6] .

➢ Avantages :

- Simplicité : Les utilisateurs peuvent accéder à plusieurs applications avec un seul ensemble d'identifiants, améliorant ainsi l'expérience utilisateur.
- Sécurité améliorée : Une gestion centralisée permet une surveillance constante et une réponse rapide aux menaces potentielles grâce à l'audit et au L'analyse. [6] [7] [8]

➢ Inconvénients :

- Point de défaillance unique : La centralisation crée un risque accru en cas de violation de données. Si le système est compromis, toutes les identités peuvent être exposées.
- Vulnérabilité aux cyberattaques : Les systèmes centralisés sont souvent ciblés par des cyberattaques, ce qui peut entraîner des fuites de données sensibles. [6] [7] [8]

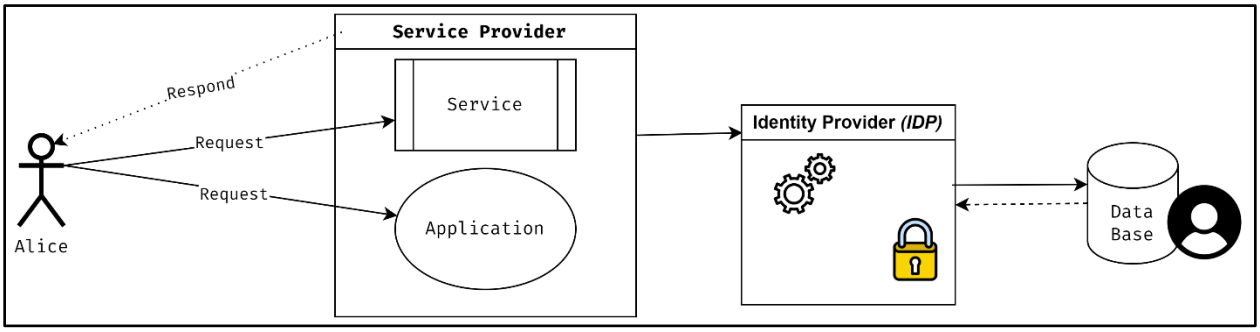

Figure I-2 Modèle de gestion d'identité centralisé [1]

Le fournisseur de service (Service Provider) désigne l'entité responsable de la mise à disposition d'un service ou d'une application à l'utilisateur. Il est chargé de traiter les requêtes formulées par ce dernier et d'y apporter une réponse appropriée.

Le fournisseur d'identité (*Identity Provider*) est chargé de la gestion des identités numériques, ou il authentifie les utilisateurs et fournit des informations d'identification au *Service Provider*. La base de données sert à stocker les identifiants et les informations des utilisateurs.

4.2.2. Modèle fédéré

Le modèle de la gestion d'identité fédéré permet à un utilisateur d'accéder à plusieurs organisations ou il permet le partage des informations d'authentification entre ces organisations grâce à la relation de confiance établie entre les différentes entités faisant partie du système tel que IDP (fournisseurs d'identité) et SP (Fournisseurs de service)

Ce modèle est basé sur le mécanisme d'authentification unique (SSO : *single sign-on*) qui permet à l'utilisateur de ne s'authentifier qu'une seule fois pour accéder à l'ensemble des services proposés par les organisations du système.

- Fonctionnement :
  - Authentification : Lorsqu'un utilisateur souhaite accéder à un service d'une organisation partenaire, il est redirigé vers un IdP pour l'authentification.
  - Vérification : L'IdP vérifie l'identité de l'utilisateur en utilisant les informations d'identification fournies.
  - Accès : Une fois authentifié, l'IdP transmet les informations nécessaires au SP, qui accorde l'accès au service demandé sans nécessiter une nouvelle authentification [9] [10] [11].

La figure I.3 représente un processus d'authentification et d'autorisation entre plusieurs fournisseurs de services et un fournisseur d'identité (IDP).

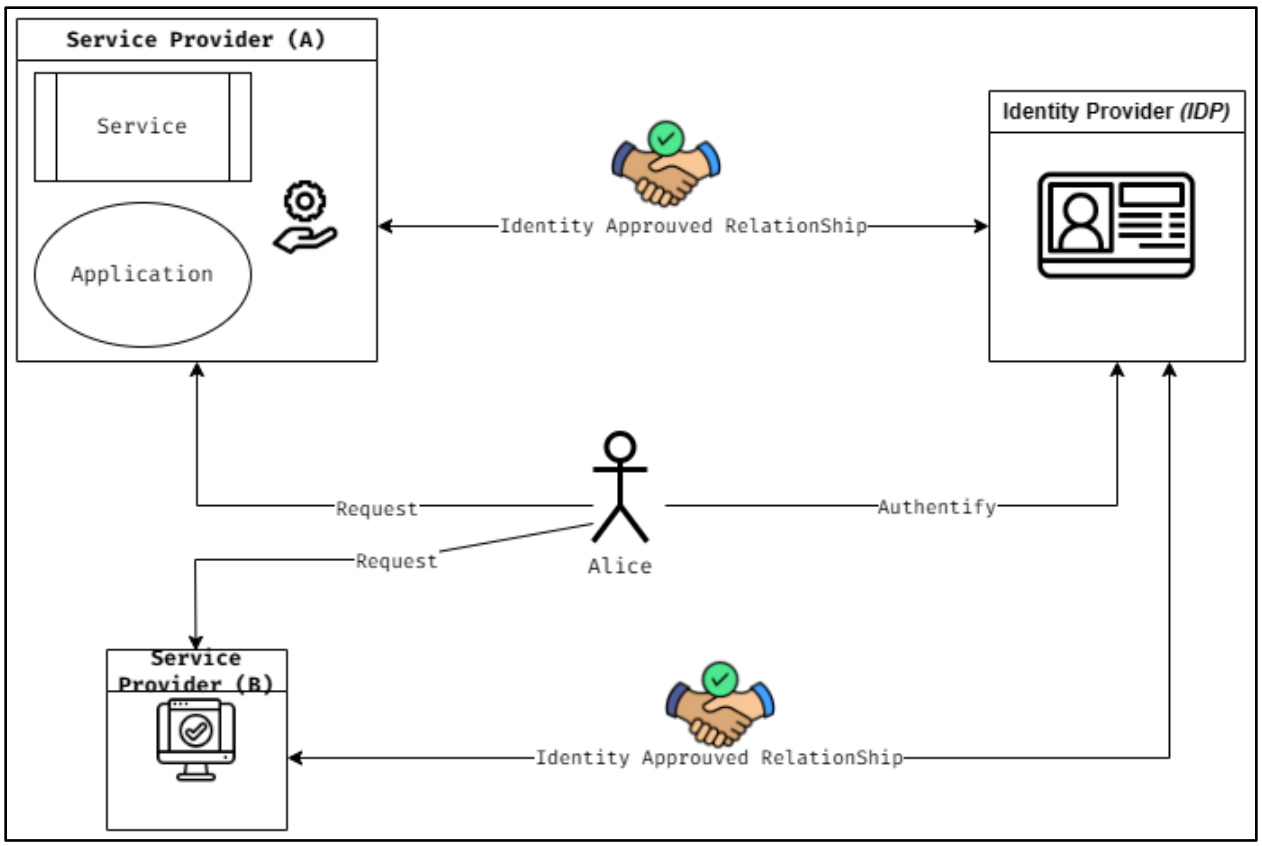

Figure I-3 Schéma du modèle de gestion d'identité Fédéré [1]

Les fournisseurs de services (A) et (B) sont des entités fournissant des services, tandis que le fournisseur d'identité (IDP) effectue la vérification des identités des utilisateurs. L'application représente le service ou l'application qui nécessite une authentification pour l'accès. La relation d'identité approuvée établit une connexion de confiance entre l'IDP et les fournisseurs de services, ce qui facilite l'authentification par l'IDP. L'authenticité de la demande et la demande de service capturent les échanges de demande d'authentification et de demande de service entre les parties. Graphiquement, il a été capturé comment un utilisateur ou une application est authentifié par l'IDP pour accéder aux services de A et B en utilisant une relation de confiance.

➢ Avantages

- Sécurité Améliorée : En déléguant l'authentification à un IdP, les organisations réduisent le besoin de gérer directement les informations d'identification des utilisateurs.
- Expérience Utilisateur Simplifiée : Les utilisateurs bénéficient d'un accès transparent à divers services avec une seule authentification.
- Scalabilité : Le modèle fédéré peut évoluer facilement avec l'ajout de nouveaux partenaires sans nécessiter des modifications majeures dans les systèmes existants. [9] [12]

La figure ci-dessous illustre ce flux d'authentification SAML de manière simplifiée au sein du modèle fédéré de la gestion d'identité

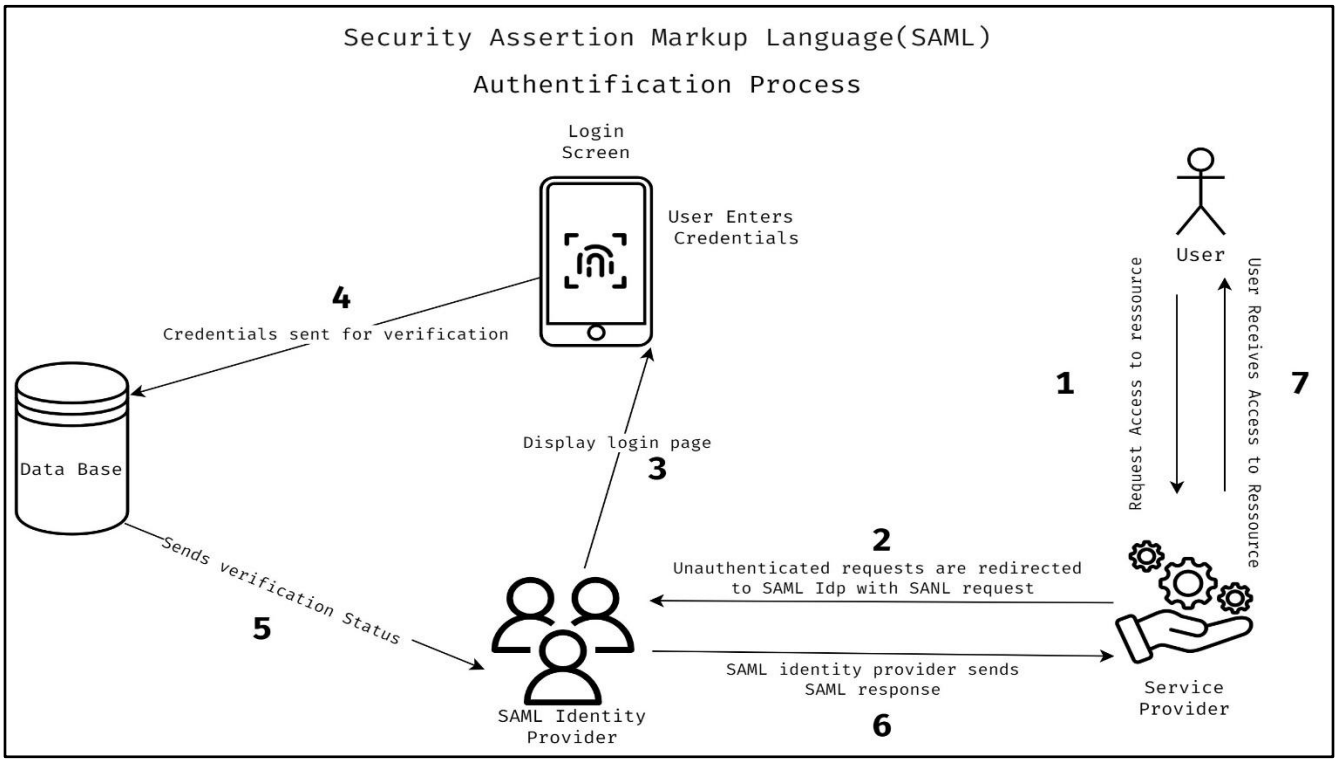

Figure I-4 : Processus d'authentification basé sur le protocole SAML

4.2.3. Modèle décentralisé

La gestion d'identité décentralisée représente un paradigme innovant permettant aux utilisateurs de conserver la maîtrise complète de leurs données personnelles. Ce concept a pris son essor avec l'avènement de la technologie Blockchain, qui a profondément renouvelé les approches traditionnelles de gestion des attributs d'identité. L'identité décentralisée représente un changement vers une forme d'identité auto-souveraine et contrôlée par l'utilisateur, favorisant des échanges de données sécurisés et fiables. Dans un système d'identité décentralisée, les individus ont la possibilité de partager sélectivement des données avec des plateformes, assurant ainsi un niveau de sécurité élevé.

Les utilisateurs peuvent s'authentifier pour accéder à des services et plateformes sans dépendre de méthodes d'authentification tierces, améliorant ainsi la confidentialité et la sécurité. [13] [14] [15] [16]

➢ Principes fondamentaux :

- Autonomie de l'utilisateur : L'identité décentralisée place l'individu au centre du dispositif, lui permettant de révéler uniquement les informations nécessaires à chaque contexte de vérification. Cette approche soutient un cadre de confiance où toutes les interactions entre utilisateurs, organisations et appareils s'effectuent de manière transparente et sécurisée.

- Technologies de registres distribués (DLT) : Ce modèle s'appuie sur des technologies comme la blockchain pour établir un réseau d'identités robuste et décentralisé, éliminant ainsi les points de défaillance uniques présents dans les systèmes centralisés traditionnels.

- Portefeuilles numériques L'architecture décentralisée permet aux utilisateurs de conserver leurs données d'identité dans un portefeuille numérique personnel, généralement hébergé sur leur appareil mobile. Ce portefeuille génère et gère une paire de clés cryptographiques (publique et privée), permettant à l'utilisateur de ne partager que les informations strictement nécessaires pour chaque transaction. [13] [15] [16].

La figure ci-dessous donne une vue générale sur ce modèle :

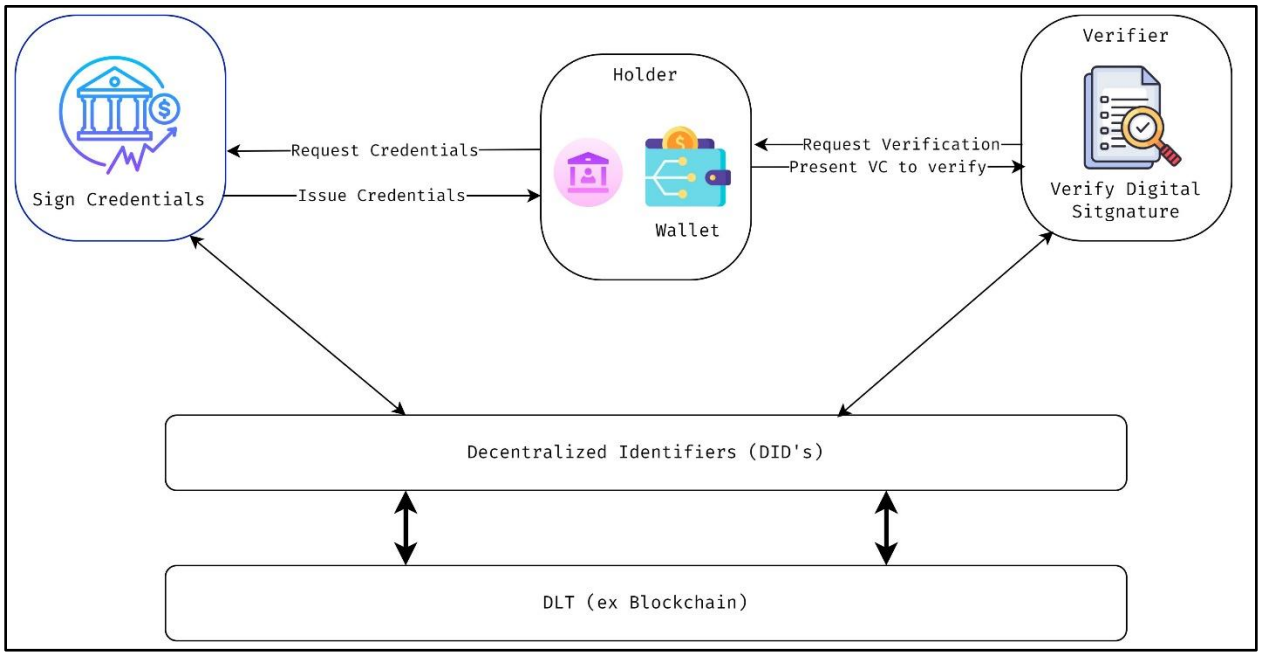

Figure I-5 Schéma du modèle de gestion d'identité décentralisé [36]

Cette figure illustre un système de gestion décentralisée des identités basé sur des identifiants décentralisés (DID) et des credentials vérifiables (VC). Elle met en évidence les interactions entre les différents acteurs : le Holder (détenteur des identifiants), le Wallet (portefeuille numérique), l'Issuer (émetteur des credentials) et le Verifier (vérificateur), le tout s'appuyant sur une technologie de registre distribué (DLT) comme la blockchain. Les étapes clés incluent l'émission, la demande, la présentation et la vérification des credentials, garantissant une authentification sécurisée et sans confiance centralisée.

Dans ce modèle, l'émetteur joue un rôle fondamental en tant qu'entité responsable de la création, de la signature cryptographique et de la transmission des VC au titulaire. Ce dernier les conserve en toute sécurité dans son portefeuille numérique personnel.

Lorsqu'une vérification devient nécessaire, le titulaire peut présenter ses attestations à un vérificateur, déclenchant ainsi un processus d'authentification. Cette vérification comporte notamment une validation rigoureuse des signatures numériques, garantissant l'intégrité et l'authenticité des informations présentées.

Les identifiants décentralisés (DID) constituent la pierre angulaire de ce système. Ces identifiants uniques sont administrés via une technologie de registre distribué (DLT), généralement une blockchain, assurant ainsi une gestion des identités caractérisée par sa transparence, sa résilience et son niveau élevé de sécurité. Cette architecture élimine efficacement la nécessité d'autorités centralisées tout en préservant la confiance dans l'ensemble du système.

➢ Fonctionnement :

L'identité décentralisée sur une DLT fonctionne en permettant aux utilisateurs de contrôler leurs identités numériques sans dépendre d'autorités centrales. Au lieu de s'appuyer sur des fournisseurs d'identité centralisés, chaque utilisateur possède un identifiant unique et sécurisé, protégé par la cryptographie. [17] [18]

Le fonctionnement de l'identité décentralisé est basé sur les identifiants décentralisés (DID) qui sont des identifiants uniques et indépendant de la centralisation associant une clé publique à l'identité numérique d'un utilisateur

Ils permettent la vérification de l'identité au sein de la blockchain, les VC qui servent à envoyer, posséder et vérifier les requêtes sur l'identité d'un utilisateur tout en assurant l'intégrité et la facilité lors de la vérification au sein de la blockchain.

➢ Avantages de de la gestion décentralisé de l'identité

L'identité décentralisée confère aux utilisateurs un contrôle accru sur leurs données personnelles tout en les protégeant des attaques massives et en renforçant leur confidentialité :

- Sécurité renforcée : La cryptographie avancée et l'architecture distribuée de la blockchain réduisent considérablement les risques de vol d'identité et de fraude grâce à des enregistrements immuables des identifiants.
- Confidentialité améliorée : Les utilisateurs partagent uniquement les informations nécessaires et s'authentifient sans intermédiaires, limitant l'exposition de leurs données personnelles.
- Souveraineté des données : Les individus conservent la propriété et le contrôle total de leurs informations personnelles.

- Efficacité des vérifications : Le système permet des validations d'identité rapides et fiables pour les organisations.
- Autonomie utilisateur : Grâce aux identifiants décentralisés (DID) et au chiffrement privé, les utilisateurs valident leur identité de manière sécurisée auprès de chaque service.
- Interopérabilité : La portabilité des données entre différentes plateformes et services est considérablement facilitée. [15] [16] [19] [20].

5. L'identité auto-souveraine (SSI : *Self-sovereign Identity*)

5.1. Définition

L'identité auto-souveraine est une approche novatrice dans le domaine de la gestion de l'identité numérique qui donne aux individus un contrôle total sur leurs données personnelles et leurs modalités de partage dans le monde numérique. Ce paradigme s'appuie sur des principes cryptographiques avancés et exploite la technologie blockchain pour permettre aux utilisateurs de créer, gérer et confirmer leurs identités numériques sans dépendre d'autorités centrales ou de fournisseurs d'identité. La SSI est un changement fondamental dans le modèle d'identification numérique, permettant aux individus la gouvernance complète de leurs données, stockées sur un réseau décentralisé. [1].

Il convient de noter que la SSI est un modèle plus global qui utilise les DID pour mettre en œuvre une identité auto-souveraine. [1].

La SSI offre plusieurs avantages, notamment :

- Confidentialité : La SSI confère aux individus un contrôle total sur leurs données personnelles, leur permettant de décider précisément quelles informations à partager et avec qui. Ce niveau de contrôle renforce la protection de la vie privée en empêchant l'accès non autorisé à leurs données.
- Sécurité : La SSI repose sur des mécanismes de chiffrement avancés et sur un stockage décentralisé, assurant une protection renforcée des données personnelles contre le piratage et les menaces cybernétiques. Cette architecture réduit significativement les risques d'accès non autorisé, de vol d'identité ou d'usage frauduleux des informations sensibles.
- Interopérabilité : La SSI est conçue pour être interopérable sur différentes plateformes et services, ce qui signifie que les individus peuvent utiliser leurs identités numériques pour accéder à une large gamme d'applications et de services. Cela réduit le besoin de multiples identifiants et mots de passe et facilite la gestion de l'identité numérique sur différentes plateformes.
- Confiance : La SSI repose sur un modèle de confiance distribué dans lequel les individus gardent le contrôle exclusif de leurs données personnelles et choisissent avec qui les partager selon le principe du « besoin de savoir ». Ce cadre renforce la

transparence et la responsabilité, favorise des relations de confiance entre les utilisateurs et les organisations, et contribue à limiter les violations de données et autres incidents de sécurité.

- Flexibilité : La SSI offre une grande flexibilité et adaptabilité qui peuvent être personnalisées pour répondre aux besoins de différents individus et organisations. Cela facilite l'intégration et le déploiement de solutions d'identité numérique adaptées à des cas d'utilisation et des exigences spécifiques. [1]

La figure suivante met en évidence les différences fondamentales entre le modèle centralisé de gestion des identités et le modèle décentralisé reposant sur le concept d'identité auto souveraine

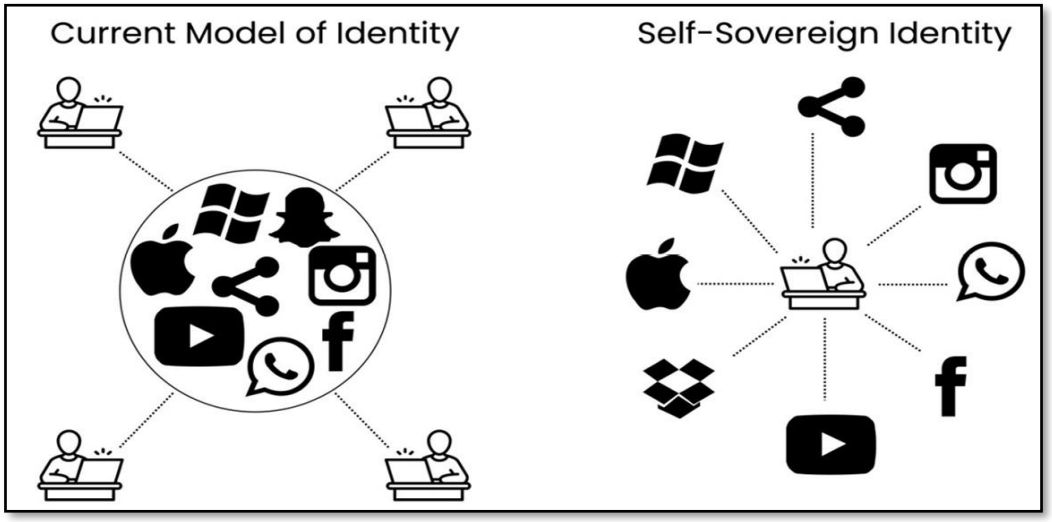

Figure I-6 Modèle actuel d'identité centralisé vs modèle basé sur SSI [1]

5.2. Cryptographie en SSI

La cryptographie constitue le pilier fondamental garantissant la sécurité, la confidentialité et le contrôle de l'identité numérique dans l'écosystème SSI. Sans entrer dans les complexités techniques, nous pouvons aborder les principes essentiels qui sous-tendent ce système. L'architecture SSI s'appuie sur plusieurs composants cryptographiques clés : les paires de clés publiques et privées, les signatures numériques et les attestations vérifiables. Ces éléments agissent comme un système sophistiqué de verrous et de clés pour votre identité numérique. Ces mécanismes cryptographiques permettent de préserver la confidentialité tout en partageant de manière sélective et sécurisée les informations d'identification avec des tiers. Ils constituent l'infrastructure technique essentielle qui rend possible une identité numérique véritablement auto-souveraine et sécurisée [1].

6. L'identité décentralisée

La décentralisation constitue le fondement opérationnel de l'identité auto-souveraine (SSI) en permettant la mise en œuvre concrète de ses principes. Ce modèle repose sur un ensemble de composants et d'entités essentielles qui assurent la sécurité, la confidentialité et l'autonomie promises par les services d'identité décentralisés. Dans ce qui suit, nous présentons les principaux éléments constitutifs de l'identité décentralisée, en commençant par les différents acteurs impliqués dans ce système.

6.1 Acteurs et composants de l'identité décentralisée

6.1.1 Les acteurs principaux

Les acteurs de ce modèle sont :

- Le Titulaire (*Holder*) : c'est l'individu ou l'entité qui reçoit et contrôle ses attestations vérifiables (VC) dans des portefeuilles numériques. Il décide quand, comment et avec qui partager ces informations. L'Émetteur (*Issuer*) : c'est l'entité de confiance (organisations ou autorité) qui crée et signe les VC. Il certifie la validité des informations contenues dans ces attestations. Le vérificateur (*Verifier*) : l'entité qui demande et vérifie les attestations présentées par les titulaires. Il valide l'authenticité et l'intégrité de ces attestations sans nécessairement contacter l'émetteur.

La figure ci-dessous illustre l'interaction de ces acteurs dans le triangle de confiance de la SSI.

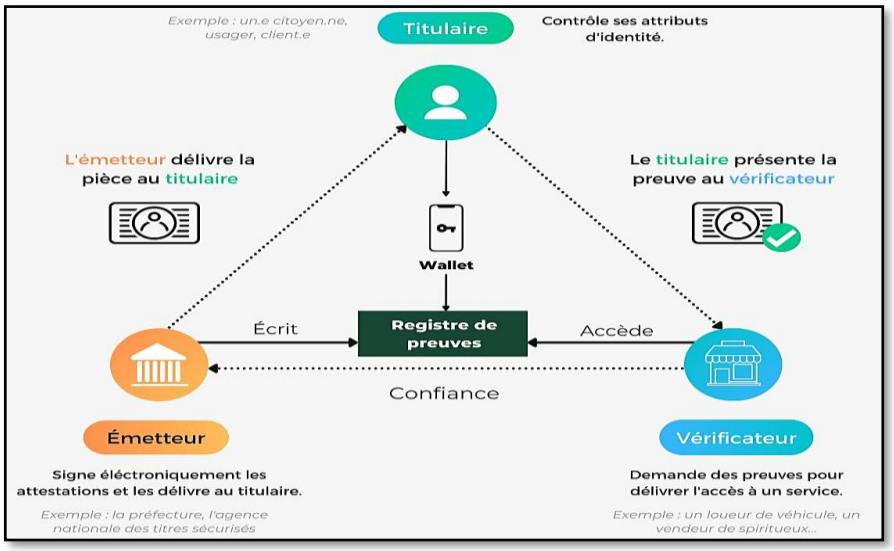

Figure I-7 Le triangle de confiance : base de l'identité souveraine [17]

6.1.2 Les composants essentiels

- Identifiants décentralisés (DID) : Des identifiants uniques créés et contrôlés par l'utilisateur, sans dépendre d'une autorité centrale. Ils permettent d'établir une identité numérique souveraine sur la blockchain.

- Attestations vérifiables (VC) : Des documents numériques signés cryptographiquement par une entité de confiance, prouvant certaines informations sur une identité (ex. : diplôme, permis), que d'autres peuvent vérifier sans contacter l'émetteur.

- Portefeuilles numériques : Des applications (comme MetaMask) permettant aux utilisateurs de gérer leurs identifiants, clés privées, attestations, et d'interagir avec des services décentralisés.

- Registres décentralisés : Des bases de données distribuées (comme les blockchains) où sont publiés les DIDs ou les métadonnées associées, garantissant transparence, intégrité et absence de contrôle central.

6.1.2.1. Identifiants décentralisés (DIDs)

Les DID sont un nouveau type d'identifiant qui est créé et géré par les individus eux-mêmes plutôt que par une autorité ou une organisation centralisée. Les DIDs sont basés sur la technologie blockchain et utilisent la cryptographie à clé publique privée pour fournir un moyen sécurisé et décentralisé de gérer l'identité numérique. Les DIDs peuvent être utilisés pour authentifier l'identité, établir des relations de confiance et contrôler l'accès aux informations personnelles. [1]

6.1.2.2. VC (Verifiable Credentials)

Les VC sont similaires aux badges numériques en ce sens qu'elles nous permettent de démontrer qui nous sommes et ce que nous pouvons accomplir en ligne. Ces informations d'identification sont extrêmement cruciales à l'ère numérique d'aujourd'hui, où nous divulguons une grande partie de nos informations personnelles en ligne. Elles nous permettent de mieux contrôler nos données, d'améliorer notre confidentialité et de rendre notre vie en ligne plus sûre. [1]

6.1.2.3. Portefeuille numérique (Wallet)

Un portefeuille numérique se compose logiciel (d'application mobile ou web) (et éventuellement d'un matériel) qui permet au contrôleur du portefeuille de générer, stocker, gérer et protéger des clés cryptographiques, des secrets et d'autres données privées sensibles. [21]

6.1.2.4. Document DID

Un document DID fait référence à chaque DID ou il contient les informations lors de la création du DID, son utilisation et la clé publique associé au DID et la signature numérique. Le document DID est universellement accessible et interopérable entre les divers plateformes et systèmes et Il peut être mis à jour par le titulaire.

6.1.2.5. DLT

Sont les registres distribués permettant l'enregistrement, la mise à jour et la vérification grâce au smart contrats exemple de Ethereum, HyperLedger Indy... etc.

7. Conclusion

L'évolution de l'identité numérique illustre un parcours significatif, depuis les systèmes centralisés traditionnels reposant sur des autorités de confiance jusqu'aux modèles contemporains plus ouverts et sécurisés. Comme nous l'avons démontré dans ce chapitre, l'identité décentralisée émerge comme une solution particulièrement prometteuse, offrant aux utilisateurs une souveraineté inédite sur leurs données personnelles tout en renforçant considérablement les aspects de sécurité et de transparence.

Cependant, la transition vers ce paradigme décentralisé n'est pas sans obstacles. Des défis majeurs persistent, notamment en ce qui concerne l'adoption à grande échelle, l'interopérabilité entre différents systèmes, et l'établissement d'un cadre réglementaire adapté. La technologie Blockchain se révèle être un catalyseur essentiel pour surmonter ces difficultés.

Le chapitre suivant approfondira le fonctionnement intrinsèque de la Blockchain et analysera en détail sa relation avec l'identité décentralisée. Nous explorerons également d'autres innovations technologiques émergentes dans ce domaine, offrant ainsi une perspective complète sur l'avenir de la gestion de l'identité numérique.

Chapitre II Intégration de la blockchain aux systèmes d'identité auto-souveraine (SSI)

1. Introduction

Depuis l'émergence du Bitcoin, la technologie blockchain a profondément transformé les mécanismes de sécurisation et de validation des transactions numériques. En tant que registre distribué, immuable et transparent, elle permet de se passer d'intermédiaires tout en renforçant la confiance entre les parties.

Bien au-delà du domaine des cryptomonnaies, son potentiel s'étend à de nombreux secteurs, en particulier celui de la gestion des identités. L'intégration de la blockchain dans ce domaine ouvre la voie à des identités numériques infalsifiables, résistantes à la fraude et entièrement sous le contrôle des utilisateurs.

Ce chapitre explore les principes fondamentaux de la blockchain, ses composants clés, ainsi que ses avantages et ses limites. Nous verrons également comment cette technologie s'applique aux systèmes d'identité décentralisée.

2. Définition de la Blockchain

La blockchain est une technologie décentralisée de stockage et d'échange d'informations reposant sur un réseau peer-to-peer. Elle permet d'établir un consensus entre les participants sans autorité centrale, grâce à des mécanismes cryptographiques et des règles de validation distribuées. Les données enregistrées sont immuables, transparentes et accessibles à tous les nœuds du réseau, ce qui garantit la fiabilité et la résilience du système [24] [25].

3. Comment fonctionne la Blockchain ?

La blockchain est une forme de technologie DLT (*Distributed Ledger Technology*), c'est-à-dire un registre numérique distribué et partagé entre plusieurs utilisateurs, sans qu'aucune autorité centrale n'en ait le contrôle. La blockchain permet de gérer des biens numériques (tels que la monnaie, les jetons, ou les certificats) de manière sécurisée et décentralisée.

La blockchain repose essentiellement sur les principes suivants :

- Un réseau peer-to-peer (pair à pair), où chaque participant partage une partie de ses ressources pour faire fonctionner le système.
- Les ordinateurs (appelés nœuds) communiquent via Internet pour maintenir le registre à jour.
- Le but est de garantir la propriété d'un bien numérique sans faire appel à un intermédiaire (comme une banque ou un service centralisé).

Les étapes clés pour qu'une blockchain fonctionne :

Pour qu'un tel système décentralisé fonctionne correctement dans un environnement ouvert (où personne ne se fait forcément confiance), plusieurs tâches doivent être assurées [22] :

1. Décrire la propriété : préciser à qui appartient quoi.
2. Protéger la propriété : empêcher les accès non autorisés ou les tentatives de vol.
3. Stocker les transactions : enregistrer chaque échange ou modification dans le registre.
4. Préparer le registre : structurer les données de manière à pouvoir les partager de façon sécurisée.
5. Partager le registre : distribuer les copies à tous les participants du réseau.
6. Ajouter de nouvelles transactions : insérer les nouvelles opérations dans le registre commun, après vérification.
7. Déterminer la bonne version : en cas de conflit, identifier la version la plus fiable du registre.

Remarque :

Il convient de noter que, plutôt que d'enregistrer un fichier entier dans la blockchain ce qui serait coûteux en termes d'espace et de ressources seule son empreinte numérique unique (appelée hash cryptographique) est stockée sur la blockchain. IPFS (*InterPlanetary File System*) est une technologie de stockage décentralisé qui permet de partager et conserver des fichiers de manière sécurisée, sans avoir recours à un serveur central.

Ainsi, la blockchain joue le rôle de preuve d'existence et d'intégrité du fichier (elle garantit qu'il n'a pas été modifié), tandis que le contenu réel est hébergé dans IPFS. Ce fonctionnement hybride permet de bénéficier d'un système plus léger, plus rapide, tout en conservant un haut niveau de sécurité et de traçabilité.

4. Caractéristiques de la Blockchain

Afin de mieux comprendre les fondements de la technologie blockchain, nous mettons en lumière les caractéristiques essentielles qui la structurent.

- Architecture distribuée : Les données de la blockchain sont stockées simultanément par les différents utilisateurs (nœuds) sur le réseau blockchain en même temps. Si un nœud est défectueux ou a perdu ses données, les autres nœuds du réseau ont toujours la copie de la blockchain et continuent de la mettre à jour. Le nœud affecté peut recopier la blockchain à partir des autres nœuds. [23]

- Intégrité et sécurité des données : La blockchain est tamper-proof dans le sens où lorsque des données d'un bloc sont modifiées, le changement est détecté en raison de la modification de l'empreinte (hachage) du bloc qui différera du hachage précédemment stocké dans le bloc suivant. [23]

- Transparence et traçabilité : Étant donné que les enregistrements de la blockchain sont horodatés et stockés sur tous les nœuds complets du réseau, toutes les activités et transactions peuvent être vérifiées et vues par tout le monde sur le réseau. Si l'adresse d'un nœud est connue, toutes ses activités et transactions peuvent être tracées. [23]

- Décentralisation sans autorités centrales et intermédiaires, La blockchain permet aux systèmes d'être autonomes et libérés des risques des intermédiaires et des autorités centrales. Cependant, les blockchains privées peuvent être partiellement ou totalement centralisées mais bénéficient tout de même des autres caractéristiques de la blockchain. [23]

- Économie de coûts L'utilisation de la blockchain permet de réaliser d'énormes économies, car les coûts associés aux systèmes intermédiaires sont économisés. C'est l'une des raisons pour lesquelles certaines banques et entreprises souhaitent intégrer la blockchain dans leurs systèmes afin de réduire les coûts [23]

- Interopérabilité La blockchain fournit une plate-forme de partage de données sécurisée qui permet à des parties distinctes de partager les mêmes données et de synchroniser leurs services. [23]

- Vérifiabilité En raison de la signature numérique de la blockchain, l'authenticité d'un enregistrement peut être vérifiée. Cela peut être difficile à réaliser dans d'autres bases de données, car cela nécessite des mécanismes cryptographiques tels que la signature numérique utilisée dans la blockchain. [23]

La figure II.1 illustre l'architecture de la blockchain reposant sur des mécanismes cryptographiques.
L'intégration du hachage du bloc précédent dans chaque en-tête assure l'immutabilité de la chaîne, empêchant toute modification rétroactive des données sans compromettre l'intégrité globale.
Par ailleurs, l'utilisation de l'arbre de Merkle permet une vérification efficace de l'intégrité des transactions, en offrant des preuves d'inclusion rapides et peu coûteuses en ressources.

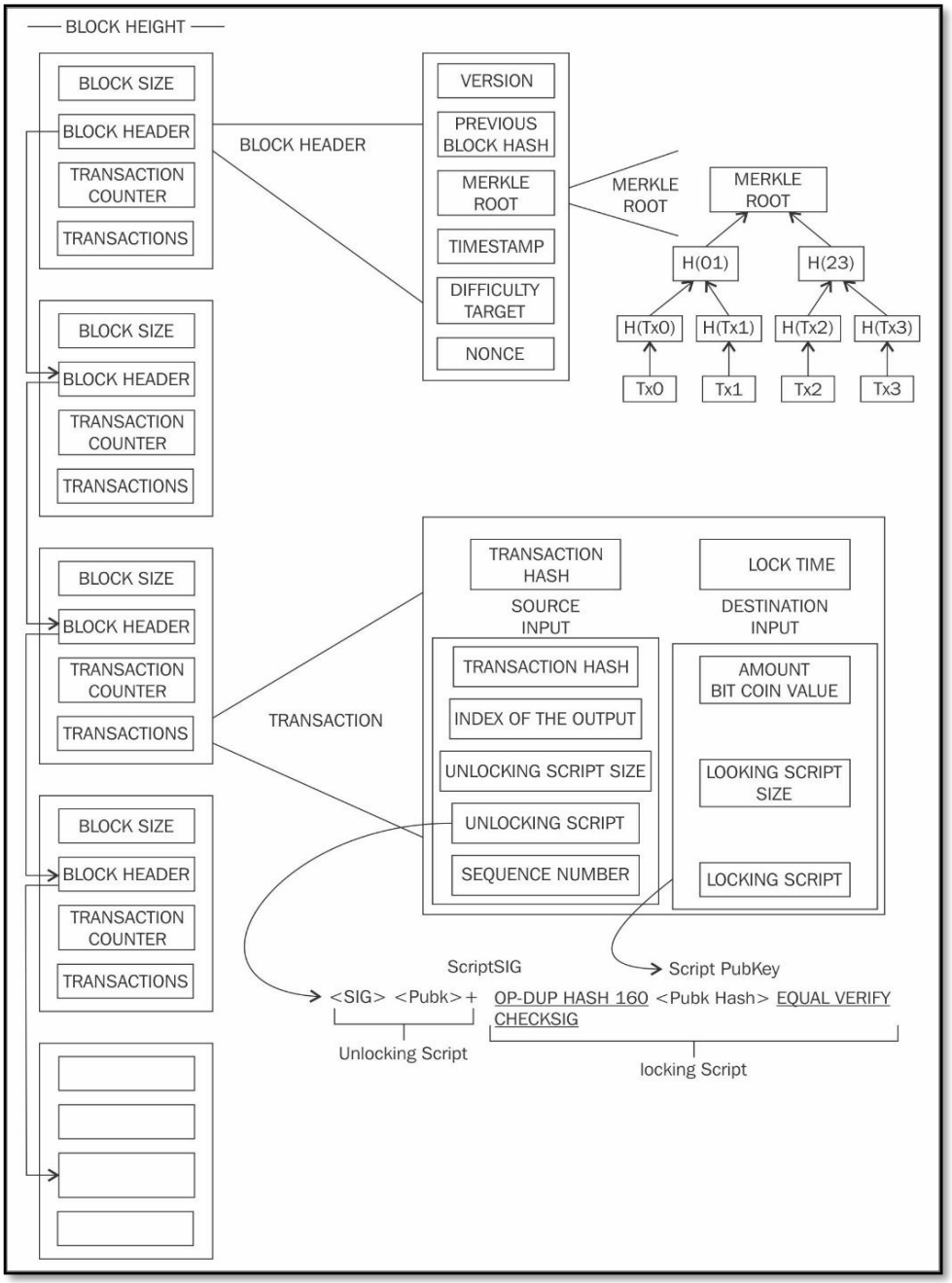

Figure II-1 Structure de la Blockchain [24]

5. Les types de blockchain

5.1. Blockchains privées : L'accès aux blockchains privées se fait sur invitation. Les utilisateurs peuvent être validés par une règle définie par l'administrateur central du réseau. Seuls certains utilisateurs peuvent avoir accès au réseau et parmi ceux-ci, il existe des limites aux types de transactions qu'ils peuvent effectuer. Ces types de blockchains utilisent une approche consensuelle de preuve d'autorité (PoA). PoA fait partie d'un algorithme de consensus qui donne à un petit groupe d'utilisateurs sur une blockchain l'autorité de valider les transactions sur le réseau. Le PoA est principalement utilisé dans des environnements commerciaux sécurisés pour contrôler l'accès et la tenue des registres, car les données de transaction ne sont généralement pas visibles pour eux. [25]
Même si la blockchain privée est moins sécurisée et centralisée, elle est plus évolutive et n'a pas d'attaque de 51 %, de confidentialité, et SelfishMining. Multichain et Blockstack sont des exemples de blockchains privées [23]

5.2. Blockchains publiques : Les blockchains publiques sont une question d'engagement et de transparence. Le consensus des transactions n'est pas centralisé, de sorte que n'importe qui peut participer à l'authentification des transactions réseau et le code du logiciel est open source. Cela signifie qu'il est disponible pour le public. Les blockchains publiques sont décentralisées par le biais de la crypto-économie afin de promouvoir la coopération au sein d'un réseau. Il n'y a pas d'entité unique contrôlant le réseau et il n'y a pas de point de défaillance central. Les algorithmes de consensus utilisés dans une blockchain publique sont le PoW et le Proof of Stake (PoS). Le PoS est un algorithme qui traite les transactions et crée de nouveaux blocs dans une chaîne. Il est utilisé pour valider toutes les entrées de données et assurer la sécurité de la base de données. Personne ne parle jamais des blockchains Consortium car elles ne sont pas largement utilisées. Elles peuvent être décrites comme une combinaison de blockchains publiques et privées, car elles ont une approche « semi-autorisée ». [25]

5.3. Blockchains de consortium : les participants sont généralement connus et ont été autorisés par une autorité centrale à participer au consensus. Ce type de blockchain est semi-décentralisé tout en offrant un certain niveau de contrôle. Les données de transaction sont

également gardées privées. Les blockchains de consortium utilisent les trois algorithmes de consensus ; PoW, PoS et PoA. [25]

5.4. Blockchain à permission : Les réseaux blockchain a permission sont ceux où les utilisateurs publiant des blocs doivent être autorisés par une autorité (qu'elle soit centralisée ou décentralisée). Étant donné que seuls les utilisateurs autorisés gèrent la blockchain, il est possible de restreindre l'accès en lecture et de restreindre qui peut émettre des transactions. Les réseaux blockchain a permission peuvent donc permettre à quiconque de lire la blockchain ou ils peuvent restreindre l'accès en lecture aux personnes autorisées. Ils peuvent également permettre à n'importe qui de soumettre des transactions à inclure dans la blockchain ou, encore une fois, ils peuvent restreindre cet accès aux seules personnes autorisées. Les réseaux blockchain autorisés peuvent être instancié et maintenu à l'aide d'un logiciel open source ou closed Source [23]

Le tableau ci-dessous présente les différentes caractéristiques des types de blockchain, ou on présente leur participation, membres, sécurité, centralisation, évolutivité, efficacité(en terme de la rapidité, le coût et les performances globales du traitement des transactions) et la consommation d'énergie

Blockchain	Participation	Membres	Sécurité	Centralisation	Scalabilité	Efficacité	Consommation d'énergie	Exemples
Publique	Sans permission	Inconnu	Meilleure	Décentralisée	Faible	Faible	Dépend du consensus	Bitcoin, Ethereum
Privée	Avec permission	Connu	Bonne	Centralisée	Élevée	Plus élevée	Très faible	Blockstack, Multichain
Consortium	Avec permission	Connu	Meilleure	Partielle	Modérée	Élevée	Très faible	Hyperledger, Corda

Tableau II-1 Comparaison des types de blockchain [26]

6. Les composants de la blockchain

La technologie blockchain repose sur une architecture constituée d'éléments fondamentaux interconnectés qui assurent collectivement son intégrité et sa performance. Ces composants essentiels, notamment les blocs, les nœuds du réseau et les systèmes cryptographiques, fonctionnent en parfaite coordination pour garantir la fiabilité et l'efficacité de l'ensemble du système. L'analyse détaillée de ces éléments constitutifs permettra d'éclairer leurs interactions et leur contribution au fonctionnement global de cette technologie révolutionnaire.

6.1. Les nœuds

Un nœud de blockchain est un appareil qui exécute le protocole d'une blockchain et se connecte à son réseau. Les nœuds sont des modérateurs qui forment l'infrastructure du réseau décentralisé, agissant comme parties prenantes de la blockchain elle-même. Leur fonction principale est de maintenir le consensus sur le registre public, avec des responsabilités variant selon la catégorie de nœud. Virtuellement tout appareil disposant d'une adresse IP peut fonctionner comme nœud, qu'il s'agisse de routeurs, modems, commutateurs, concentrateurs, serveurs ou autres. [26]

6.1.1. Pourquoi les nœuds sont-ils nécessaires ?

Une blockchain n'a pas d'autorité centrale, de sorte que le contrôle du réseau est démocratisé par un ensemble de nœuds sélectionnés. Ces nœuds collaborent pour implémenter des mécanismes essentiels comme l'authentification des transactions et l'exécution des protocoles décisionnels.

La distribution de l'autorité entre de multiples nœuds incarne le principe fondamental de décentralisation. Plus une blockchain compte de nœuds, plus son degré de décentralisation est élevé. Un nombre important de nœuds garantit la résilience du réseau en consolidant les systèmes majoritaires tout en complexifiant considérablement toute tentative d'infiltration malveillante. [26]

6.1.2. Types de nœuds

Les nœuds d'une blockchain sont catégorisés en fonction des rôles spécifiques qu'ils assurent au sein du réseau. Bien que leurs fonctions diffèrent, tous participent à la préservation de l'intégrité et de la sécurité du système. Un protocole blockchain repose généralement sur une variété de nœuds, chacun jouant un rôle complémentaire dans l'écosystème global [28].

On distingue principalement deux types de nœuds : noeuds complets et nœuds légers.

➢ Nœuds complets (*full nodes*)

En termes simples, un nœud complet stocke une copie complète du registre numérique du réseau. Ces types de nœuds crypto sont le fondement de la plupart des blockchains : ils stockent l'historique de la chaîne et communiquent avec d'autres nœuds complets. Ce sont également ces nœuds qui sont chargés de fournir une copie de la chaîne aux nouveaux nœuds. Ils constituent la colonne vertébrale d'une blockchain et c'est pourquoi la plupart des blockchains sont dotées de ce type de nœuds. [27]

- Nœuds de minage (*miner nodes*)

  Les nœuds de minage existent sur les blockchains utilisant la PoW. Ce sont les participants chargés de vérifier les transactions et de les ajouter à la blockchain. Le minage nécessite une grande puissance de calcul pour résoudre des puzzles complexes. C'est pourquoi, en contrepartie de leur travail, les mineurs reçoivent des récompenses sous forme de cryptomonnaies. [27]

- Nœuds de validation (*validator nodes*)

  Les nœuds de validation sont similaires aux nœuds de minage, mais sur un réseau de PoS. Ils valident également les transactions et créent des blocs. Cependant, ils n'ont pas à résoudre des problèmes de calcul complexes. Ils sont choisis en fonction du montant des cryptos qu'ils immobilisent dans le système. Comme les mineurs, ils sont récompensés pour les blocs qu'ils créent. [27]

Les nœuds de minage et les nœuds de validation constituent des variantes de nœuds complets.

➢ Nœuds légers (light nodes)

Les nœuds légers, sont un peu différents. Tout d'abord, ils ne stockent pas l'intégralité de la blockchain comme le font les nœuds complets. Ils se contentent de télécharger les « en-têtes de bloc ». Sans trop entrer dans les détails techniques, cela signifie qu'ils n'ont pas besoin d'une grande capacité de stockage. Ces nœuds crypto ont pour seule tâche de vérifier les transactions sur la blockchain à l'aide d'un système SVP (vérification simplifiée des paiements). Notons toutefois que toutes les blockchains n'utilisent pas ce type de nœuds. Ils sont principalement utilisés dans les blockchains dont l'espace de stockage par bloc est limité, comme Bitcoin. [27]

6.2. Les blocs

Un bloc est une structure de données (enregistrement) qui agit comme un conteneur qui agrège les transactions à inclure dans la blockchain. Le bloc est composé de deux parties principales :

- Un en-tête qui contient des métadonnées telles que le hachage du bloc précédent, l'horodatage, un nonce ainsi que la racine de l'arbre de Merkle regroupant les transactions.
- Le corps du bloc qui contient une liste de transactions qui constitue la partie majeure de la taille du bloc.

Par exemple, dans Bitcoin, la taille de l'en-tête du bloc est de 80 octets, tandis que la transaction moyenne est d'au moins 250 octets et que le bloc moyen contient plus de 500 transactions. Un bloc complet, avec toutes les transactions, est donc 1 000 fois plus grand que l'en-tête du bloc. [28]

La figure II.3 illustre la structure d'un bloc dans la Blockchain dont le premier bloc est appelé « *Genesis Block* » et chaque bloc est lié au bloc précédent via la valeur d'un hachage.

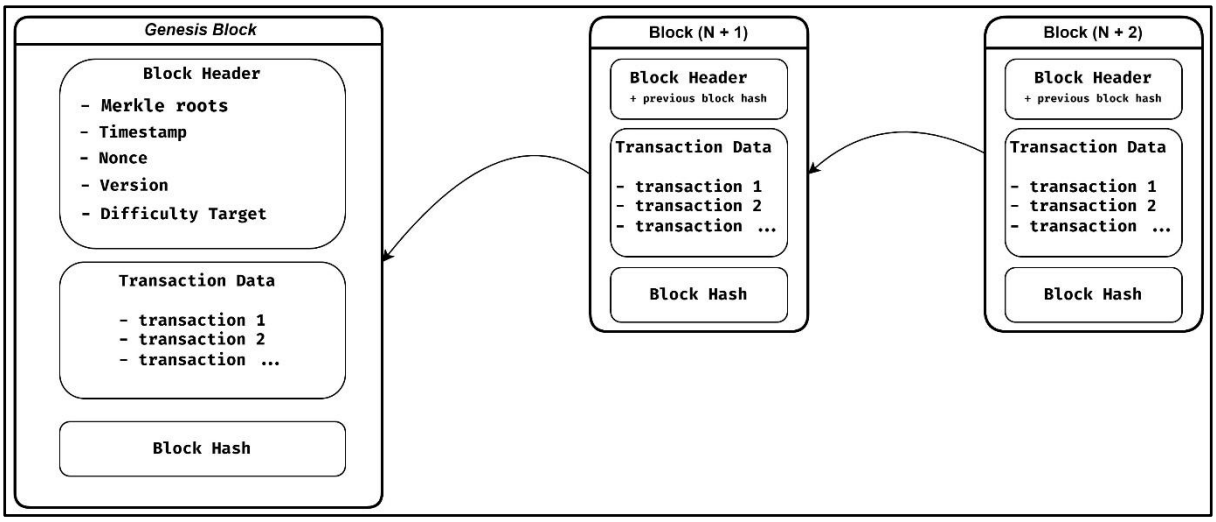

Figure II-2 Structure d'un Bloc

6.2.1. Arbre de Merkle (Merkle Tree)

Un arbre de *Merkle* est une structure arborescente obtenue en hachant les transactions d'un bloc, puis en regroupant ces hachages deux à deux pour les hacher de nouveau, et ce de manière récursive jusqu'à l'obtention d'un unique hachage : la racine de Merkle (*Merkle root*).

Cette structure permet une vérification efficace de l'intégrité et de l'inclusion des transactions, sans nécessiter le traitement de l'ensemble du bloc. Certaines blockchains exploitent cette propriété pour comparer rapidement les racines de Merkle générées par différents nœuds et identifier les divergences éventuelles.

La racine de Merkle constitue ainsi un mécanisme cryptographique simple mais puissant pour garantir l'authenticité et l'intégrité des données, en particulier dans les systèmes distribués. [29]

La figure ci-dessous illustre un arbre de Merkle dans lequel chaque nœud interne est le hash des nœuds enfants, et la racine (Merkle Root) permet de prouver l'existence de transactions (T1-T10) de manière efficace et sécurisée. [28, p. 374]

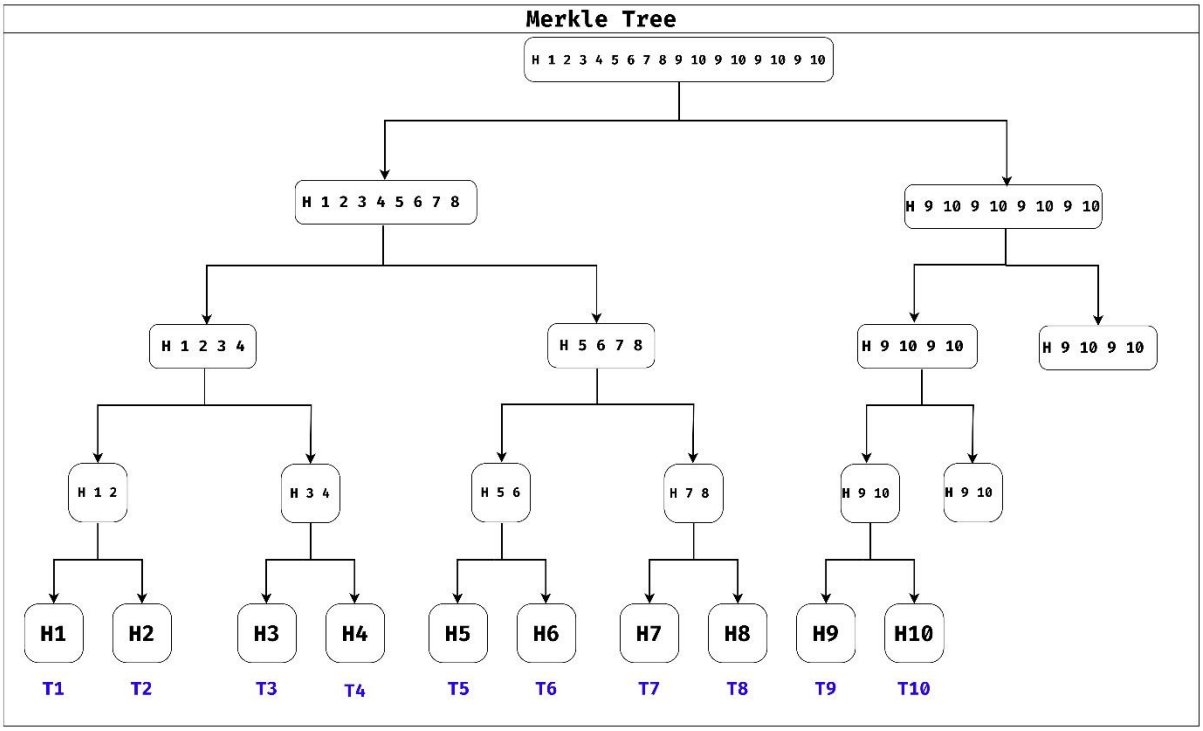

Figure II-3 Arbre de Merkle

6.3. Les transactions

Les transactions constituent l'unité fondamentale d'activité dans un système blockchain, représentant chaque échange ou modification d'état enregistrée dans la chaîne. Cette section explore en détail les transactions blockchain, les techniques cryptographiques qui les sécurisent, et leur cycle de vie complet.

Une transaction est un ensemble de données qui représente un transfert de valeur ou d'information entre des participants du réseau. Typiquement, une transaction contient généralement les éléments suivants :

- Identifiant de transaction (TXID) : une empreinte unique générée à partir de hachage cryptographique du contenu de la transaction. Il permet de distinguer chaque transaction dans la blockchain et de la référencer de manière immuable.

- Entrées (Inputs) : Les références aux transactions précédentes dont les sorties n'ont pas encore été dépensées (UTXOs – Unspent Transaction Outputs). Elles indiquent la source des fonds ou des données à transférer.
- Sorties (Outputs) : Les adresses de destination associées aux informations à transférer. Chaque sortie spécifie un destinataire, et devient une entrée potentielle pour une transaction future.
- Horodatage : L'horodatage indiquant quand la transaction a été créée ou diffusée.
- Signature(s) numérique(s) : Elles garantissent l'authenticité de la transaction, l'intégrité des données et l'autorisation du propriétaire des fonds.

6.3.1. Le cycle de vie d'une transaction

Une transaction dans un environnement blockchain suit généralement les étapes suivantes :

a) Création : L'utilisateur initie une transaction en spécifiant les entrées, les sorties et le montant ou les informations à transférer.

b) Signature : La transaction est signée numériquement à l'aide de la clé privée de l'émetteur pour garantir son authenticité.

c) Diffusion : La transaction est diffusée dans le réseau peer-to-peer aux nœuds participants.

d) Vérification : Les nœuds valident la transaction en contrôlant la signature, les fonds disponibles et les règles du protocole.

e) Mise en pool : Une fois validée, la transaction est placée dans le mempool en attente d'inclusion dans un bloc.

f) Inclusion dans un bloc : Un mineur sélectionne la transaction et l'intègre dans un nouveau bloc lors du processus de minage.

g) Confirmation : Le bloc contenant la transaction est ajouté à la blockchain, et chaque bloc ultérieur renforce la confirmation.

6.3.2. Techniques cryptographiques

- Fonction de hachage

Le hachage est une autre méthode importante. Le hachage de vos données est similaire à l'établissement d'une empreinte numérique de celles-ci. Il convertit vos données en une chaîne unique de caractères, semblable à un code secret qui symbolise vos données. Si même une petite partie de vos données change, l'empreinte digitale entière sera modifiée. Ceci est utile pour maintenir l'intégrité des données et s'assurer qu'elles n'ont pas été altérées. [1]

Le hachage est un processus cryptographique qui consiste à transformer une entrée (données de toute taille) en une sortie de longueur fixe, appelée haché, empreinte ou digest. Cette transformation est réalisée à l'aide d'une fonction de hachage qui applique un algorithme déterministe, produisant toujours la même sortie pour la même entrée donnée, [1]

Une fonction de hachage cryptographique est une fonction de hachage **H : {0,1} * → {0,1}** ^{n} qui vérifie les trois conditions à la résistante à la pré-image, la résistante à la seconde pré-image et la résistante à la collision.

Les principales propriétés d'une fonction de hachage cryptographique sécurisée sont :

- L'unidirectionnalité : Il est pratiquement impossible de retrouver l'entrée originale à partir de son empreinte de hachage, En d'autres termes étant donné une empreinte *y* Alors, il est difficile de trouver un x tel que $H(x) = y$.
- Déterminisme : La même entrée produit toujours le même hachage,
- Résistance aux collisions : Il est difficile de trouver deux entrées différentes produisant le même hachage, étant donné un x et son haché y (y = h(x)), alors, il est difficile de trouver un autre $x' \neq x$ tel que $h(x') = y$,
- Effet d'avalanche : Une modification mineure de l'entrée entraîne un changement complet de l'empreinte de hachage.

Parmi les exemples d'algorithmes de hachage : (Keccak-256, BLALE3, Bcrypt et SHA-256 ...etc.)

- Clés publiques et privées

Considérez la clé publique comme un verrou unique que tout le monde peut voir et utiliser pour vous fournir des communications cryptées ou des informations d'identification vérifiées. Le partage de votre clé publique avec n'importe qui est sécurisé car il lui permet simplement de vous envoyer des informations et non d'accéder à vos données personnelles. La clé privée, en revanche, est comme une clé secrète que vous seul avez. Il est utilisé pour déchiffrer les communications ou les informations d'identification qui vous sont envoyées à l'aide de votre clé publique. Votre clé privée est incroyablement précieuse car elle permet à toute personne qui la possède d'accéder à vos informations personnelles. Il est essentiel de garder votre clé privée en sécurité. [1]

- Signature numérique

Supposons que vous deviez signer un document essentiel, tel qu'un contrat. C'est possible dans le monde numérique SSI avec une signature numérique. C'est comme signer votre nom, mais en utilisant une méthode infalsifiable. Voici comment cela fonctionne : vous générez une signature numérique pour une donnée à l'aide de votre clé privée. Cette signature est unique en son genre et garantit que les données sont authentiques et n'ont pas été modifiées. Lorsque vous partagez ce contenu signé avec quelqu'un, il peut vérifier la signature à l'aide de votre clé publique. Si tout se passe bien, ils sauront que les données proviennent de vous et qu'elles n'ont pas été falsifiées. En SSI, les signatures numériques sont équivalentes à des sceaux mystiques qui garantissent la légitimité des informations personnelles. Ils donnent confiance et protection dans un monde en ligne plein de dangers. [1]

Parmi les algorithmes de signatures numérique (FALCON, ECDSA, EdDSA ...etc.) Ces algorithmes sont utilisés dans de nombreuses applications, comme les certificats SSL/TLS, les signatures de fichiers, les cryptomonnaies (Bitcoin utilise ECDSA), etc.

La signature numérique repose sur les algorithmes asymétriques ou la clé privé est utilisée pour la signature tandis que la clé publique est utilisée pour la vérification de la signature ci-dessous le schéma qui illustre la signature numérique :

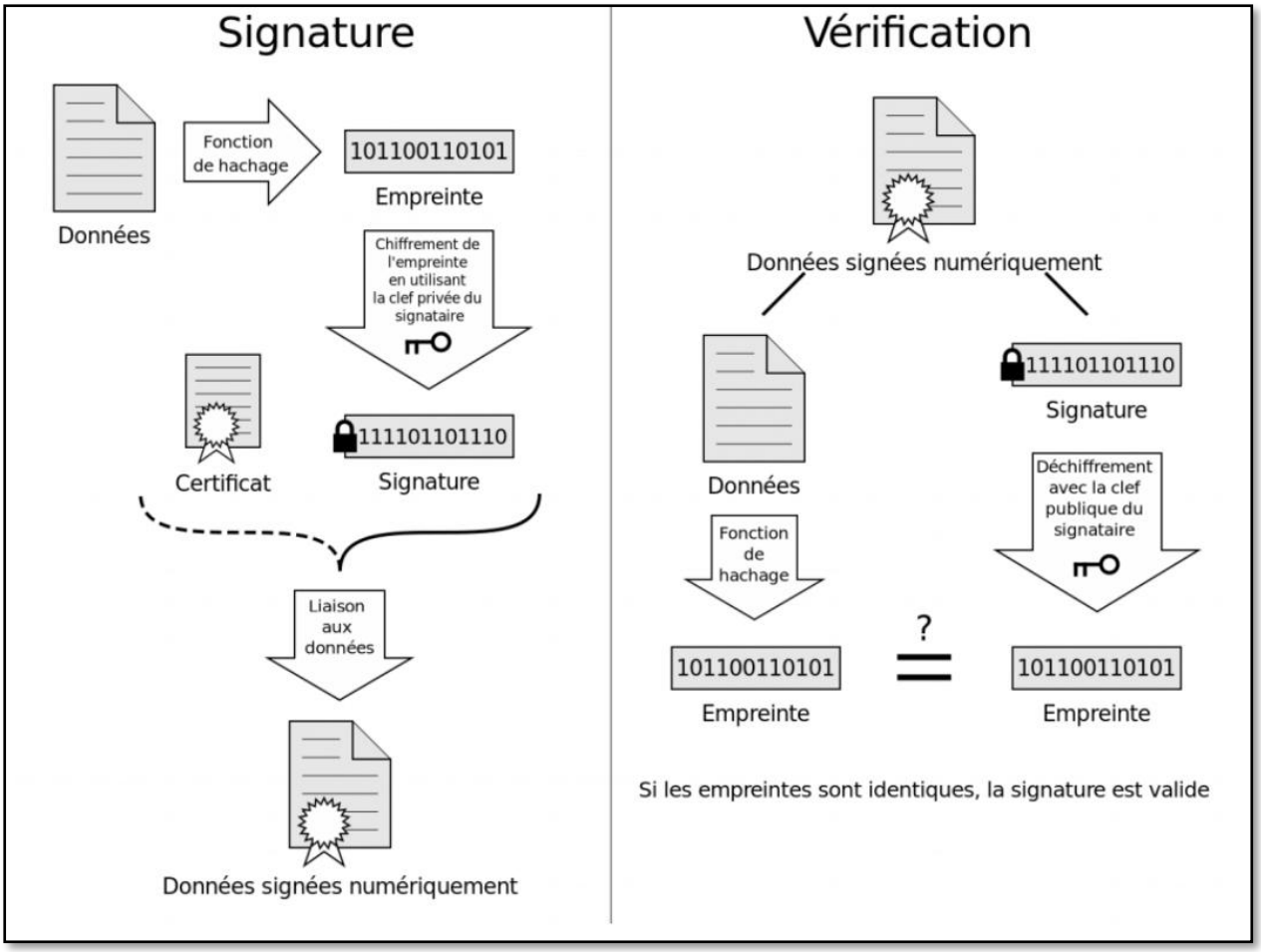

Figure II-4. Schéma générale de la signature numérique [30]

Cette illustration présente le mécanisme de création et de vérification d'une signature numérique, un élément crucial pour garantir l'authenticité et l'intégrité des informations. Tout commence par la conversion des données en une empreinte unique, obtenue grâce à une fonction de hachage, représentée sous forme binaire. Cette empreinte est ensuite chiffrée à l'aide de la clé privée du signataire, générant ainsi la signature numérique. Par la suite, cette signature est associée aux données originales afin de faciliter leur vérification. Pour confirmer la validité de la signature, le destinataire déchiffre celle-ci en utilisant la clé publique du signataire, ce qui lui permet de récupérer l'empreinte d'origine. Le destinataire procède également au calcul de l'empreinte des données qu'il a reçues, puis compare les deux empreintes. Si elles correspondent, cela signifie que la signature est valide, indiquant ainsi que les données n'ont subi aucune modification et proviennent effectivement du signataire. Ce mécanisme repose sur l'utilisation de techniques de hachage et de chiffrement asymétrique pour garantir la sécurité et l'authenticité des données.

➢ Courbe-elliptique Edward

Les courbes elliptiques représentent une approche moderne et efficace en cryptographie asymétrique, offrant un haut niveau de sécurité avec des clés de taille réduite. Parmi les différentes familles de courbes, les courbes de type Edwards se distinguent par leur efficacité et leur sécurité accrues lors des opérations cryptographiques.

Proposées initialement par Harold Edwards, ces courbes sont définies par une équation symétrique de la forme :

$$x^2+y^2=1+dx^2y^2$$

Où d est un paramètre non nul dans un corps fini. Une variante largement utilisée est Curve25519, sur laquelle repose le schéma de signature Ed25519, utilisé notamment dans l'algorithme EdDSA. Ce dernier est optimisé pour la performance, la compacité des signatures et la robustesse cryptographique.

Génération des clés cryptographiques

Dans ce contexte, chaque utilisateur génère une paire de clés asymétriques :

- La clé privée est un entier aléatoire de 256 bits, gardé strictement confidentiel.
- La clé publique est dérivée de cette clé privée par une multiplication scalaire d'un point de base G sur la courbe elliptique, selon la relation :

$$\text{PublicKey}=k\cdot G$$

Où K est la clé privée. Cette opération repose sur la difficulté du problème du logarithme discret elliptique, garantissant la sécurité du schéma.

La figure suivante montre différentes formes de courbes elliptiques de type Edwards, obtenues pour des valeurs variées du paramètre d

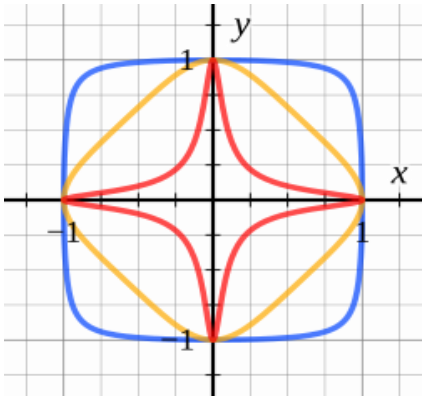

Figure II-5 Représentation graphique de courbes elliptiques Edwards dans le plan (x, y)

Comme indiqué auparavant la blockchain est une base de données décentralisé permettant des transferts sécurisés d'actifs tels que de l'argent, des contrats et des biens sans intermédiaire. Pour effectuer une transaction sur la blockchain, un portefeuille (wallet) est nécessaire.
Un portefeuille interagit avec la blockchain et est protégé par la cryptographie et à l'aide d'une clé privée et publique. Une adresse publique (identifiant : la clé publique est haché) permet de recevoir des fonds, tandis que la clé privée est gardée secrète et permet de signer les transactions afin d'autoriser la dépense des fonds présents sur le wallet. [31]
Prenons l'exemple d'une transaction entre Alice et Bob
Supposons qu'Alice souhaite envoyer des bitcoins à Bob. Voici les étapes qu'ils doivent suivre pour réaliser une transaction :

- Adresses et clés : Alice et Bob possèdent tous deux une paire de clés (publique et privée) et une adresse publique bitcoin associée. L'adresse publique est une version encodée de la clé publique et est utilisée pour recevoir des bitcoins. La clé privée est gardée secrète et sert à autoriser les transactions.

- Création de la transaction : Alice crée une transaction dans laquelle elle indique l'adresse de Bob et le montant en bitcoins qu'elle souhaite lui envoyer. Elle ajoute également une petite somme (frais de transaction) pour inciter les mineurs à valider et inclure cette transaction dans un bloc.

- Signature de la transaction : Alice signe numériquement la transaction avec sa clé privée. Cette signature garantit que la transaction provient bien d'Alice et qu'elle a autorisé le transfert des bitcoins à Bob.

- Diffusion de la transaction : La transaction signée est diffusée sur le réseau bitcoin. Les nœuds du réseau (ordinateurs participant à la validation des transactions) vérifient la validité de la transaction et de la signature d'Alice.

- Confirmation de la transaction : Les mineurs sélectionnent les transactions non confirmées et tentent de résoudre le problème cryptographique pour créer un nouveau bloc contenant ces transactions. Une fois le problème résolu, le bloc est ajouté à la blockchain, ce qui confirme la transaction d'Alice à Bob.

- Réception des bitcoins : Après plusieurs confirmations (généralement 2 à 6 confirmations sont considérées comme suffisantes), Bob peut considérer les bitcoins comme reçus et dépensables. Une confirmation = 1 bloc, 6 confirmations = 6 blocs soit environ 60 minutes dans le cas d'une transaction sur le réseau Bitcoin. [31]

6.4. Le consensus

6.4.1. Qu'est-ce qu'un consensus ?

Le terme de consensus dans une blockchain représente le mécanisme par lequel l'ensemble des nœuds parvient à un accord sur l'état de la blockchain. [32]
Plus formellement, le consensus peut être défini comme un processus algorithmique permettant à un réseau de nœuds (ordinateurs distribués geographiquement) de s'accorder sur une version unique et cohérente de l'historique des transactions, même en présence d'acteurs potentiellement malveillants.

6.4.2. Propriétés du consensus

- Accord (Agreement) : Tous les nœuds honnêtes du réseau doivent parvenir à un accord sur l'état actuel de la blockchain, c'est-à-dire sur les transactions validées et l'ordre dans lequel elles sont ajoutées aux blocs. Cela garantit qu'il n'y a qu'une seule version de la vérité partagée par tous les participants.

- Validité (Validity) : Seules les transactions valides selon les règles du protocole peuvent être ajoutées à la blockchain. Le mécanisme de consensus doit s'assurer que les transactions invalides sont rejetées par le réseau.

- Finalité (Finality) : Une fois qu'un bloc est validé et ajouté à la chaîne, il doit être considéré comme définitif, c'est-à-dire irréversible.
  Cela permet d'éviter la double dépense et assure la confiance dans l'historique des transactions.

- Décentralisation : Le consensus doit être atteint sans autorité centrale, par la coopération et l'accord d'un grand nombre de nœuds répartis sur le réseau. Cela garantit que personne ne contrôle seul la validation des transactions.

- Sécurité : Le mécanisme de consensus protège la blockchain contre les attaques, notamment celles où un acteur malveillant tenterait de prendre le contrôle du réseau (par exemple en contrôlant plus de 51 % des nœuds). [33] [34]

6.4.3. Types de mécanismes de consensus

Plusieurs mécanismes de consensus ont été développés, chacun présentant des caractéristiques, avantages et inconvénients distincts. Voici les principaux :

➢ Proof of work (PoW)

Principe de fonctionnement:

Les mineurs se font concurrence pour créer de nouveaux blocs remplis de transactions traitées. Le gagnant partage le nouveau bloc avec le reste du réseau et gagne des ETH fraîchement frappés. La course est gagnée par l'ordinateur qui est capable de résoudre un puzzle mathématique le plus rapidement. Cela produit le lien cryptographique entre le bloc actuel et le bloc qui l'a précédé. Résoudre ce travail en « proof-of-work ». La chaîne canonique est ensuite déterminée par une règle de choix de fourche qui sélectionne l'ensemble des blocs qui ont fait l'objet du plus de travail pour les exploiter. [32]

Avantages et inconvénients :

Bien que le POW est très résistant aux attaques grâce au coût élevé du minage et décentralisé ou il est ouvert à tous les mineurs et offre une gouvernance sans autorité centrale mais il est Très énergivore à cause du minage intensif et lent en transaction avec des frais élevés.

- Proof of stake (PoS)

Principe de fonctionnement:

Le fonctionnement repose sur la sélection aléatoire, mais pondérée par la quantité de cryptomonnaies mises en jeu, des validateurs qui vont créer et valider les nouveaux blocs. Plus, un validateur possède de tokens mis en jeu, plus il a de chances d'être choisi pour forger un bloc et recevoir une récompense. En cas de mauvaise conduite ou d'inactivité, le validateur peut subir une pénalité financière (slashing). [35]

Avantages et inconvénients :

Contrairement au Proof of Work, le PoS ne nécessite pas de puissants calculs, ce qui réduit drastiquement la consommation d'énergie, Ainsi qu'il permet des validations plus rapides des transactions, rendant le réseau plus scalable et moins coûteux en termes de temps et de ressources. [36]

Pour pouvoir forger des blocs en PoS, il est nécessaire de mettre en jeu un minimum de jetons. Si ce minimum est trop élevé, en matière de prix notamment (ex : minimum 32 ETH= ~60,000$), seuls ceux qui peuvent se le permettre ou les pools (groupement de forgeurs) pourront construire des blocs. [37]

- Proof of autority (PoA)

Principe de fonctionnement :

Le modèle de preuve d'autorité repose sur un nombre limité de validateurs de blocs et c'est ce qui en fait un système hautement évolutif. Les blocs et les transactions sont vérifiés par des participants préapprouvés, qui agissent en tant que modérateurs du système.

Le PoA exploite la valeur des identités, ce qui signifie que les validateurs de blocs ne jalonnent pas des pièces, mais leur propre réputation. Par conséquent, les blockchains PoA sont sécurisées par les nœuds de validation qui sont arbitrairement sélectionnés comme entités dignes de confiance. [38]

Avantages et inconvénients :

Contrairement à PoW, où l'anonymat peut protéger les utilisateurs malveillants, les validateurs PoA sont connus et donc responsables de leurs actions. Ainsi que Le PoA ne nécessite pas de validateurs pour résoudre des problèmes mathématiques complexes, ce qui signifie qu'il consomme beaucoup moins d'énergie que les systèmes PoW. [39]

Une critique courante est que l'identité des validateurs PoA est visible par tout le monde. L'argument contre cela est que seuls les acteurs établis capables d'occuper ce poste chercheraient à devenir un validateur (en tant que participant connu du public). Néanmoins, la connaissance de l'identité des validateurs pourrait potentiellement conduire à une manipulation par un tiers. [38]

Si la PoS de et la PoW dominent les discussions sur les consensus, une multitude d'autres Protocoles sont disponibles tels que Proof of History et Delegated Proof of Stake... etc. chacune ayant ses avantages et ses inconvénients. Ci-dessous un tableau qui englobe chacune avec ses avantages et ses inconvénients :

Algorithme	Efficacité énergétique	Sécurité	Scalabilité	Décentralisation	Finalité
Preuve de travail (Proof of Work)	Faible	Élevée	Faible	Élevée	Probabiliste
Preuve d'enjeu (Proof of Stake)	Élevée	Élevée	Élevée	Moyenne	Probabiliste
Preuve d'historique (Proof of History)	Élevée	Élevée	Élevée	Élevée	Probabiliste
Preuve d'enjeu déléguée (Delegated PoS)	Élevée	Élevée	Élevée	Moyenne	Déterministe
Preuve d'autorité (Proof of Authority)	Élevée	Élevée	Élevée	Faible	Déterministe

Tableau II-2 Comparaison des Algorithmes de Consensus [40]

6.5. Le minage

6.5.1. Définition

Le minage est le mécanisme du Proof of Work qui permet de valider les transactions, créer les blocs et sécuriser la blockchain. Des participants, appelés mineurs, utilisent leur puissance de calcul pour résoudre un problème cryptographique complexe, ajoutant ainsi de nouveaux blocs après un consensus réseau. Ce processus est crucial pour la sécurité (résistance aux attaques par la preuve de travail coûteuse) et la décentralisation du réseau. Les mineurs sont incités, par des récompenses sous forme de tokens (jetons numériques), à maintenir la blockchain et garantir son intégrité. [41]

6.5.2. Fonctionnement du minage

Le processus de minage comprend plusieurs étapes clés exécutées par les mineurs :

- Validation des transactions : Les mineurs collectent et vérifient les transactions en attente (signatures, etc.) pour les inclure dans un bloc candidat.
- Preuve de travail : Les mineurs calculent de nombreux hachages en variant une partie de l'en-tête du bloc (le "nonce") jusqu'à trouver une valeur de hachage qui respecte les critères de difficulté définis par le réseau (par exemple, commence par un certain nombre de zéros). Cela demande une puissance de calcul intensive.

- Diffusion et ajout de nouveaux blocs : Le premier mineur arrivant à résoudre la preuve de travail diffuse le bloc "prouvé" au réseau. Les autres nœuds vérifient la validité du bloc et de la preuve de travail. S'il est accepté par la majorité (consensus), il est ajouté à la blockchain.
- Récompense : En échange du travail effectué, le mineur qui réussit à ajouter un nouveau bloc reçoit une récompense sous forme d'une certaine unité de cryptomonnaie (par exemple : 3,125 bitcoins par bloc pour Bitcoin). [41] [42] [43] [44]

6.5.3. Qu'est-ce que la difficulté de minage ?

La difficulté de minage est une unité de mesure lors de minage d'une cryptomonnaie telle que le Bitcoin qui se réfère à la difficulté et au temps nécessaires pour trouver la bonne empreinte (hash) pour chaque bloc. Cette difficulté permet de mesurer le niveau de complexité du calcul cryptographique réalisé dans le processus de minage.

Cette difficulté est ajustée périodiquement afin de maintenir constant le temps moyen nécessaire à la création d'un bloc (appelé *target block time*, fixé à environ 10 minutes pour le Bitcoin). [41]

6.6. Les contrats intelligents (Smart contracts)

Un contrat intelligent est un programme auto-exécutable dont les clauses contractuelles sont inscrites directement dans un code. Stocké sur une blockchain, le contrat s'exécute automatiquement lorsque des conditions prédéfinies sont remplies. Conçu initialement par Nick Szabo, ce concept a trouvé sa pleine réalisation technologique avec l'émergence des blockchains (souvent associé à la "Blockchain 2.0"). Le code du contrat est déployé sur la blockchain, associé à une adresse unique. L'exécution est déclenchée par une transaction envoyée à cette adresse. Le bon déroulement du contrat et son exécution fiable sont assurés par le protocole de consensus du réseau. [41, p. 52].

Les contrats intelligents apportent des bénéfices significatifs, notamment l'automatisation des accords, la réduction des intermédiaires (et donc des coûts et délais), une transparence accrue et une fiabilité basée sur l'exécution décentralisée et immuable.

7. Avantages de la Blockchain

La blockchain offre nombreux avantages qui en font une solution fiable basée sur la décentralisation des systèmes et jouant un rôle clé dans l'avenir de la sécurité.

- Ouvert : L'un des principaux avantages de la technologie blockchain est qu'elle est accessible à tous, ce qui signifie que tout le monde peut devenir un participant à la contribution à la technologie blockchain, on n'a besoin d'aucune autorisation de quiconque pour rejoindre le réseau distribué. [45]
- Vérifiable : La technologie blockchain est utilisée pour stocker des informations de manière décentralisée afin que tout le monde puisse vérifier l'intégrité des informations. [45]
- Permanent : Les enregistrements ou les informations stockés sont permanents, ce qui signifie qu'il ne faut pas s'inquiéter de perdre les données car des copies en double sont stockées à chaque nœud local car il s'agit d'un réseau décentralisé qui possède plusieurs nœuds dignes de confiance. [45]
- Libre de censure : La technologie blockchain est considérée comme exempte de censure car elle n'a pas le contrôle d'une seule partie, mais plutôt le concept de nœuds dignes de confiance pour la validation et les protocoles de consensus qui approuvent les transactions en utilisant des contrats intelligents. [45]
- Sécurité renforcée : La blockchain utilise des techniques de hachage pour stocker chaque transaction sur un bloc. Nous utilisons l'algorithme de hachage SHA-256 pour sécuriser et stocker les transactions. [45]
- Immuabilité : Les données ne peuvent pas être altérées dans la technologie blockchain en raison de sa structure décentralisée, de sorte que tout changement sera reflété dans tous les nœuds afin que l'on ne puisse pas commettre de fraude ici, d'où l'on peut affirmer que les transactions sont infalsifiables. [45]
- Transparence : les historiques des transactions sont transparents partout où tous les nœuds du réseau ont une copie de la transaction dans le réseau. Si des modifications se produisent dans la transaction, elles sont visibles par les autres nœuds. [45]

8. Limites et défis de la Blockchain

Bien que les blockchains offrent un certain nombre d'avantages, elles présentent certaines limites qui pèsent souvent sur le cas d'utilisation et la mise en œuvre finale, en particulier pour les applications en temps réel. Les principales limitations sont énumérées ci-dessous afin d'évaluer la pertinence de son utilisation selon les cas d'usage spécifiques. [45] :

- Évolutivité : les blockchains utilisant mécanisme de preuve de travail sont lentes et énergivores par rapport au systèmes centralisé traditionnels. [45]

- Complexité et versions : la blockchain est complexe, il est donc difficile pour un non-expert de l'utiliser. Les implémentations sont hybrides et les entreprises utilisent des blockchains isolées. [45]

- Inefficiences intrinsèques : les transactions ne peuvent pas être modifiées, seule l'addition est autorisée. Aucune transaction transmise n'a un sens sans transfert précédent. Par conséquent, cela crée une partie redondante dans la blockchain qui devient de plus en plus lourde avec le temps. [45]

- Sécurité : Les blockchains, bien que sécurisées, sont toujours attaquables, et par exemple l'attaque à 51 % d'Ethereum de 2016. De plus, même DDoSing la blockchain. En outre, on sait à peu près qu'un ordinateur quantique peut facilement briser les algorithmes de chiffrement cryptographique actuels, c'est pourquoi diverses solutions post-quantiques sont à l'étude. [45]

9. Domaines d'applications de la Blockchain

Initialement popularisée par les cryptomonnaies (Bitcoin), la technologie blockchain a révélé son vaste potentiel applicatif avec l'avènement de la "Blockchain 2.0" (vers 2013). Ses applications s'étendent aujourd'hui bien au-delà de la finance décentralisée, touchant de nombreux autres secteurs (comme l'Internet des objets, la gestion des droits numériques, les domaines gouvernementaux ou juridiques). Les sections suivantes détailleront divers cas d'utilisation. [40]

➢ Internet des objets (*IOT*)

L'Internet des objets a récemment gagné en popularité en raison de son potentiel de transformation des applications professionnelles et de la vie quotidienne.

La blockchain permet *aux objets* de communiquer et d'effectuer des transactions directement les uns avec les autres et, avec la disponibilité de contrats intelligents (*smart contracts*), la négociation et les transactions financières peuvent également se produire directement entre les appareils au lieu de nécessiter un intermédiaire, une autorité ou une intervention humaine. Par exemple, si une chambre d'hôtel est vacante, il peut se louer tout seul, négocier le loyer et ouvrir la serrure de la porte pour un humain qui a payé le bon montant de fonds. Un autre exemple pourrait être que si une machine à laver est à court de détergent, elle pourrait la commander en ligne après avoir trouvé le meilleur prix et la meilleure valeur sur la base de la logique programmée dans son contrat intelligent. [22]

➢ Gouvernement

Il existe actuellement diverses applications de la blockchain à l'étude qui peuvent soutenir les fonctions gouvernementales et faire passer le modèle actuel de l'e-gouvernement au niveau supérieur. [22]

Le concept n'est pas nouveau et a été mis en œuvre dans divers pays du monde, mais avec la blockchain, une nouvelle voie d'exploration s'est ouverte. De nombreux gouvernements étudient la possibilité d'utiliser la technologie blockchain pour gérer et fournir des services publics. La transparence, l'auditabilité et l'intégrité sont des attributs de la blockchain qui peuvent grandement contribuer à la gestion efficace de diverses fonctions gouvernementales. [22]

➢ Identification des citoyens (ID cards)

À l'heure actuelle, les cartes d'identité électroniques ou cartes d'identité nationales sont délivrées par divers pays du monde. Ces cartes sont sécurisées et possèdent de nombreuses fonctionnalités de sécurité qui contrecarrent les tentatives de duplication ou de falsification. Cependant, avec l'avènement de la technologie blockchain, plusieurs améliorations peuvent être apportées à ce processus.

Ou la blockchain fournit un enregistrement immuable de chaque modification et transaction effectuée par une identité numérique, garantissant ainsi l'intégrité et la transparence du système. Les citoyens peuvent également notarier les certificats de naissance, les mariages, les actes et de nombreux autres documents sur la blockchain liés à leur identité numérique comme preuve d'existence.

➢ Santé

L'industrie de la santé a été identifiée comme une autre industrie majeure qui peut bénéficier de l'adaptation de la technologie blockchain.

La blockchain fournit un système immuable, auditable et transparent. Dans le secteur de la santé, des problèmes majeurs tels que les compromissions de la vie privée, les violations de données, les coûts élevés et la fraude peuvent découler d'un manque d'interopérabilité, de processus trop complexes, de transparence, d'auditabilité et de contrôle. Les médicaments contrefaits constituent un autre problème brûlant. En particulier dans les pays en développement, il s'agit d'une source de préoccupation majeure.

Grâce à l'adaptabilité de la blockchain dans le secteur de la santé, plusieurs avantages peuvent être réalisés, allant de la réduction des coûts, une confiance accrue, un traitement plus rapide des réclamations, une haute disponibilité, l'absence d'erreurs opérationnelles dues à la complexité des procédures opérationnelles et la prévention de la distribution de médicaments contrefaits. [22]

➢ Vote électronique

Les systèmes de vote basés sur la blockchain peuvent résoudre ces problèmes en introduisant une sécurité et une transparence de bout en bout dans le processus. La sécurité est assurée sous la forme de l'intégrité et de l'authenticité des votes en utilisant la cryptographie à clé publique qui est standard dans une blockchain. De plus, l'immuabilité garantie par la blockchain garantit que les votes exprimés une fois ne peuvent pas être exprimés à nouveau. Cela peut être réalisé grâce à une combinaison de fonctionnalités biométriques et d'un contrat intelligent qui maintient une liste de votes déjà exprimés. Par exemple, un contrat intelligent peut conserver une liste de votes déjà exprimés avec l'identifiant biométrique (par exemple, une empreinte digitale) et peut l'utiliser pour détecter et empêcher le double casting. Deuxièmement, les preuves à divulgation nulle de connaissance peuvent également être utilisées sur la blockchain pour protéger la vie privée des électeurs sur la blockchain. [22]

10. Plateformes Blockchain

Le développement et le déploiement d'applications basées sur la technologie blockchain s'appuient sur des plateformes dédiées. Celles-ci fournissent l'environnement et les outils nécessaires pour créer et exécuter des applications décentralisées (dApps), en garantissant des propriétés clés telles que la transparence et l'immuabilité. Nous présentons dans cette section présente les principales plateformes blockchain courantes.

➢ BlockApps

BlockApps est une plate-forme qui fournit un riche ensemble d'outils pour créer des applications blockchain. Cette plateforme est écrite en Haskell et est basée sur une architecture modulaire. La solution est évolutive et facilite le déploiement de contrats intelligents et d'applications blockchain. [22]

➢ Eris

Eris n'est pas une blockchain unique, il s'agit d'une plateforme modulaire ouverte développée par *Monax* pour le développement d'applications d'écosystème basées sur la blockchain. Il propose divers cadres, *SDK* et outils qui permettent d'accélérer le développement et le déploiement d'applications blockchain. L'idée centrale de la plate-forme d'application *Eris* est de permettre le développement et la gestion d'applications de l'*écosystème* avec un backend blockchain. Il permet l'intégration avec plusieurs blockchains et permet à divers systèmes tiers d'interagir avec divers autres systèmes. Cette plate-forme utilise des contrats intelligents écrits en langage *solidity*. Il peut interagir avec des blockchains telles que Ethereum ou bitcoin. L'interaction peut inclure des commandes de connectivité, le démarrage, l'arrêt, la déconnexion et la création de nouvelles blockchains. La complexité liée à la configuration et à l'interaction avec les blockchains a été abstraite dans *Eris*. Toutes les commandes sont standardisées pour différentes blockchains, et les mêmes commandes peuvent être utilisées sur toute la plateforme, quel que soit le type de blockchain ciblé. [22]

Une application de l'écosystème peut être constituée de la plate-forme *Eris*, ce qui permet à la passerelle API de permettre aux applications existantes de se connecter aux systèmes de gestion clés, aux moteurs de consensus et aux moteurs d'applications. La plate-forme *Eris* fournit

diverses boîtes à outils qui sont utilisées pour fournir divers services aux développeurs. Ces modules sont décrits comme suit :

1. Chaînes : Cela permet la création et l'interaction avec des blockchains.
2. Package : Cela permet le développement de contrats intelligents.
3. Clés : elles sont utilisées pour la gestion des clés et les opérations de signature.
4. Fichiers : Cela permet de travailler avec des systèmes de gestion de données distribués. Il peut être utilisé pour interagir avec des systèmes de fichiers tels que IPFS et des lacs de données.
5. Services : Il s'agit d'un ensemble de services qui permet la gestion et l'intégration d'applications de l'écosystème. [22]

➢ Corda

Corda n'est pas une blockchain traditionnelle, mais plutôt une plateforme distribuée open-source conçue spécifiquement pour les applications d'entreprise. Développée par R3, elle se distingue par son approche unique axée sur la confidentialité, l'interopérabilité et la scalabilité. Contrairement aux blockchains publiques comme Bitcoin ou Ethereum, Corda est conçue pour les réseaux permissionnés, où seuls les participants autorisés peuvent interagir et valider les transactions. [46]

L'idée centrale de Corda est de permettre aux entreprises de collaborer tout en conservant la confidentialité de leurs données. Les transactions sur Corda sont partagées uniquement entre les parties concernées, ce qui réduit le besoin de divulguer des informations sensibles à l'ensemble du réseau. Corda utilise des contrats intelligents (appelés "CorDapps" pour Corda Distributed Applications) écrits en langages de programmation courants tels que Java et Kotlin. [46]

Corda peut interagir avec d'autres systèmes et blockchains grâce à son architecture modulaire. Elle prend en charge l'intégration avec des bases de données existantes, des systèmes de gestion de fichiers et d'autres technologies distribuées. La complexité liée à la configuration et à la gestion des réseaux est simplifiée grâce à des outils tels que le Corda Network Map et le Notary Service, qui permettent de gérer les identités et de valider les transactions de manière sécurisée. [46]

Une application basée sur Corda peut inclure les composants suivants :

- Réseau : Permet la création et la gestion de réseaux distribués permissionnés.
- CorDapps : Facilite le développement et le déploiement d'applications distribuées pour exécuter la logique métier.
- Identités : Gère l'authentification et l'autorisation des participants via des certificats numériques.
- Stockage : Prend en charge des systèmes de stockage distribués pour les données transactionnelles et les états partagés.
- Services : Fournit des services supplémentaires tels que la gestion des clés, le chiffrement et l'intégration avec des systèmes tiers.

Corda est largement utilisée dans des secteurs tels que la finance, les assurances et la chaîne d'approvisionnement, où la confidentialité et la conformité réglementaire sont essentielles. Par exemple, dans le secteur financier, Corda est utilisée pour automatiser les processus de règlement-livraison (*settlement*) et de gestion des contrats intelligents. [46]

11. Le rôle de la blockchain dans la gestion d'identité décentralisé

Face aux limitations des modèles traditionnels d'identité centralisée (vulnérabilités, risques pour la vie privée, manque de contrôle utilisateur), la blockchain offre une infrastructure décentralisée, immuable et sécurisée. Elle permet de stocker et de vérifier cryptographiquement des éléments d'identité sans dépendre d'une autorité centrale, agissant comme un registre de confiance résilient. Cette approche s'appuie sur des concepts clés, tels que:

- Verifiable credentials (VC) : attestations sécurisées émises par des tiers de confiance et vérifiables cryptographiquement. Cela sécurise l'identité tout en respectant la vie privée des utilisateurs. [47]
- Identifiants Décentralisés (DID) : identifiants uniques, créés et contrôlés par l'utilisateur, souvent liés à la blockchain. Qui permettent de relier efficacement une clé publique à l'identité numérique d'un utilisateur, renforçant l'intégrité des données et le contrôle des utilisateurs sur leurs informations personnelles. [47]

Le mécanisme de cette approche implique qu'un utilisateur crée un DID, des émetteurs délivrent des VCs signés liés à ce DID, et l'utilisateur gère ses VCs dans un portefeuille d'identité. Pour prouver une information, l'utilisateur présente le VC concerné à un vérificateur, qui utilise la cryptographie et les informations publiques (souvent inscrits sur la blockchain) pour valider le VC sans intermédiaires centraux. Cette gestion d'identité décentralisée offre des avantages significatifs tels que :

- Contrôle utilisateur : L'utilisateur est au centre, il possède et gère son identité et ses données. Cela aligne avec le concept d'identité autonome (Self-Sovereign Identity - SSI). [47]
- Vie privée : Possibilite d'un partage sélectif d'informations (juste la preuve nécessaire via un VC spécifique).
- Sécurité : Pas de base de données centrale unique à pirater, cryptographie forte.

12. Solutions existantes de gestion d'identité décentralisée (DID)

Dans cette section nous allons explorer les solutions existantes de la gestion des identifiants décentralisés (DID), en tant qu'approches pour les systèmes d'identité décentralisée basés sur la blockchain.

12.1. UPort

UPort est un *Framework* open source permettant de fournir une identité déclinée pour une identité auto-souveraine. Il est basé sur la blockchain publique sans autorisation Ethereum et utilisant ses contrats intelligents. Grâce à ce *Framework*, les utilisateurs peuvent publier en toute sécurité leur identité, y compris le transfert de leurs VC, signer des transactions et contrôler leurs clés et données. Une identité uPort peut être créée pour les utilisateurs, les organisations et d'autres ressources. L'identité est entièrement détenue et régie par son propriétaire et non par les tiers. De plus, toutes les données personnelles et confidentielles liées à l'identité sont détenues par le propriétaire dans son portefeuille numérique, de sorte que les divulgations d'informations sont réduites au minimum. Les composants du système uPort sont illustres dans la figure suivante [48] :

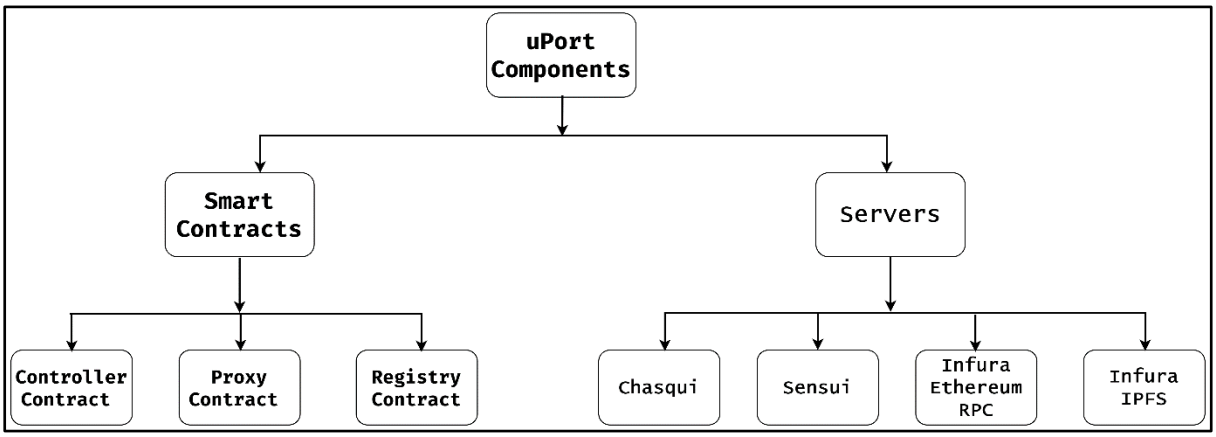

Figure II-6 Composants uPort [48]

➢ Composants du contrat intelligent :

- Contrat de registre : 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 permettant de stocker et d'extraire des données sur un système de fichiers distribué. Le contrat de proxy ne peut mettre à jour que le contrat de registre [48].

- Contrat de contrôleur : Il s'agit de la logique de contrôle globale avec la fonctionnalité de contrôle de l'accès au contrat proxy. De plus, il permet à l'utilisateur de récupérer son identité s'il perd sa clé mobile et 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 proxy : Il s'agit de l'identifiant permanent d'un utilisateur lié à la clé privée de l'utilisateur, par conséquent, il permet à l'utilisateur de remplacer sa clé privée sans affecter son identité permanente.

➢ Composants du serveur :

- Chasqui : Le serveur de messages gère tous les aspects des communications avec n'importe quelle application décentralisée.
- Sensui : Le serveur de ravitaillement en gaz évite qu'un nouvel utilisateur d'Ethereum n'ait à acheter de l'Ether et à payer des frais pour utiliser le réseau. Il paie les frais de gaz pour le nouvel utilisateur, ce qui lui permet de créer instantanément un nouveau compte uPort.
- Infura Ethereum RPC : Cette API Infura fournit une interface RPC (*Remote Procedure Call*) standard pour permettre à uPort de communiquer avec le réseau Ethereum.
- Infura IPFS : L'API Infura fournit une interface standard pour permettre à uPort de communiquer avec le réseau IPFS. [48]

Le diagramme illustré ci-dessous décrit avec précision le fonctionnement du système UPort, ainsi que les diverses interactions entre ses différents éléments et parties prenantes.

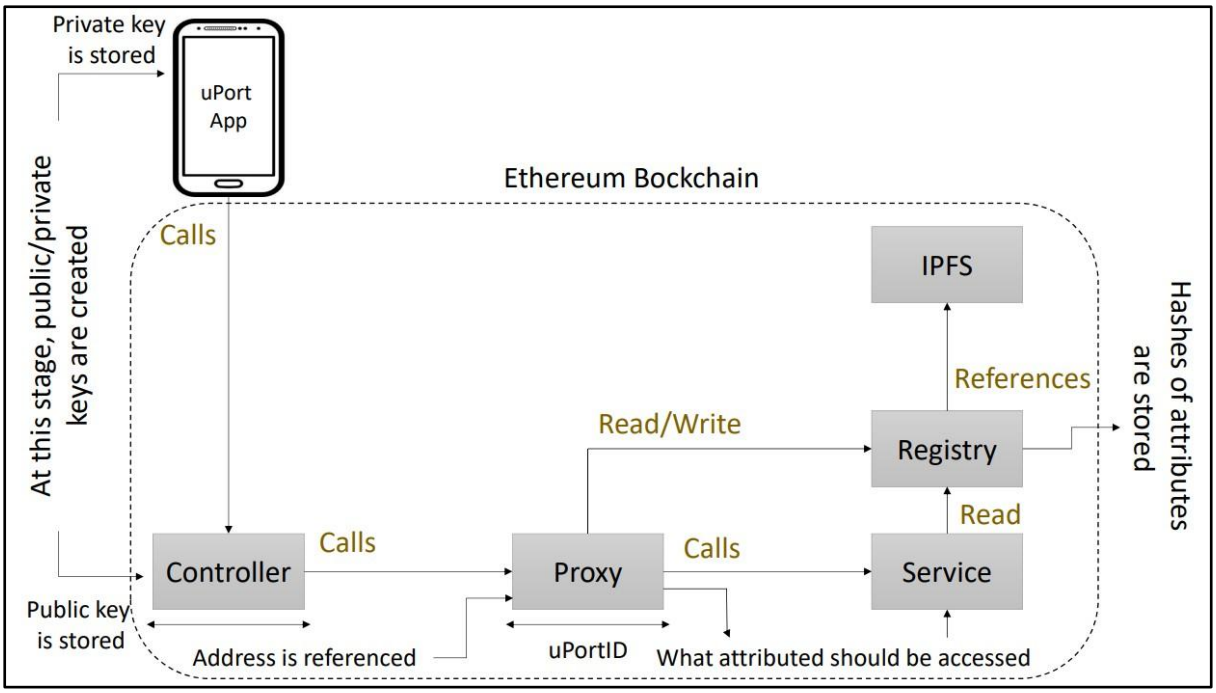

Figure II-7 L'architecture générale de uPort. IPFS, InterPlanetary File System. [49]

➢ Les composants systèmes uPort :

Les principaux composants sont les suivants :

- Application uPort : Stocke la clé privée de l'utilisateur (gestion des identités sécurisée) et génère des identifiants décentralisés uPort.
- Ethereum Blockchain : Sert de couche d'infrastructure immuable pour enregistrer les clés publiques liées aux identités. Elle héberge le Registre (Registry) permettant les opérations de lecture telles que la vérification d'identité et l'association entre DID et adresses.

  Stockez les clés publiques et leurs adresses en association avec les identités.

  Héberge le Registre (Registry) pour les opérations de lecture (par exemple, vérification d'identité).

- L'IPFS : Stockage décentralisé des données, telles que les attributs d'identité, et référencement de celles-ci par l'utilisation de liens.
- Proxy et service : Le Proxy agira en tant qu'intermédiaire lors de la communication entre l'application et la blockchain. Le Service détermine quelles propriétés d'identité peuvent être consultées (contrôle des autorisations).

- Controller : Coordonnez les interactions, tels que la validation des clés et l'accès aux données.

12.2. Sovrin

Sovrin est un Framework open-source permettant de fournir une identité décentralisée. Il est basé sur la blockchain Hyperledger Indy, autorisée par le public. Comme il s'agit d'une blockchain autorisée, seules les institutions de confiance appelées stewards peuvent exploiter des nœuds tout en participant au processus de consensus. Grâce à ce Framework, les utilisateurs peuvent publier en toute sécurité leur identité, y compris transférer leurs informations d'identification, signer des transactions et contrôler leurs clés et données. Une identité Sovrin peut être créée pour les utilisateurs, les organisations et d'autres ressources. Sovrin permet aux utilisateurs de créer une identité différente avec sa propre paire de clés privées et publiques pour différents contextes afin de préserver la confidentialité. Un utilisateur détermine le type d'attributs à associer à son identité. Il utilise des VC anonymes basées sur des preuves à divulgation nulle de connaissance (ZKP), pour préserver l'anonymat de l'identité d'un utilisateur. L'identité est entièrement détenue par le propriétaire de cette identité et gérée par l'utilisateur ou le service de tutelle désigné par les utilisateurs. De plus, toutes les données personnelles et confidentielles liées à l'identité sont détenues par le propriétaire dans son portefeuille numérique en périphérie ou dans le cloud. Les composants du système Sovrin sont les suivants [48]:

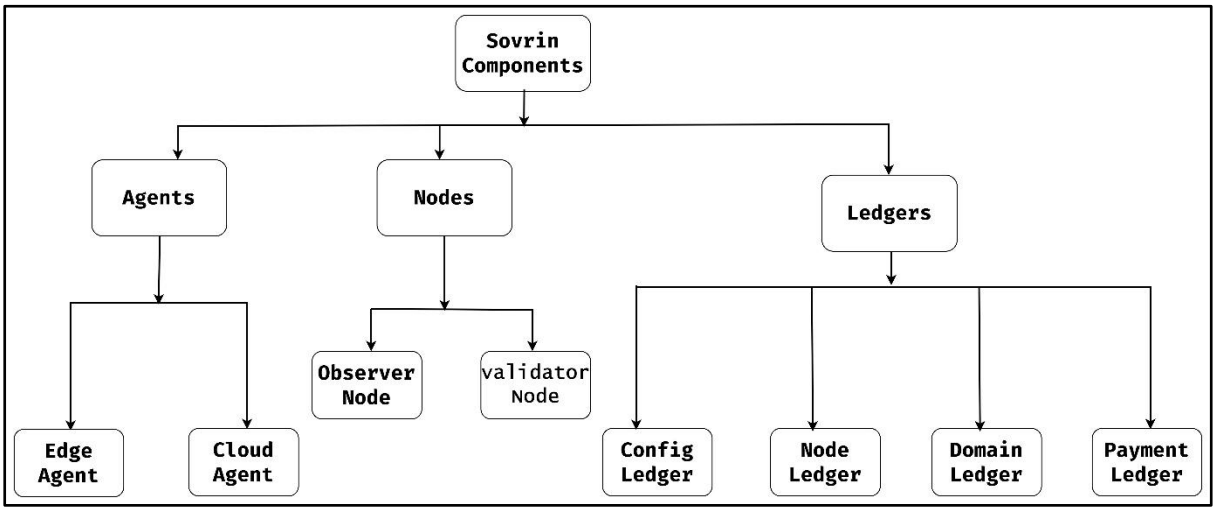

Figure II-8 Composants Sovrin [48]

Le diagramme présenté ci-dessous modélise de manière détaillée l'architecture opérationnelle Sovrin, mettant en lumière les flux d'interactions entre ses composants techniques et acteurs institutionnels

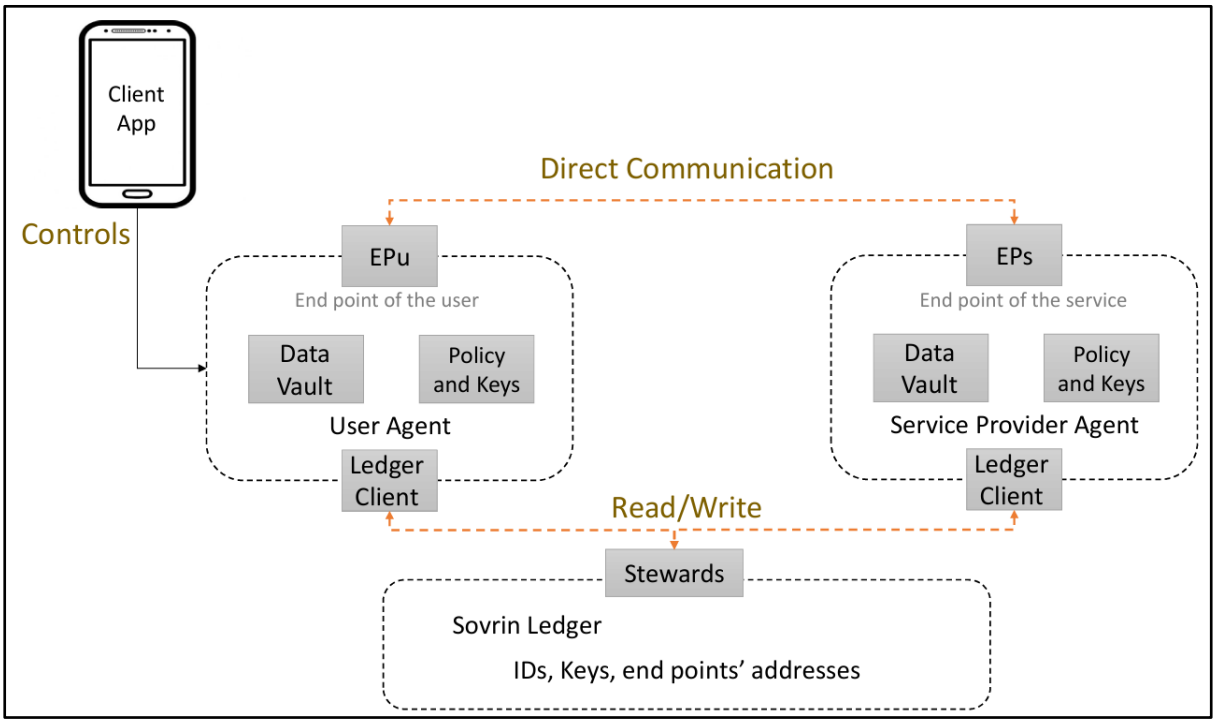

Figure II-9 L'architecture du système de gestion d'identité Sovrin [49]

➢ Les principaux composants

Les principaux composants de la solution sont détaillés ci-dessous :

- Cloud Agent : Cet agent est hébergé sur le cloud qui n'est pas directement contrôlé par un propriétaire d'identité. L'agent de périphérie communique avec l'agent cloud qui s'exécute 24 h/24 et 7 j/7 et propose un service de stockage et de transfert pour acheminer les demandes vers et depuis l'agent de périphérie.

- Edge Agent : Cet agent est hébergé sur l'appareil de l'utilisateur (Edge of the network) tel que les mobiles, les tablettes ou les ordinateurs portables. Cet agent peut être connecté à une autre application et accéder à un portefeuille contenant des clés et des VC, exécutant les fonctions cryptographiques pour cette entité.

- Agents Sovrin : Un agent Sovrin est un programme requis pour qu'un propriétaire d'identité ou toute autre entité participante interagisse les uns avec les autres dans le processus gestion DID. Les agents travaillent dans un modèle peer-to-peer et partagent

entre eux des VC DID et autres informations. Ils ne nécessitent pas d'accès à la blockchain et communiquent des messages complets signés et cryptés. Chaque agent accède au portefeuille et exécute des fonctions cryptographiques pour cette entité.

- Sovrin Nodes : un nœud est un serveur qui exécute une instance du code nécessaire au fonctionnement d'un registre. Un nœud peut être soit un nœud de validation, soit un nœud observateur ; Cependant il ne peut agir qu'un par un. [48]

    - Nœud observateur : ce nœud exécute l'instance en lecture seule du registre. N'importe qui peut exécuter ce nœud en n'importe quel nombre ; cependant, sa réponse peut être vérifiée par des preuves d'État.

    - Nœud de validation : ce nœud valide toutes les nouvelles transactions et écrit dans le registre en fonction du protocole de consensus. Un steward exécute un nœud de validation à la fois.

- Sovrin Ledgers : Il s'agit d'un registre distribué permettant de tenir les registres de différents types de transactions.

    - Registre de configuration : il s'agit d'un registre spécial pour l'enregistrement des actions de transaction liées à la configuration d'un registre. Il n'est pas accessible en écriture publique et seuls les administrateurs de Sovrin ou leurs délégués peuvent écrire dans ce registre.

    - Registre des nœuds : Ce registre permet d'enregistrer les transactions liées à l'identification des nœuds autorisés. Ce registre est lisible par le public, mais pas par le public, et seuls les fiduciaires ou les intendants peuvent y écrire.

    - Domain Ledger : Ce registre permet d'enregistrer les transactions liées à la gestion des identités (hors paiements). Ce registre est lisible et accessible en écriture sur la base de la méthode de protection du cadre de gouvernance Sovrin.

- Registre des paiements : Ce registre permet d'enregistrer les transactions liées aux paiements. Il est lisible et accessible en écriture sur la base de la méthode d'accès public en écriture du cadre de gouvernance Sovrin [48]

12.3. ShoCard

ShoCard est une plateforme d'authentification d'identité basée sur la blockchain qui vise à fournir une identité de confiance en protégeant l'identité des utilisateurs. Il utilise la technologie blockchain pour lier un identificateur d'utilisateur, des informations d'identification de confiance existantes et des attributs d'identité supplémentaires, ainsi que des hachages cryptographiques à prendre en compte pour des transactions similaires aux transactions en face à face. ShoCard utilise pour les hachages cryptographiques signés des informations d'identité de l'utilisateur. La figure montre comment les transactions se déroulent dans un réseau ShoCard. [49]

Le serveur ShoCard fonctionne comme un intermédiaire pour gérer la certification échangée entre les utilisateurs et les RP (Relying Parties). Cependant, la conception est plus centralisée que de s'appuyer sur la blockchain, ce qui signifie que si une entreprise émettrice cesse d'exister, les identités sont inutilisables. Les utilisateurs doivent télécharger des informations d'identification, ce qui nécessite plus d'informations que nécessaire. Par exemple, une copie numérisée d'un passeport doit être téléchargée. De plus, ShoCard stocke des données cryptées, mais les données peuvent être associées à la fois aux ShoCardID et aux parties utilisatrices, ce qui compromet la confidentialité. [49]

ShoCard prend généralement en charge les identifiants unidirectionnels, bien que des identifiants omnidirectionnels soient nécessaires pour réaliser un écosystème où les certifications peuvent être réutilisées. ShoCard prend en charge une multitude de fournisseurs d'identité et offre une expérience utilisateur cohérente et simple. D'autre part, il est ambigu de savoir si les utilisateurs seront motivés à utiliser un IdM avec de telles limitations et disposés à être éduqués sur les implications de la gestion de leur identité sur la technologie blockchain. [49]

La figure illustrée ci-dessous représente de manière schématique l'architecture globale de la solution ShoCard, en mettant en évidence les différents composants qui la constituent ainsi que les interactions entre eux dans le cadre du fonctionnement.

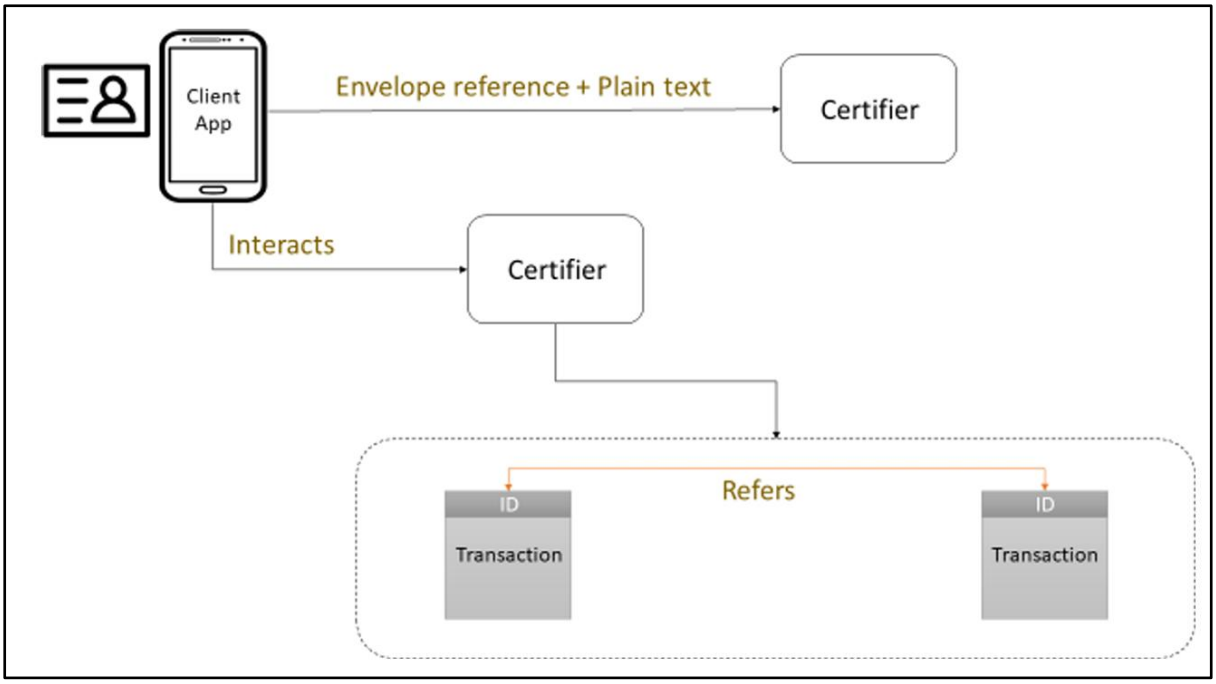

Figure II-10 L'architecture de ShoCard [49]

La figure illustre l'interaction entre une application cliente et un certifier, où l'utilisateur envoie une référence d'enveloppe et des données en clair. Le certifier vérifie ces données et enregistre la preuve sur la blockchain sous forme de transactions identifiables.

L'application client représente l'application mobile ou web utilisée par l'utilisateur pour gérer identité, elle permet de génèrer et envoie des informations d'identité au certifier le plain text peut contenir des données lisibles (nom, email...), tandis que la référence d'enveloppe peut pointer vers des données chiffrées ou stockées ailleurs (par exemple : IPFS).

13. Conclusion

Avec ses propriétés fondamentales de sécurité, d'immuabilité et de transparence, la Blockchain se positionne comme une technologie prometteuse pour la gestion des identités numériques. En éliminant les intermédiaires et en offrant un contrôle accru aux utilisateurs, elle permet de poser les bases d'un nouveau modèle d'identité plus robuste et résilient. Cependant, son adoption à grande échelle reste un défi, en raison de limitations techniques (scalabilité, gestion des cles, ... etc.) et de considérations réglementaires.

Ce chapitre a ainsi exploré les fondements, les caractéristiques clés et les domaines d'application de la technologie Blockchain, en mettant en évidence son potentiel pour la décentralisation, notamment appliqué à la gestion d'identité.
Le chapitre suivant mettra en pratique ces principes en détaillant le développement d'une solution de gestion d'identité décentralisée.

Chapitre III Conception et implémentation du système

1. Introduction

L'identité décentralisée constitue une évolution innovante dans la gestion d'identité numérique, offrant aux utilisateurs un control total de leurs informations personnelles. Basée sur la technologie blockchain, elle permet une protection de la vie privée.

Dans ce chapitre, consacré au développement de notre application de gestion d'identité décentralisée, nous détaillerons les outils et technologies utilisés, l'architecture de la solution et sa mise en œuvre. Nous commencerons par présenter l'environnement de développement, les frameworks et les protocoles blockchain utilisés. Ensuite, nous examinerons les principaux composants de l'application.

Ce chapitre vise à démontrer la faisabilité d'une solution DID opérationnelle dans son contexte, tout en soulignant son potentiel pour de futurs usages et applications.

2. Environnement de travail

➢ Matériel :

- Nom Machine: VivoBook_ASUSLaptop X409FA
- Système d'exploitation: Win 11 Home x64bits 24H2 (OS Build 26100.2605)
- CPU : Intel(R) Core (TM) i7-8565U CPU @ 1.80GHz (8 CPUs), ~2.0GHz
- Memoire: 16GO RAM
- GPU : Intel(R) UHD Graphics 620, 1920 x 1080 (32 bits) (60Hz)

➢ Outils de développement

- Editeur de texte :

Un éditeur de code open-source et multiplateforme développé par Microsoft. Il support de nombreux langages avec un débogueur intégré et des extension puissantes telles que Truffle for VS Code pour la blockchain [50] [51].

- Truffle Suite :

Un Framework de développement, de test et de déploiement des contrats intelligents pour les blockchains utilisant la machine virtuelle Ethereum (EVM) [52].

- Ganache

Ganache fait partie de la suite Truffle, il permet de simuler une blockchain Ethereum localement. Ganache permet de développer, déployer et tester des applications décentralisées (D'Apps) dans un environnement local rapide, sûr et déterministe [50].

- Metamask SDK

Est un portefeuille numérique sous forme d'extension de navigateur ou d'application mobile permettant de gérer des crypto-actifs et d'interagir avec des applications décentralisées (dApps). Il sert aussi de passerelle entre l'utilisateur et la blockchain (comme Ethereum) en gérant les clés privées localement. [53].

- Solidity

Solidity est un langage de haut niveau orienté objet conçu pour la mise en œuvre des contrats intelligents sur diverses plateformes blockchain, notamment Ethereum et d'autres compatibles avec la Machine Virtuelle Ethereum (EVM). [54]

- ReactJs

Une bibliothèque JavaScript pour construire des interfaces utilisateur (UI) interactives. React utilise une structure modulaire basée sur des blocs réutilisables [55]

- NodeJs

Node.js s'agit d'un environnement d'exécution JavaScript open source et multiplateforme. C'est un outil populaire pour presque tous les types de projets ! [56]

Node.js présente plusieurs avantages pour le développement web, notamment sa rapidité, sa flexibilité et sa simplicité. Il permet également de développer des applications en temps réel, de gérer des flux d'entrée simultanés et de tirer parti d'un vaste écosystème de bibliothèques.

- Draw.io

Est un outil en ligne gratuit de modélisation graphique permettant de créer des diagrammes variés tels que des diagrammes de flux, des schémas UML et OCL ou des architectures logicielles.

3. Analyse et conception logicielle

3.1. Analyse

L'analyse est l'étape cruciale dans le processus de développement de notre application de gestion d'identités décentralisées (DID).

On vise les objectifs du projet et les exigences essentielles auxquels la solution devrait répondre. Notre projet vise à permettre aux utilisateurs de créer, gérer, partager et vérifier leurs identités numériques de manière sécurisée et transparente en utilisant la technologie de la blockchain et l'authentification basée sur des wallets Web3 comme MetaMask. Une réflexion sur la faisabilité et la pertinence de l'application a été réalisée sur plusieurs aspects :

- Besoins des utilisateurs : il faut proposer un moyen simple et sûr de gérer plusieurs types d'identités, nationale, professionnelle, bancaire, médicale, etc.
- Sécurité et confidentialité : garantir que les données personnelles sont sécurisées mais restent accessible de manière vérifiable sur un réseau décentralisé
- Interopérabilité : utiliser des standards reconnus tels que les DIDs (Decentralized Identifiers) pour assurer une compatibilité future avec d'autres systèmes.
- Simplicité d'usage : rendre l'interface intuitive, notamment lors de la connexion via MetaMask et des opérations de gestion des identités.

Cette analyse permet d'englober une définition précise du fonctionnement général du système, ainsi que les cas d'utilisation principaux et des interactions entre l'utilisateur, l'IHM, la blockchain, et les contrats intelligents déployés.

3.1.1. Besoins Fonctionnels

Les besoins fonctionnels définissent les principales fonctionnalités requises du système pour la gestion décentralisée des identités.

- Cas d'utilisation : Gestion d'une identité (Nationale, bancaire ou de santé).

- Gestion de l'Identité : La gestion de l'identité décentralisée implique globalement les opérations suivantes :
  - Création d'identité : Les utilisateurs peuvent enregistrer leurs identifiants uniques sur la blockchain à l'aide d'identifiants distribués (DID).
  - Mise à jour et révocation : Les modifications d'attributs d'identité, y compris les adresses e-mail et les noms, sont autorisées. Le système permet aux utilisateurs de révoquer leurs identités compromises au moyen d'un contrat intelligent.

- Contrôle d'accès : le mécanisme du contrôle d'accès est basé sur les principes suivants :
  - L'authentification décentralisée : L'utilisateur s'authentifie via un portefeuille comme MetaMask, garantissant une gestion autonome de son identités numérique.
  - Connexion sécurisée : Chaque utilisateur peut accéder à son compte grâce à une signature cryptographique générée par son portefeuille.
  - Contrôle d'accès basé sur les rôles (RBAC) : Le système met en œuvre le contrôle d'accès RBAC pour la vérification des autorisations.

- Partage et vérification d'Identité : le partage et la vérification sécurisée des identités se font à l'aide d'attestations vérifiables (VCs), tout en préservant la vie privée de l'utilisateur.
  - Attestations vérifiables (VCs) : Les organisations délivrent des certificats numériques attestant par exemple des diplômes ou la conformité KYC (Know Your Customer). Lors du partage, les utilisateurs peuvent restreindre les données à des parties spécifiques telles que les informations sur l'âge.

3.1.2. Besoins Non-Fonctionnels

Ces besoins définissent les critères de qualité et les contraintes techniques du système.

- Sécurité : Pour protéger les données stockées en dehors la blockchain ou transmises sur réseau, l'algorithme AES-256-GCM est utilisé ainsi que l'algorithme EdDSA pour la signature numérique.

- Confidentialité : la confidentialité des utilisateurs est garantie en utilisant des pseudonymes pour représenter les identifiants publics.

- Interopérabilité : Le système prend en charge les normes DID du W3C ainsi que les normes VC.

3.2. Conception Logicielle

Dans cette section, nous présentons la conception logicielle de notre application qui repose sur une modélisation progressive à l'aide des diagrammes UML détaillés ci-dessous. Nous débutons par une représentation globale à travers un diagramme de contexte (figure III-1), qui définit l'environnement de l'application et identifie les principales interactions avec les acteurs externes.

Un diagramme de cas d'utilisation est illustré dans la figure III-2 pour formaliser les différentes fonctionnalités offertes par le système, en mettant en évidence les scénarios d'interaction entre les utilisateurs et l'application.

Le diagramme d'activités (figure III-3) détaille les processus internes et les enchaînements d'actions.

Le diagramme de séquence (figure III-4) complète cette analyse en décrivant les échanges dynamiques entre les objets au cours de l'exécution de scénarios spécifiques, permettant de mieux comprendre les interactions temporelles.

Ensuite, le diagramme de classes (figure III-5) présente la structure statique du système, en spécifiant les classes principales, leurs attributs, leurs méthodes et les relations entre elles.

Enfin, un diagramme de flux d'information dans la figure III-6 détaille les processus internes et externes au sein du système.

➢ Diagramme de contexte :

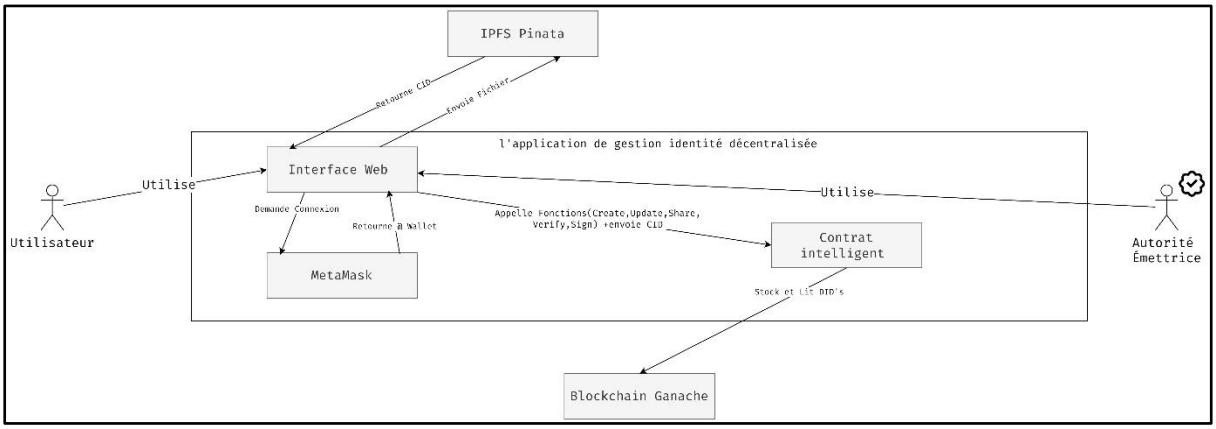

Figure III-1 Diagramme de Contexte

Ce diagramme schématise l'architecture globale de notre système d'identité décentralisée. Ci-après une brève description des différents composants :

- Acteur externe : Utilisateur, autorité émettrice

Sont les acteurs principaux qui interagit avec le système pour gérer l'identité décentralisée.

- Système principal : Application Gestion d'identité décentralisé

Permet à l'utilisateur d'interagir et d'utiliser les fonctions des modules internes, telles que la création, la mise à jour et le partage de DIDs ;

- Modules internes :
  - Interface Web (React) : Point d'entrée utilisateur
  - Contrat Intelligent (Solidity) : Logique métier sur blockchain
  - MetaMask : Portefeuille de connexion blockchain

- Module externe :

La blockchain Ganache un environnement local (Réseau Blockchain) pour tester la solution sans coûts réels et IPFS (Pinata) pour le stockage décentralisé des fichiers.

- Diagramme de cas d'utilisation

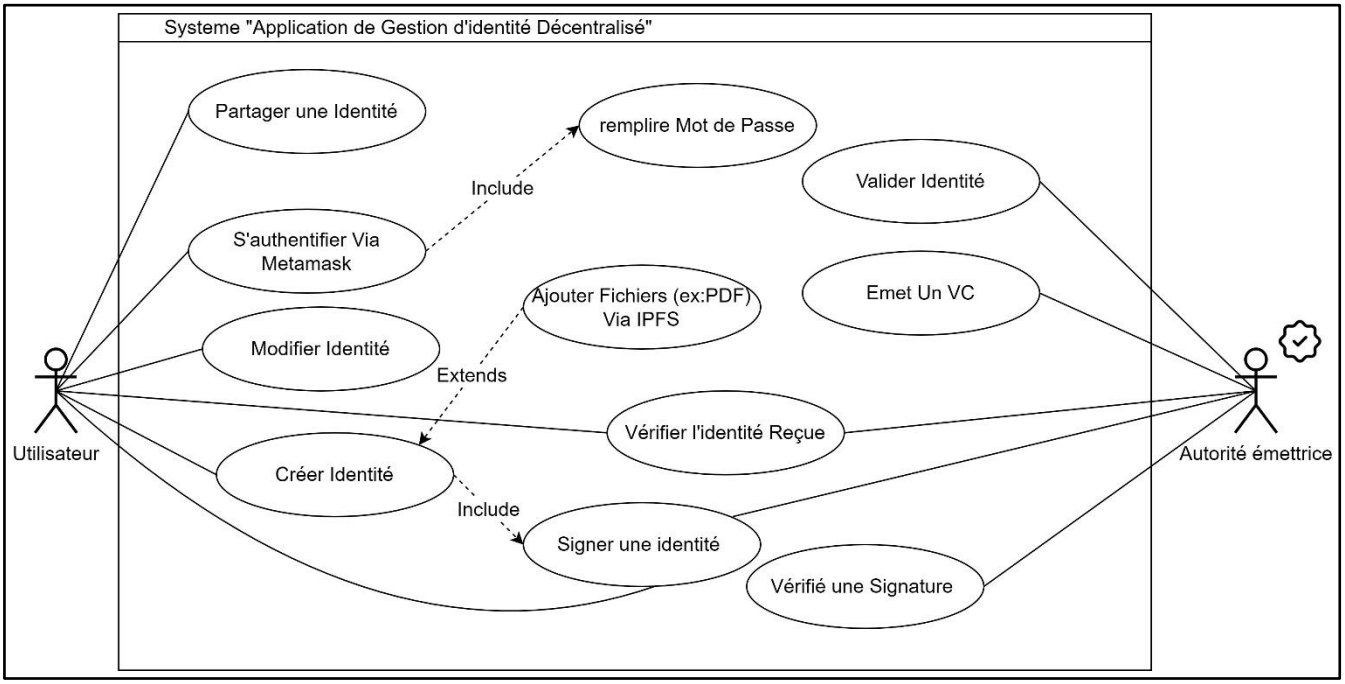

Figure III-2 Diagramme de Cas d'utilisation

Ce diagramme met en évidence les fonctionnalités essentielles offertes par l'application, telles que l'authentification via le portefeuille MetaMask, la création, la modification, le partage et la vérification des fiches d'identité. Il précise également les cas d'utilisation accessibles à l'utilisateur, tout en représentant les relations entre ces cas à travers les dépendances d'inclusion. Ce schéma permet ainsi de délimiter clairement le périmètre fonctionnel de l'application et de structurer les interactions entre ses principaux composants.

➢ Diagramme d'activité

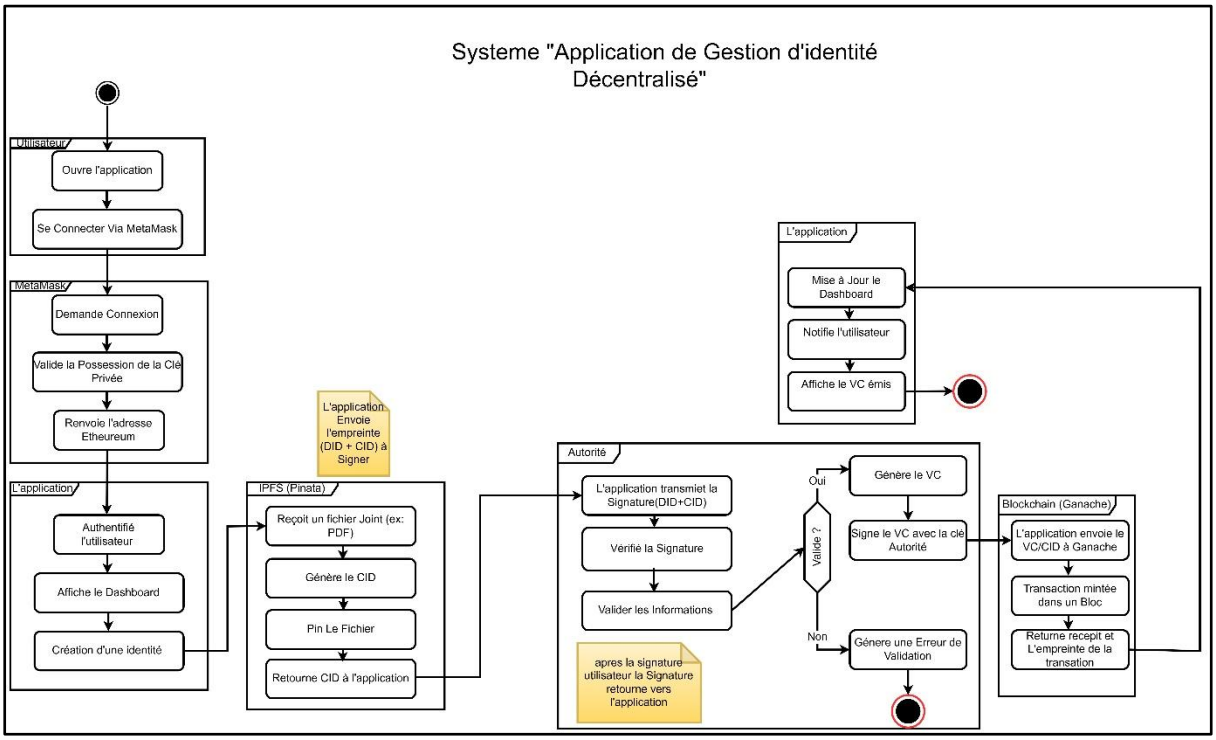

Figure III-3 Diagramme d'Activité

Ce diagramme illustre de manière détaillée et séquentielle le processus de création d'une nouvelle identité intègre l'authentification et la validation en s'appuyant sur wallet MetaMask, IPFS et la blockchain Ganache. Il présente une architecture en plusieurs phases qui assure à la fois l'authentification de l'utilisateur et la sécurisation des données transmises, tout en garantissant leur traçabilité sur la blockchain.

- Authentification

L'utilisateur se connecte via MetaMask.

- Enregistrement des données

L'utilisateur ajoute un fichier (PDF, js...etc et non pas exe) et remplit les champs d'informations et une génération d'une empreinte (DID) pour l'identité et d'un CID pour le document (stockage décentralisé).

- Signature et vérification

L'utilisateur signe le hash combiné (DID+CID) via MetaMask et L'autorité valide la signature.

- Émission du VC

Si tout est valide, l'autorité signe le VC, le Dashboard affiche le VC émis avec le hash de transaction.

➢ Diagramme de séquence

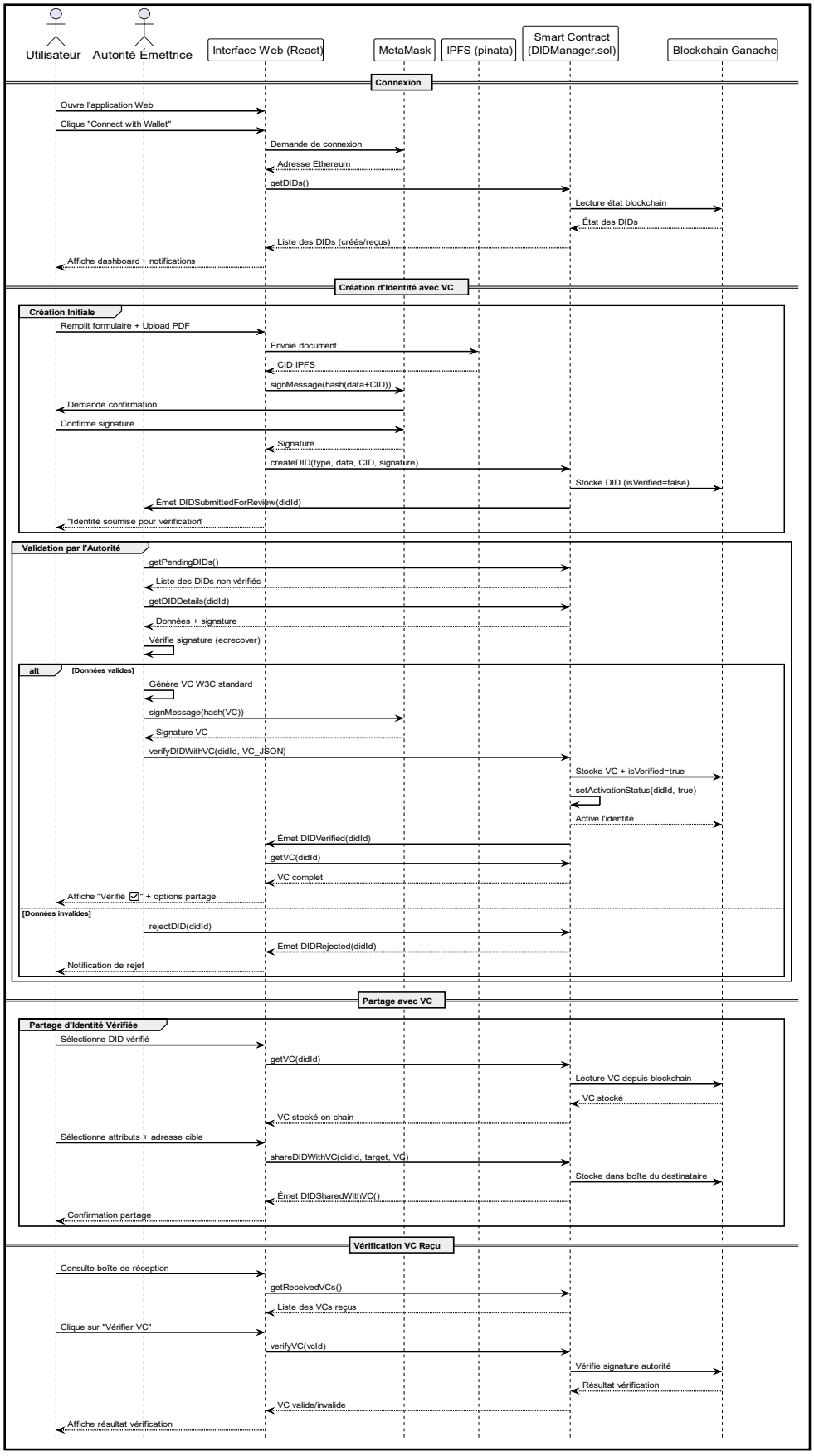

Figure III-4 Diagramme de Séquence

Le diagramme de séquence qui décrit le fonctionnement général selon l'ordre chronologique de l'application dans le cadre de (création, modification, partage et vérification de l'identité)

Ce diagramme modélise le processus complet d'interaction entre les différents acteurs et composants techniques d'une application de gestion d'identité décentralisée (DID) intégrant les (VC), la blockchain (Ganache), IPFS, et l'autorité de vérification.

➢ Diagramme de Classe

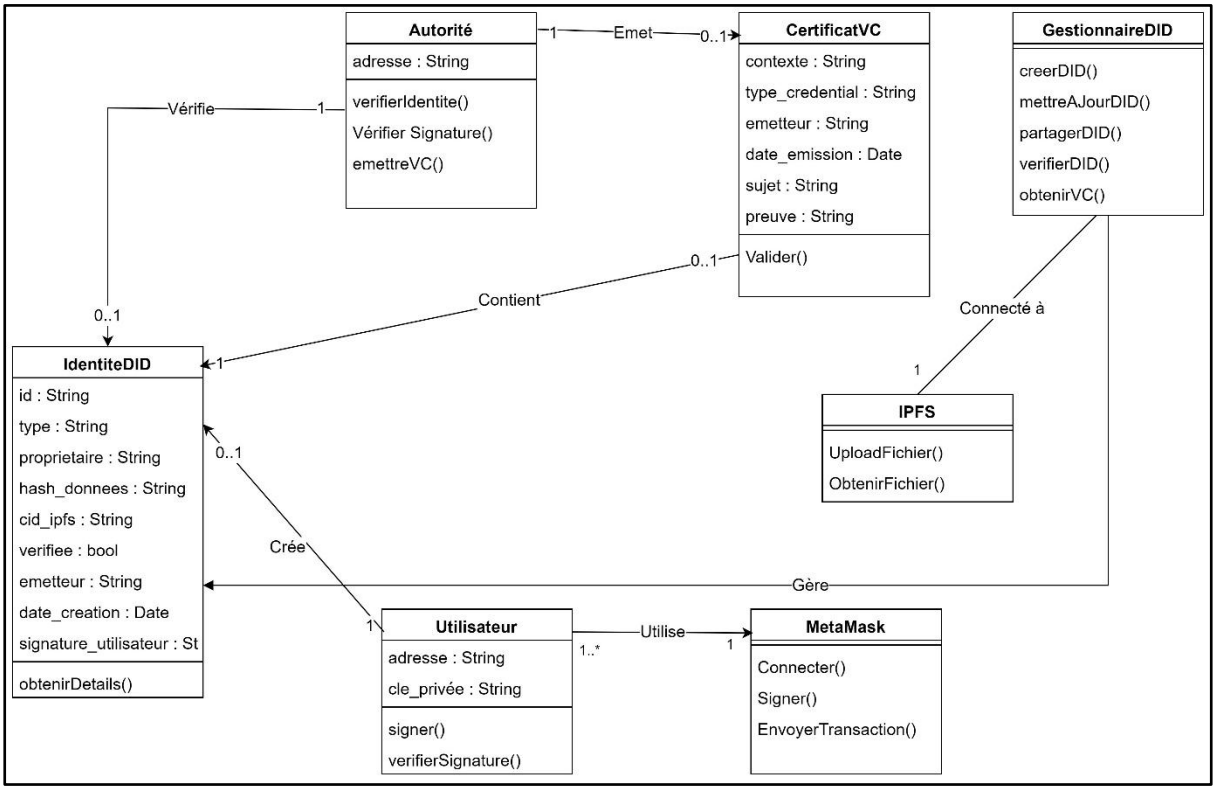

Figure III-5 Diagramme de Classe

Ce diagramme de classes représente un les différents acteurs et les différentes interactions entre eux dans le système. Voici une explication concise des composants principaux :

1. Autorité : Une entité qui vérifie les identités, valide les signatures et émet des certificats vérifiables (VC).
2. CertificatVC : Contient les détails d'un certificat vérifiable (contexte, type, émetteur, date, sujet, preuve).
3. GestionnaireDID : Gère la création, mise à jour, partage et vérification des identités décentralisées (DID).
4. IdentiteDID : Représente une identité décentralisée avec des attributs comme l'ID, le propriétaire, et des données stockées sur IPFS.
5. IPFS : Service de stockage décentralisé pour uploader et récupérer des fichiers.
6. Utilisateur : Interagit avec le système en signant des données et vérifiant des signatures.
7. MetaMask : Portefeuille crypto utilisé pour la connexion, la signature et les transactions.

➢ Diagramme de Flux d'information

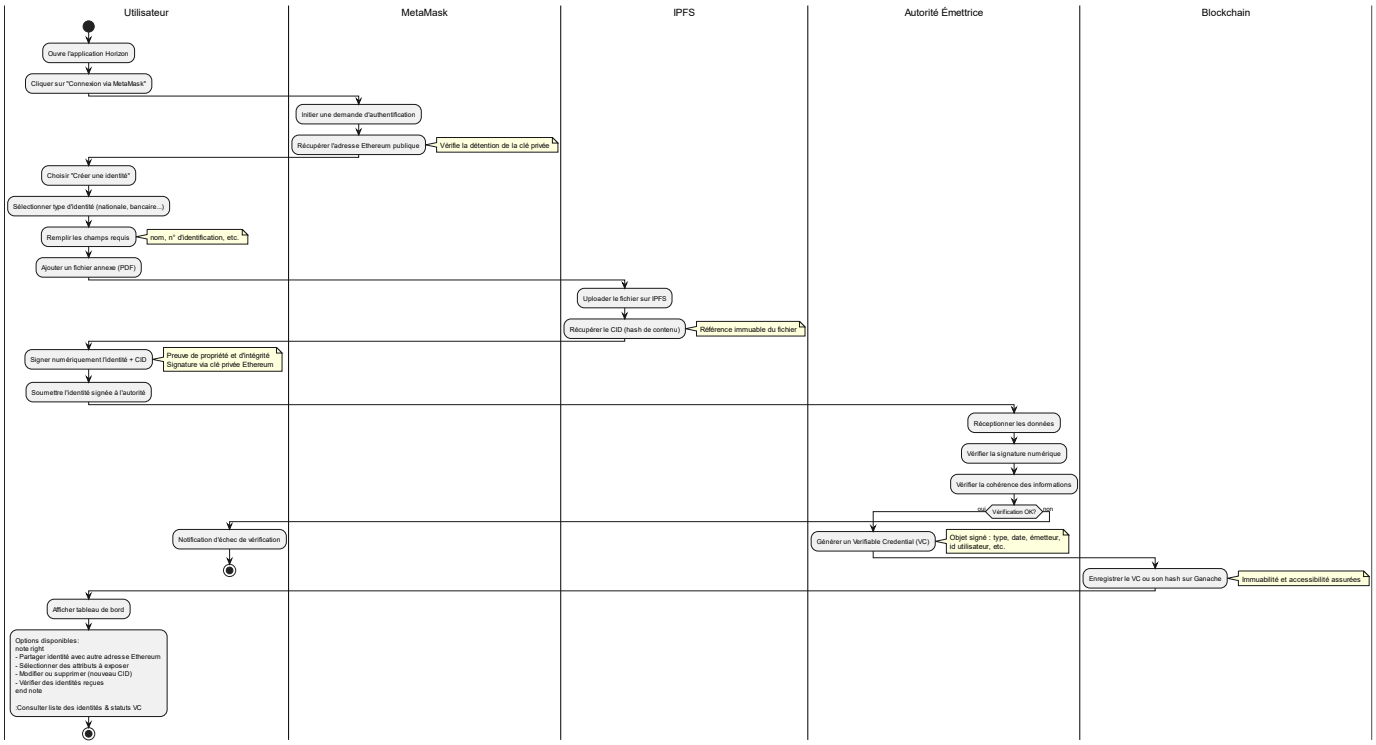

Figure III-6 Diagramme de Flux d'information

Ce diagramme illustre le fonctionnement du système basant sur des identités numériques (DID) et des certificats vérifiables (VC), intégrant des technologies comme IPFS pour le stockage et MetaMask.

- Fonctionnement :
  1. L'utilisateur crée une identité via le GestionnaireDID, signée.
  2. Une autorité vérifie l'identité et émet un certificat (VC) stocké dans la blockchain.
  3. Les données sont stockées de manière sécurisée sur IPFS.
  4. Les tiers peuvent vérifier les certificats et identités via la signature Numérique.

4. Développement de l'application

Dans cette section, nous exposons la solution DID développée, en mettant en évidence ses principales fonctionnalités à travers des captures d'écran de l'application réalisée

4.1. Front-end

Commençons d'abord par l'interface principale utilisateur et de l'autorité de vérification combinant l'ergonomie, la sécurité et la facilité d'accès aux principales fonctionnalités de l'application.

L'autorité est chargée de valider les identités crée par les différents utilisateurs Une identité est considérée comme non valide si et seulement si elle n'a pas été vérifiée et approuvée par cette autorité. Le processus de validation suit les étapes suivantes :

1. Vérification de la signature utilisateur : L'autorité contrôle l'authenticité de la signature numérique associée à l'identité soumise.

2. Génération d'un justificatif de vérification : Si la signature est valide, l'autorité émet un VC attestant de la légitimité de l'identité.

3. Enregistrement sur la blockchain : Le VC est ensuite stocké de manière immuable dans la blockchain via le compte de l'utilisateur concerné, garantissant ainsi sa traçabilité et son intégrité.

L'interface UI du système dispose de :

Barre latérale gauche

Elle offre un accès rapide aux principales fonctionnalités :

- Tableau de Bord : regroupe l'ensemble des fonctionnalités.
- Créer Identité : création d'une nouvelle identité numérique.
- Modifier Identité : Modification des identités existantes.
- Boîte de Réception : Consultation des identités reçues.

Zone supérieure principale

- Utilisateur connecté : Affichage de l'adresse publique Ethereum liée à MetaMask ce qui indique que l'utilisateur est connecté et que le contrat intelligent est actif.

- Identités totales : Nombre total d'identités créées par l'utilisateur courant.
- Dernière création : Informations sur la dernière identité créée.
- Actions rapides : Un bouton permettant de démarrer la création d'une nouvelle identité.

Zone centrale

- Identités créées : Liste des identités existantes.

Options supplémentaires

- Mode Clair/sombre : Bouton situé en bas à gauche permettant de basculer entre un thème clair et sombre pour améliorer le confort visuel selon les préférences de l'utilisateur.
- Barre de recherche : pour faciliter l'accès rapide aux identités ou autres données

Les figures ci-dessous présentent l'interface de l'application utilisateur et l'interface de l'autorité de vérification

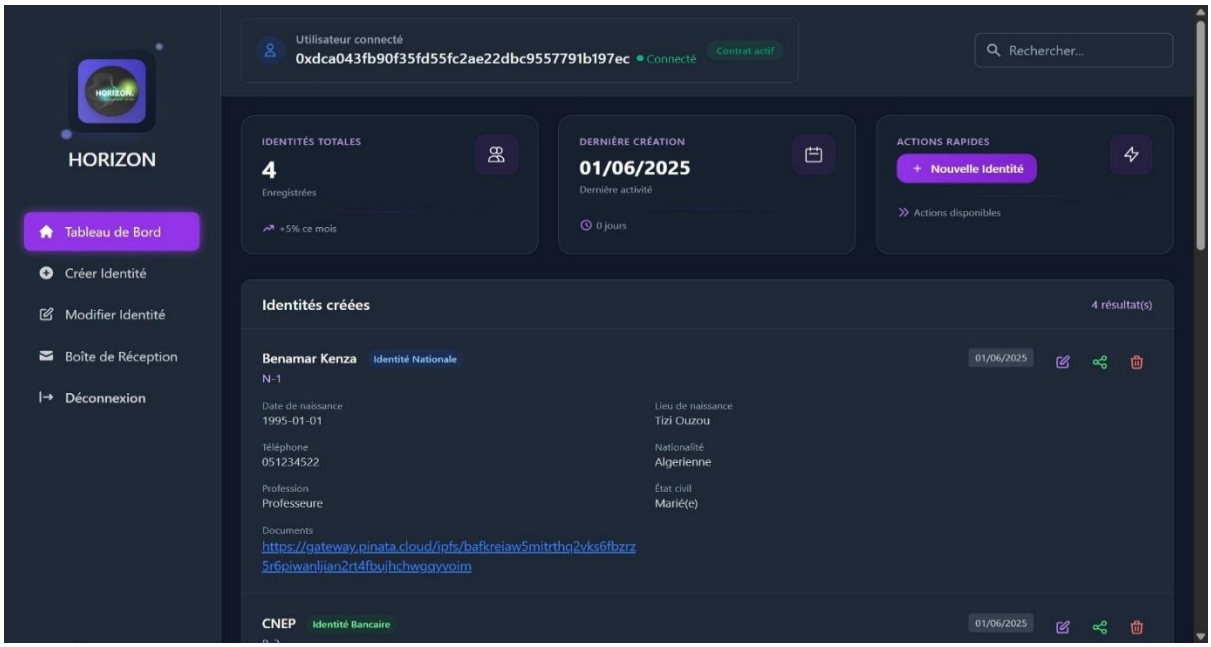

Figure III-7 Interface (Tableau de Bord)

L'interface de l'autorité de vérification est illustrée ci-dessous

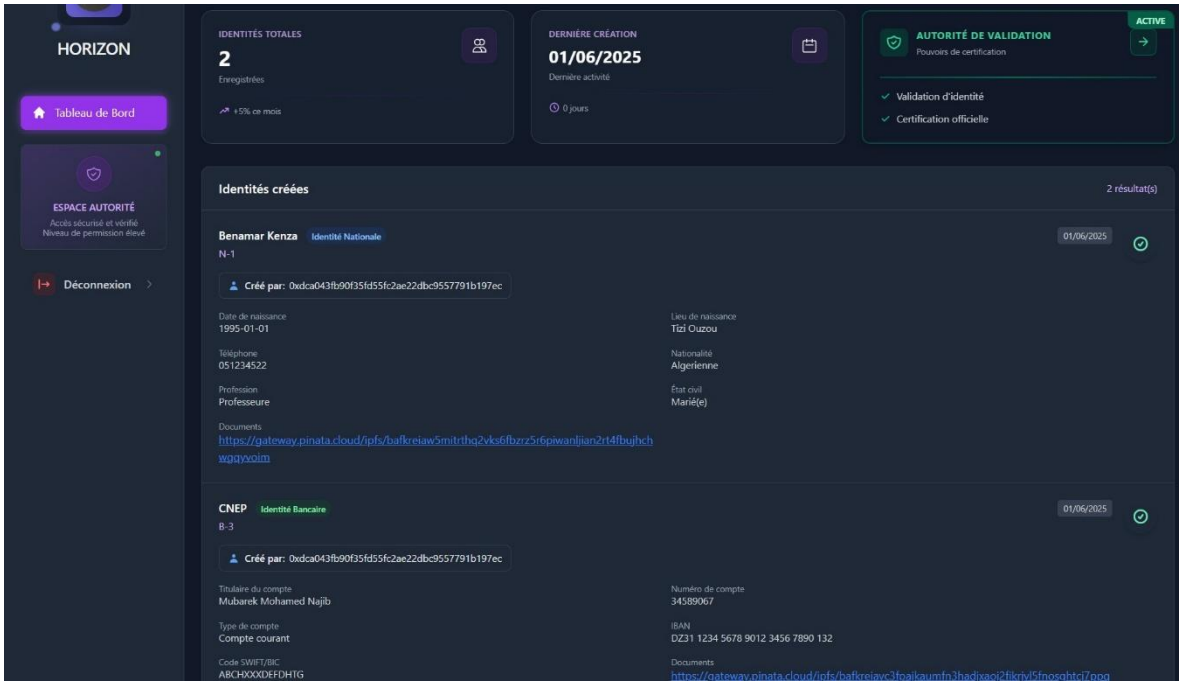

Figure III-8 Interface (Tableau de Bord « autorité de vérification »)

4.1.1. Création d'une nouvelle identité

Lors de la création d'une nouvelle identité l'utilisateur devrez commencer d'abord sélectionner le type d'identité (identité Nationale, bancaire, identité santé ou de ensuite remplir le formulaire d'information comme illustré dans la figure « Sélection Type d'identité » :

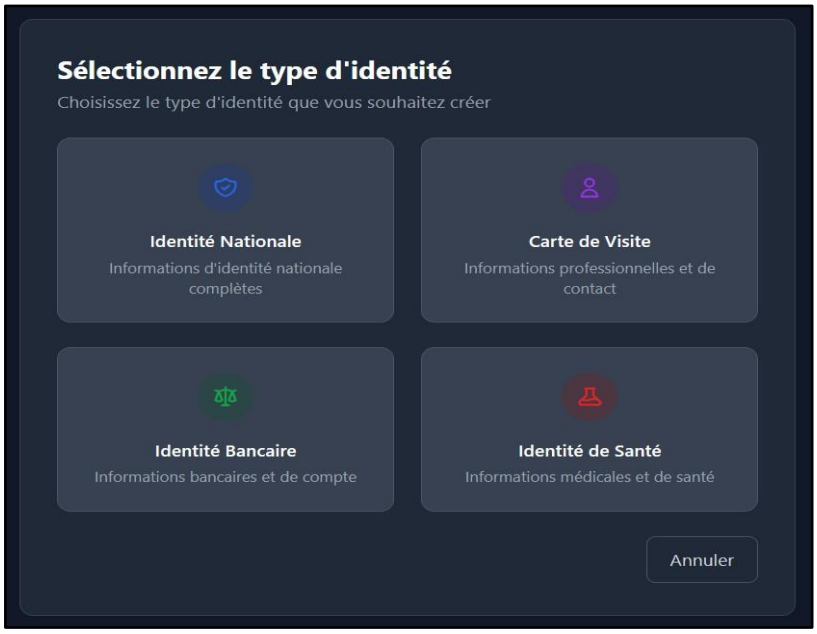

Figure III-9 Selection type d'identité

Et renseigner les différents champs du formulaire « Créer une Nouvelle Identité »et ajouter un fichier joint de type (ex :PDF, docx ... etc.) qui est stocké sur IPFS

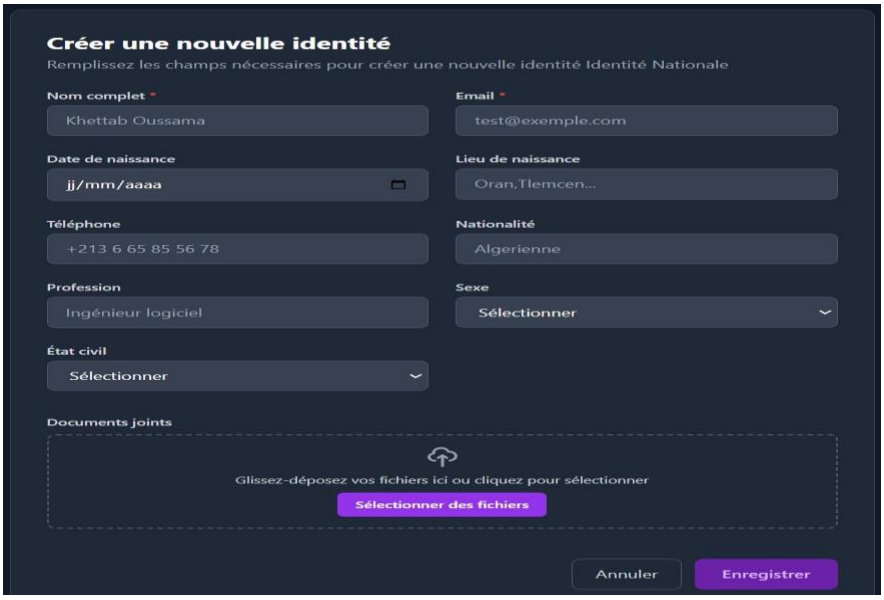

Figure III-10 Formulaire de Création d'une nouvelle identité

Dans le but de favoriser la décentralisation, la pérennité et l'accessibilité des documents portant sur les identités numériques, nous avons utilisé IPFS comme vaste système de stockage décentralisé. Pour interagir avec IPFS de manière simplifiée, nous avons choisi la plateforme Pinata, qui permet de téléverser, gérer et partager des fichiers sur le réseau IPFS. La figure ci-après montre l'interface de la plateforme Pinata, utilisée pour stocker des fichiers (une carte d'identité nationale, des statistiques médicales) qui seront rendus accessibles via des identifiants de contenus (CIDs).

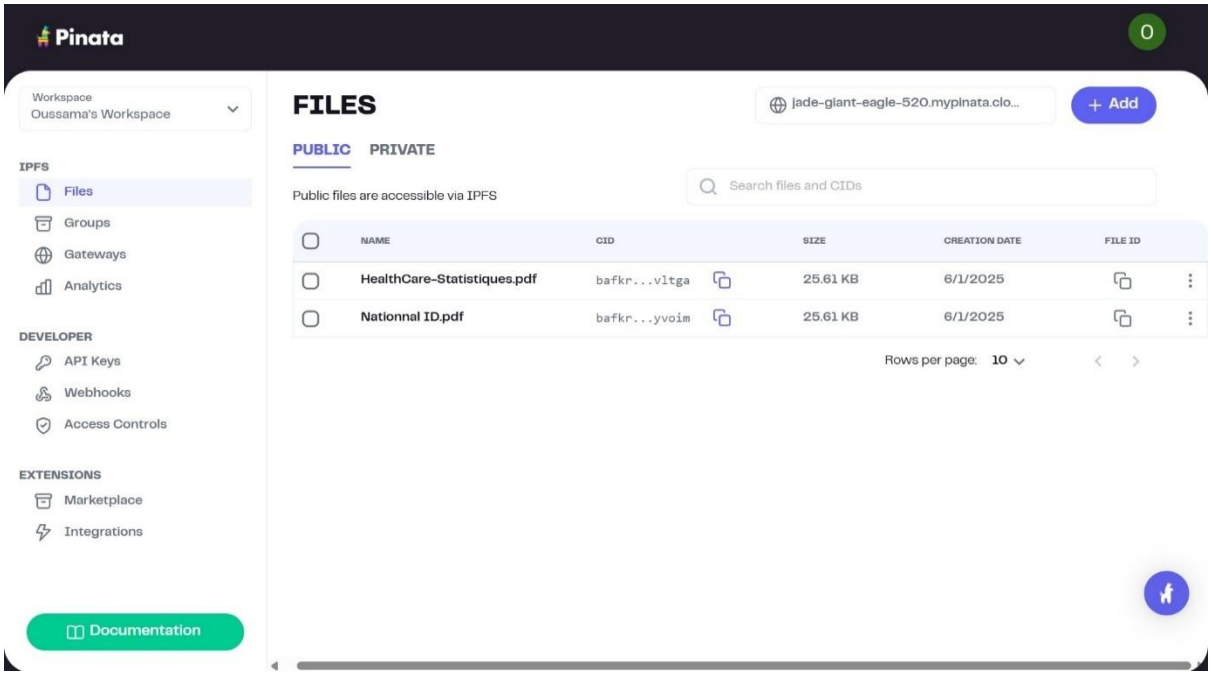

Figure III-11 les joints (ex : PDF) sur IPFS à l'aide de la plateforme Pinata

Une fois que l'utilisateur valide ses informations, une demande de confirmation de transaction apparaît. Celle-ci invite l'utilisateur à confirmer l'opération ou la confirmation contient la source du requête, Adresse du contrat intelligent avec lequel l'utilisateur interagit, comme indiqué dans la figure « demande de confirmation de transaction » :

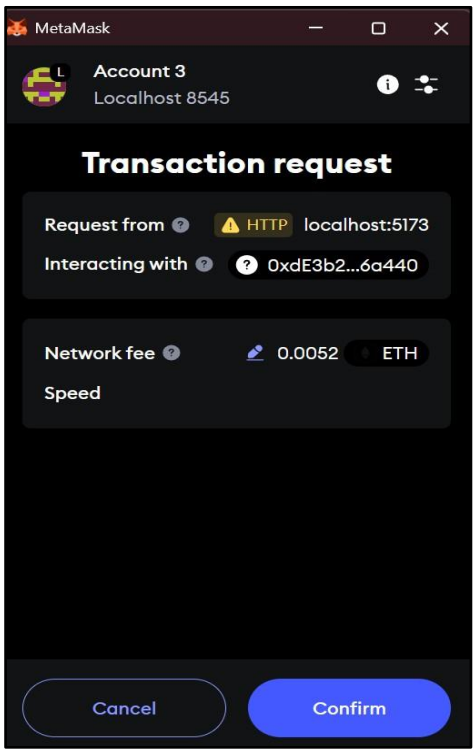

Figure III-12 demande de confirmation de transaction

Une notification s'affichera suite à la confirmation de la transaction par l'utilisateur.

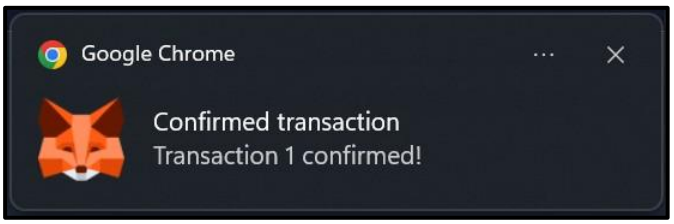

Figure III-13 Transaction "1" Confirmée

4.1.2. Modification d'une identité existante

En raison de l'immutabilité de la blockchain alors le processus de modification une identité vas générer une nouvelle identité reprenant les mêmes informations, mais avec les attributs mis à jour, comme illustré ci-dessous dans les figures :

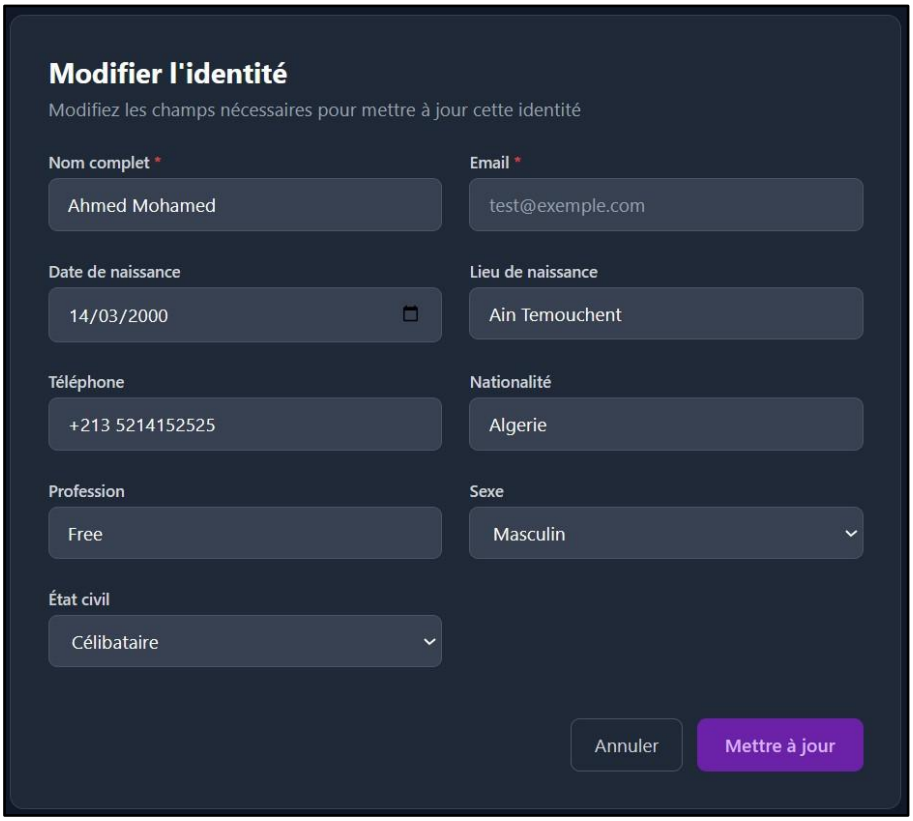

Figure III-14 Modification d'une identité existante

Une confirmation MetaMask apparait pour demander à l'utilisateur de confirmer la transaction

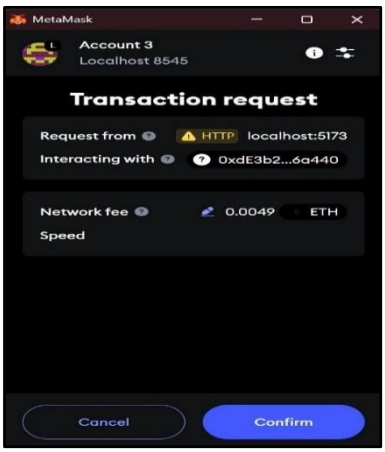

Figure III-15 Confirmation de modification

La figure ci-dessous illustre une identité nationale après modification, où les nouvelles informations sont immédiatement reflétées dans l'interface.

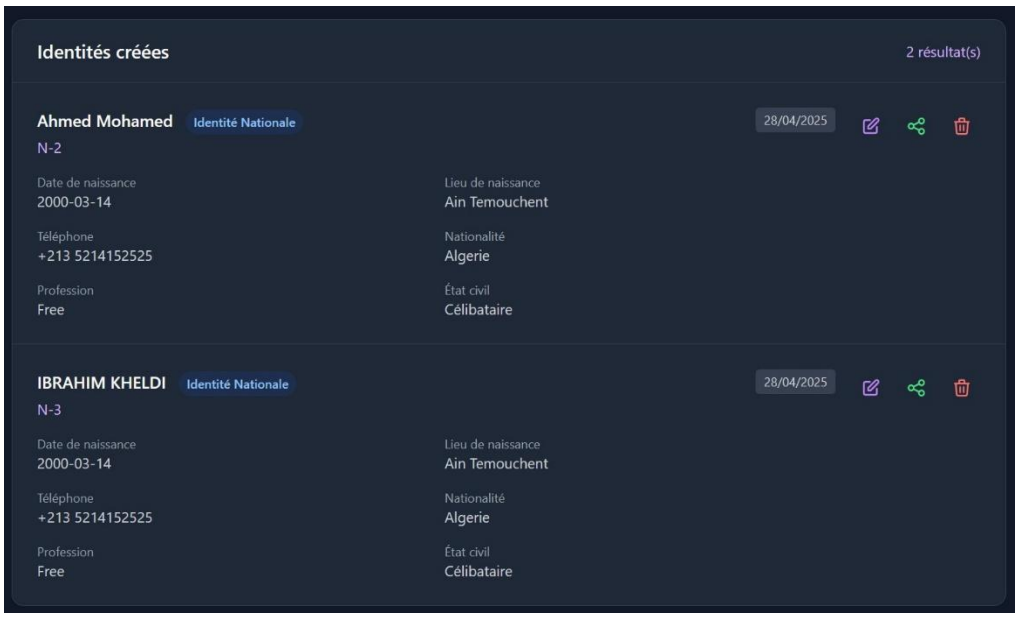

Figure III-16 identité Modifiée

4.1.3. Partage d'identité

Notre solution offre à l'utilisateur la possibilité de partager son identité tout en sélectionnant les attributs à partager avec l'extrémité, garantissant ainsi la confidentialité et la protection de sa vie privée. Le processus de partage d'identité est illustré dans la figure « Figure III-19 ».

L'utilisateur doit sélectionner les attributs à partager :

(Par exemple : nom complet, adresse e-mail, etc.) et renseigner l'adresse du destinataire, 0x901a9476de2adAA58Fe7b427Fd7Fd1CBD9f81402 Et confirmer la transaction

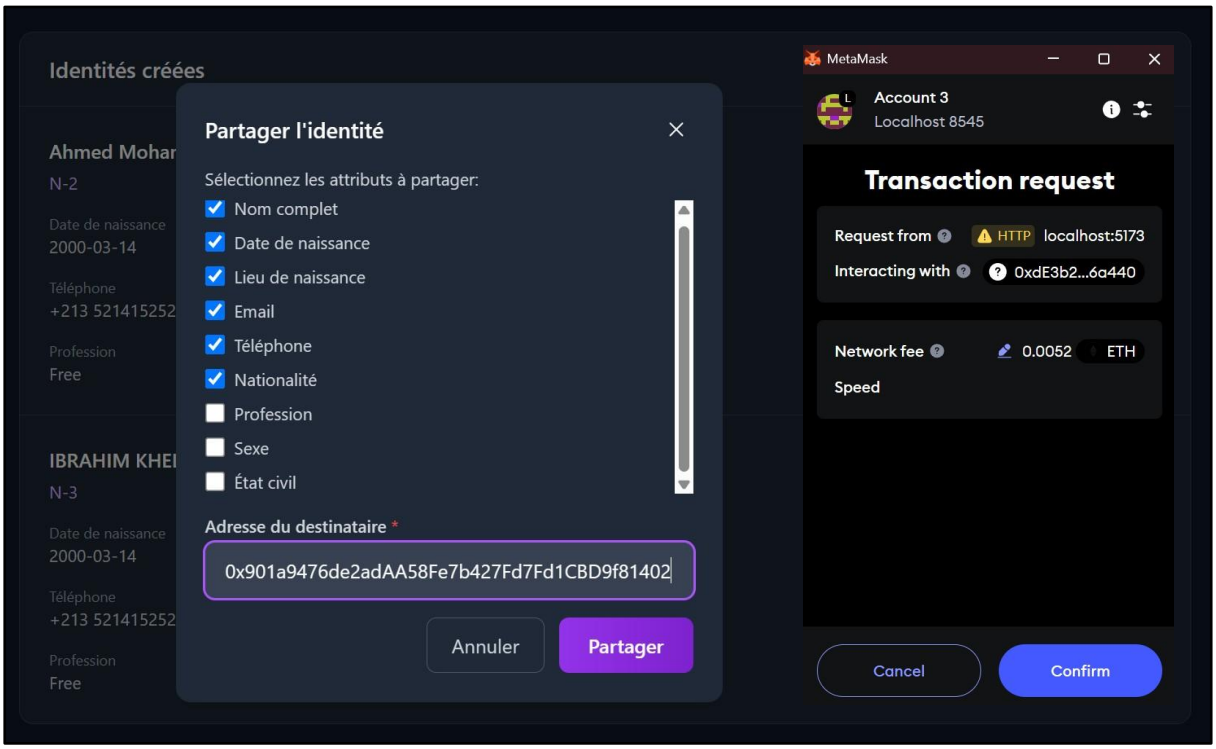

Figure III-17 Partage de l'identité

Après le partage de l'identité, une notification est générée pour confirmer à l'utilisateur la réussite de l'opération.

Une notification sera émise dans la boîte de réception du destinataire afin de l'informer de la réception de nouvelles identités, comme illustré dans la figure III-22.

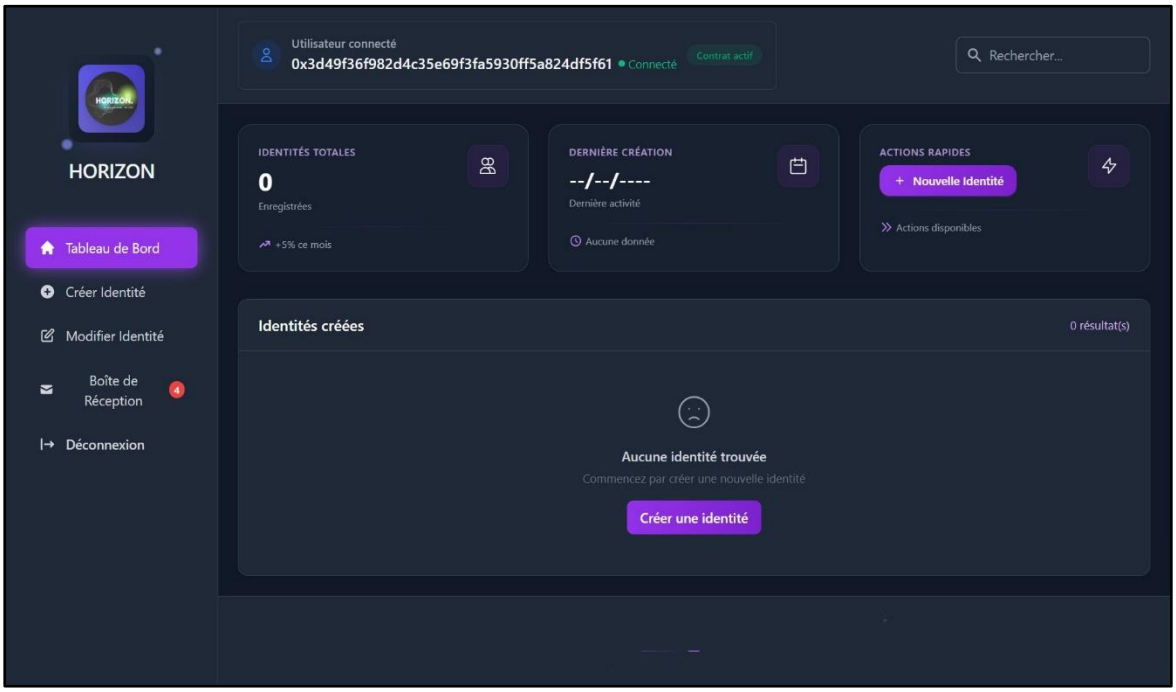

Figure III-18 Notification Boite de Réception

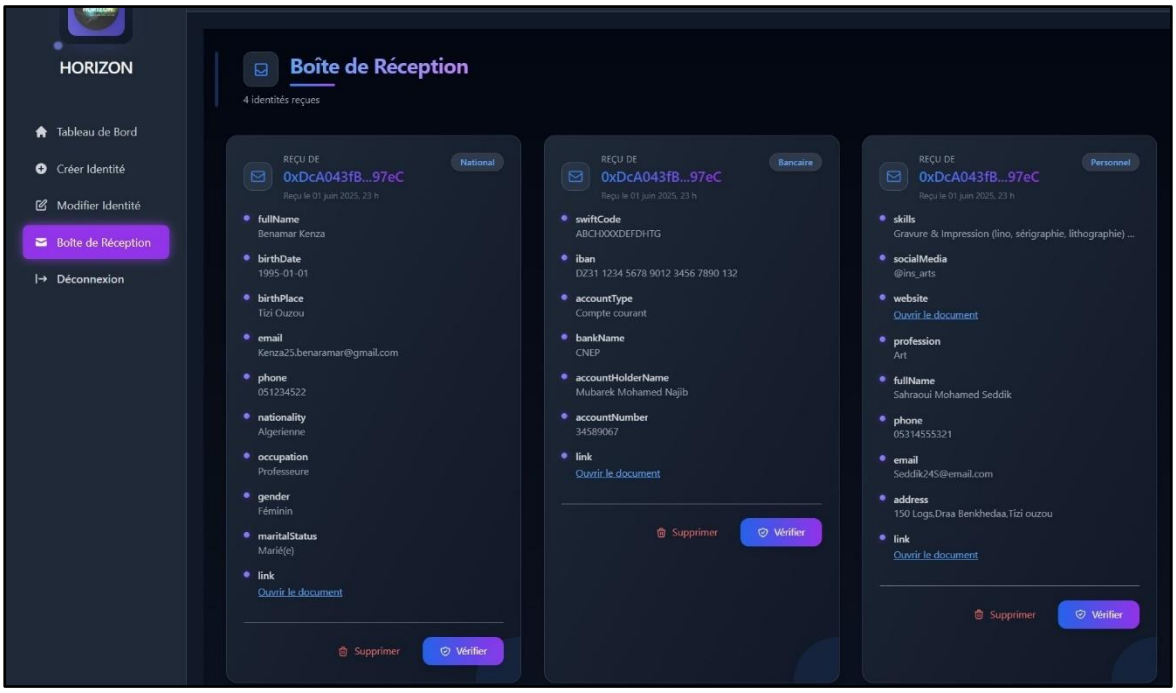

Figure III-19 identités reçues

4.1.4. Vérification de l'identité

Étant donné que notre solution intègre un mécanisme de partage d'identité, il est essentiel qu'elle offre également un mécanisme de vérification des identités reçues

La figure III-24 illustre l'identité partagée de l'adresse expéditrice vers l'adresse destinataire, comme résumé ci-dessous.

L'adresse expéditeur	*L'adresse destinataire*
`0xDcA043fB90F35fD55fC2Ae22DBc9557791b197eC`	`0x3d49f36f982d4C35e69F3fa5930ff5a824Df5F61`

Tableau III-1 vérification d'une identité entre un expéditeur et un destinataire

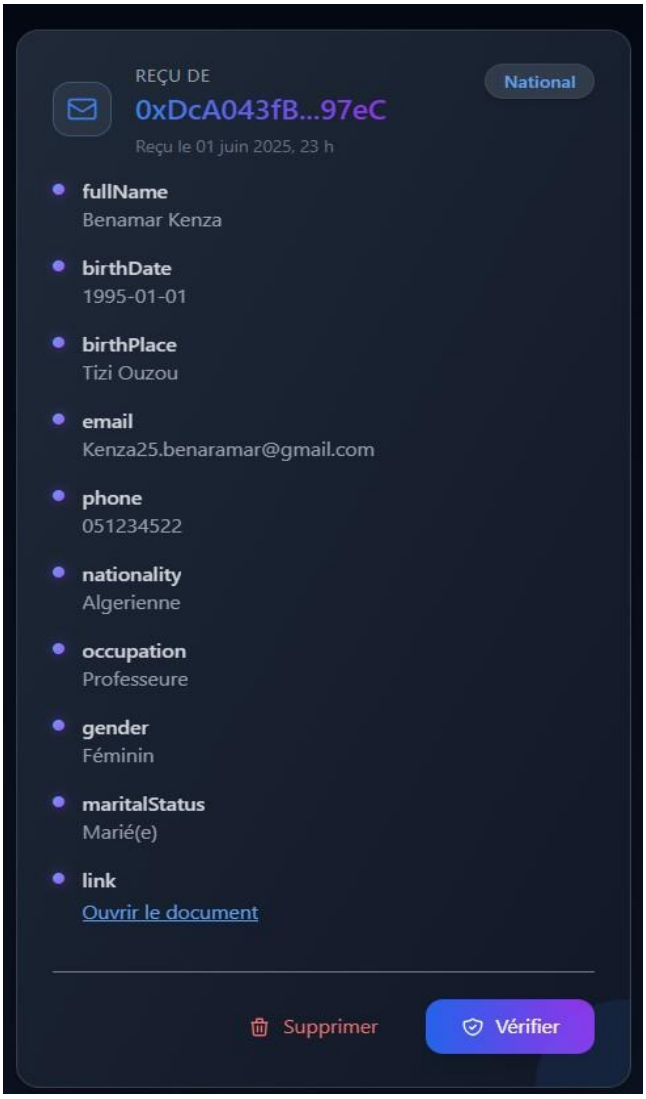

Figure III-21 Notification de réception d'une identité nationale dans la boîte de réception du destinataire

Une fois l'identité vérifiée et la transaction confirmée, une notification est envoyée à l'utilisateur destinataire pour l'informer du succès de l'opération.

La figure met en évidence que l'identité a été vérifiée et validée avec succès

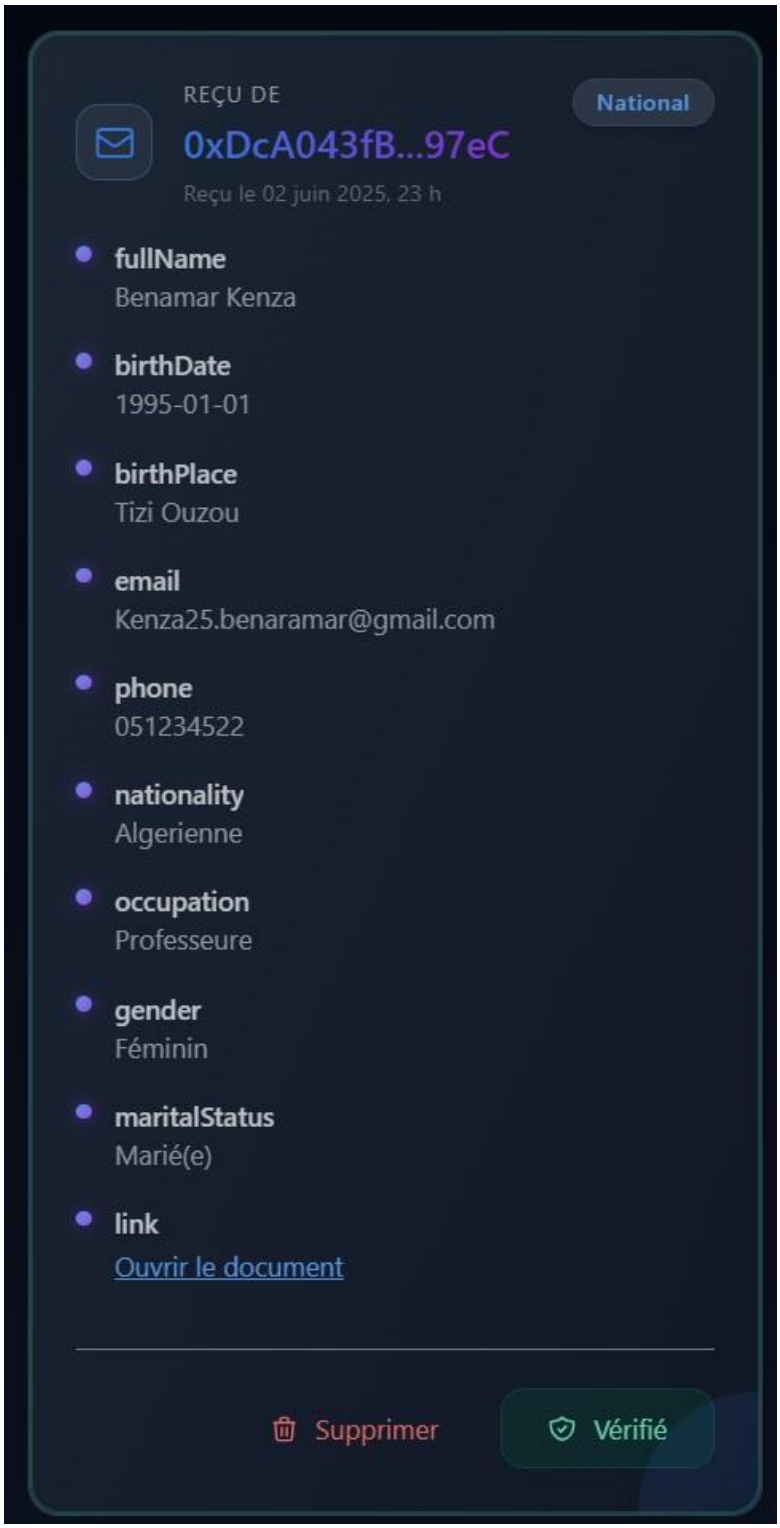

Figure III-22 Identité Vérifié2

4.2. Back-end

4.2.1. Configuration Truffle

`truffle-config.js` est le fichier de configuration qui gère le déploiement et la compilation des smart contracts., Il définit tous les paramètres nécessaires pour que Truffle interagir avec la blockchain, compiler les contrats, et migrer les déploiements.

```
MODULE.EXPORTS = {
  NETWORKS: {
    DEVELOPMENT: {
      HOST: "127.0.0.1",     // ADRESSE DE GANACHE
      PORT: 7545,            // PORT PAR DEFAUT GANACHE
GUI
      NETWORK_ID: "*",       // ACCEPTE N'IMPORTE QUEL
ID RESEAU
    },
  },

  // SPECIFIE LE COMPILATEUR SOLIDITY
  COMPILERS: {
    SOLC: {
      VERSION: "0.8.0",    //LA VERSION SMART CONTRACT
      SETTINGS: {
        OPTIMIZER: {
          ENABLED: TRUE,
          RUNS: 100}}},},};
```

Figure III-23 Configuration truffle-config.js

Truffle compile : Pour compiler tous les smart contracts Solidity (.sol) présents dans le projet Truffle et permet de générer le fichier `DIDManager.json` dans le répertoire /build/contracts

Truffle migrate --network development : utilisée pour déployer (ou migrer) les smart contracts compilés vers une blockchain (par exemple Ganache), le réseau de développement local.

4.2.2. Les Codes des Smart Contracts

- La fonction createNationalID()
- La fonction sendDID()
- La fonction getInbox()
- La fonction verifyReceivedDID()

Lorsqu'un utilisateur confirme la création d'une identité, la fonction handleCreateIdentity () envoie une transaction à MetaMask, qui exécute ensuite la fonction createNationalID() du smart contract pour stocker l'identité sur la blockchain.

La partie de code Solidity, issu du fichier `DID.sol` et chargé de la création d'une identité nationale, est présenté ci-dessous.

```
// CREATE
FUNCTION CREATENATIONALID(
        STRING MEMORY FULLNAME,
        STRING MEMORY EMAIL,
        STRING MEMORY BIRTHDATE,
        STRING MEMORY BIRTHPLACE,
        STRING MEMORY PHONE,
        STRING MEMORY NATIONALITY,
        STRING MEMORY PROFESSION,
        STRING MEMORY GENDER,
        STRING MEMORY MARITALSTATUS,
        STRING MEMORY LINK
     ) PUBLIC {
        NATIONALIDS[MSG.SENDER].PUSH(NATIONALID(
            GENERATEID("N-"),
            FULLNAME,EMAIL, BIRTHDATE, BIRTHPLACE, PHONE,
            NATIONALITY, PROFESSION, GENDER, MARITALSTATUS,LINK
        ));
        EMIT NATIONALIDCREATED(MSG.SENDER);
    }
```

Figure III-24 Fonction Solidity de creation d'une identité Nationnal

La partie du code Solidity correspondant à la modification de l'identité nationale, via la fonction `modifyNationalID`, est illustrée ci-dessous

```
// MODIFY (ID UNCHANGED)
    FUNCTION MODIFYNATIONALID(
        UINT INDEX,
        STRING MEMORY FULLNAME,
        STRING MEMORY BIRTHDATE,
        STRING MEMORY BIRTHPLACE,
        STRING MEMORY PHONE,
        STRING MEMORY NATIONALITY,
        STRING MEMORY PROFESSION,
        STRING MEMORY GENDER,
        STRING MEMORY MARITALSTATUS
    ) PUBLIC {
        REQUIRE(INDEX < NATIONALIDS[MSG.SENDER].LENGTH, "INVALID INDEX");
        NATIONALID STORAGE ID = NATIONALIDS[MSG.SENDER][INDEX];
        ID.FULLNAME = FULLNAME;
        ID.BIRTHDATE = BIRTHDATE;
        ID.BIRTHPLACE = BIRTHPLACE;
        ID.PHONE = PHONE;
        ID.NATIONALITY = NATIONALITY;
        ID.PROFESSION = PROFESSION;
        ID.GENDER = GENDER;
        ID.MARITALSTATUS = MARITALSTATUS;
        EMIT NATIONALIDMODIFIED(MSG.SENDER, INDEX);
    }
```

Figure III-25 Fonction Solidity de Modification d'une identité existante

La figure III-26 présente la fonction en Contrat Intelligent sendDID Solidity, qui permet d'envoyer une identité sélectionnée d'un utilisateur vers un autre utilisateur via la blockchain.

Objectif de la fonction :

- Envoyer une identité partagée vers le destinataire
- Enregistrer l'identité reçue dans la boîte de réception (`inbox`) du destinataire.
- Garantir que les données partagées soient valides.

```
// SEND + INBOX
FUNCTION SENDDID(
ADDRESS RECIPIENT,
DIDTYPE DIDTYPE,
STRING[] MEMORY SELECTEDATTRIBUTES
) PUBLIC {
REQUIRE(RECIPIENT != ADDRESS(0), "INVALID RECIPIENT");
REQUIRE(SELECTEDATTRIBUTES.LENGTH > 0, "MUST INCLUDE AT LEAST THE ID");

INBOX[RECIPIENT].PUSH(RECEIVEDDID({
SENDER: MSG.SENDER,
DIDTYPE: DIDTYPE,
SHAREDATTRIBUTES: SELECTEDATTRIBUTES,
TIMESTAMP: BLOCK.TIMESTAMP,
VERIFIED: FALSE
}));
EMIT DIDSENT(MSG.SENDER, RECIPIENT, DIDTYPE);
}
```

Figure III-26 Envoi d'identité vers un destinataire via la blockchain

La figure III-27 présente la fonction verifyReceivedDID en Solidity, qui permet à un utilisateur de vérifier une identité qu'il a reçue, ou les objectifs de la fonction sont :

- Permettre au destinataire d'une identité partagée de confirmer que l'identité a été examinée et validée.
- Marquer l'identité comme vérifiée dans la structure de stockage (inbox) sur la blockchain.
- Émettre un événement pour notifier cette action.

```
FUNCTION VERIFYRECEIVEDDID(UINT256 INDEX) PUBLIC {
        REQUIRE(INDEX < INBOX[MSG.SENDER].LENGTH, "INVALID INDEX");
        INBOX[MSG.SENDER][INDEX].VERIFIED = TRUE;
        EMIT DIDVERIFIED(MSG.SENDER, INDEX);
    }
```

Figure III-27 Fonction Solidty pour vérifier une identité partagée

Ainsi que la fonction qui permet de charger les identités reçues, illustrée ci-dessous.

```
fUNCTION GETINBOX() PUBLIC VIEW RETURNS (RECEIVEDDID[]
MEMORY) {
        RETURN INBOX[MSG.SENDER];
    }
```

Figure III-28 Fonction pour la récupération de la boîte de réception des identités

Suite à la validation de la transaction, le solde ETH du compte « 0xB42243633B754Fb046692DFbcF112957B329Cf15 » a été réduit, comme le montre la figure intitulée « Compte 0xB42243633B754Fb046692DFbcF112957B329Cf15 ».

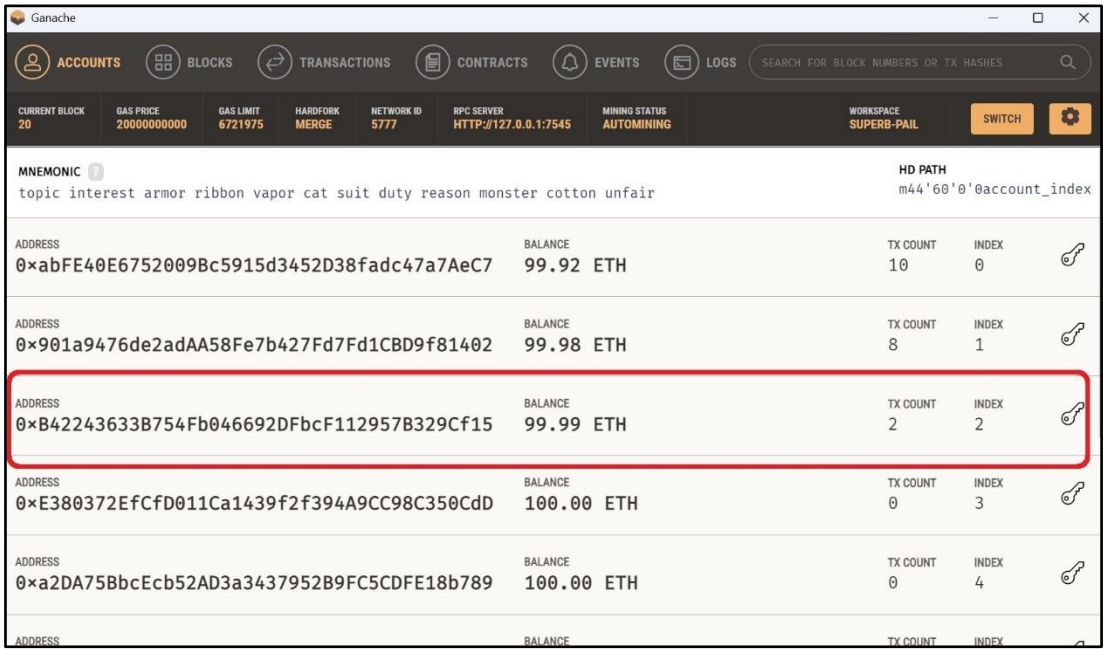

Figure III-29 0xB42243633B754Fb046692DFbcF112957B329Cf15

4.2.3. Signature Numérique

Notre système intègre un mécanisme de signature numérique à plusieurs niveaux. Lors de la création d'une identité est d'abord signée par l'utilisateur en premier lieu puis sera signé par l'autorité de vérification lors de sa validation, lorsqu'un destinataire vérifiant reçoit cette identité, le vérifieur vérifie d'abord la signature utilisateur ensuite vérifie si existe-'il un VC correspondant à cette identité ensuite il vérifié la signature du VC ; ce mécanisme permet de garantir l'intégrité et l'authenticité des identités échangées au sein du système.

L'implémentation de la signature numérique est illustrée dans les figures ci-dessous

Pour l'algorithme de signature numérique eD25519 Le nombre 25519 dans Ed25519 fait référence à une courbe elliptique spécifique utilisée dans la cryptographie, appelée Curve25519.

- Pourquoi "25519" ?

Le nombre 25519 dans "Ed25519" est le nombre premier utilisé pour définir le corps fini sur lequel la courbe est construite : $P=2^{255}-19$

La fonction generateKeyPair () dans la figure III-30 permet de générer une paire de clés cryptographiques (clé privée et clé publique) en utilisant la courbe elliptique ed25519 :

```
IMPORT * AS ED FROM '@NOBLE/ED25519';
IMPORT { BYTESTOHEX, HEXTOBYTES } FROM "@NOBLE/HASHES/UTILS";
EXPORT ASYNC FUNCTION GENERATEKEYPAIR() {
    CONST PRIVATEKEY = ED.UTILS.RANDOMPRIVATEKEY(); // UINT8ARRAY(32)
    CONST PUBLICKEY = AWAIT ED.GETPUBLICKEYASYNC(PRIVATEKEY); // UINT8ARRAY(32)
    CONST PRIVATEKEYHEX = BYTESTOHEX(PRIVATEKEY);
    CONST PUBLICKEYHEX = BYTESTOHEX(PUBLICKEY);
    CONSOLE.LOG("🔐 PRIVATE KEY (HEX):", PRIVATEKEYHEX);
    CONSOLE.LOG("🟢 PUBLIC KEY  (HEX):", PUBLICKEYHEX);
    RETURN {
      PRIVATEKEY: PRIVATEKEYHEX,
      PUBLICKEY: PUBLICKEYHEX
    };
}
```

Figure III-30 génération une paire de clés : privé et publique

La fonction signMessage () permet de générer une signature numérique à partir d'un message et d'une clé privée. Elle prend en entrée un message, qui peut être soit un objet JavaScript, soit un tableau d'octets (Uint8Array), ainsi qu'une clé privée sous forme hexadécimale. Si le message est un objet, il est d'abord converti en chaîne JSON, puis encodé en octets à l'aide de TextEncoder. Si c'est déjà un tableau d'octets, il est utilisé tel quel.

```
EXPORT ASYNC FUNCTION SIGNMESSAGE(MESSAGE: OBJECT | UINT8ARRAY, PRIVATEKEYHEX: STRING) {
    LET MESSAGEBYTES: UINT8ARRAY;

    IF (MESSAGE INSTANCEOF UINT8ARRAY) {
      MESSAGEBYTES = MESSAGE;
    } ELSE {
      // SERIALISER L'OBJET EN JSON STRING PUIS ENCODER EN BYTES
      CONST JSONSTRING = JSON.STRINGIFY(MESSAGE);
      MESSAGEBYTES = NEW TEXTENCODER().ENCODE(JSONSTRING);
    }
    CONST PRIVATEKEYBYTES = HEXTOBYTES(PRIVATEKEYHEX);

    CONST SIGNATURE = AWAIT ED.SIGNASYNC(MESSAGEBYTES, PRIVATEKEYBYTES);
    RETURN BYTESTOHEX(SIGNATURE);
  }
```

Figure III-31 signMessage – Signature d'un message à l'aide d'une clé privée Ed25519

La fonction verifySignature permet de vérifier l'authenticité d'un message à l'aide d'une signature numérique et d'une clé publique, en utilisant l'algorithme EdCDSA

Le message est d'abord converti en tableau d'octets si ce n'est pas déjà le cas, en le sérialisant au format JSON puis en l'encodant avec TextEncoder. Ensuite, les formats hexadécimaux de la signature et de la clé publique sont convertis en tableaux d'octets à l'aide de la fonction hexToBytes.

La figure III-31 ci-dessous présente la fonction verifySignature en TypeScript

```
EXPORT ASYNC FUNCTION VERIFYSIGNATURE (
    MESSAGE: STRING | UINT8ARRAY,
    SIGNATUREHEX: STRING,
    PUBLICKEYHEX: STRING
  ) {
    CONST MESSAGEBYTES = TYPEOF MESSAGE === "STRING"
      ? NEW TEXTENCODER().ENCODE(MESSAGE)
      : MESSAGE;
    CONST SIGNATUREBYTES = HEXTOBYTES(SIGNATUREHEX);
    CONST PUBLICKEYBYTES = HEXTOBYTES(PUBLICKEYHEX);

    CONST ISVALID = AWAIT ED.VERIFYASYNC(SIGNATUREBYTES, MESSAGEBYTES, PUBLICKEYBYTES);
    RETURN ISVALID; // TRUE OU FALSE
  }
```

Figure III-32 verifySignature – Vérification d'une signature numérique avec Ed25519

Afin d'apporter une meilleure compréhension, l'exemple suivant illustre le mécanisme de signature numérique d'une identité existante crée par l'utilisateur avec l'adresse « 0xDcA043fB90F35fD55fC2Ae22DBc9557791b197eC » en utilisant l'algorithme cryptographique Ed25519.

Clé privée (hexadécimal) :

5d453ce73627e98e8b10078f6a731a9662e298cf7289fca85c5909a2c5565bd1

Cette clé est secrète et sert à signer les données.

Clé publique (hexadécimal) :

b0b98cbf43a27b9a9f1e7120e2efffc89c7a51b3d16bdef4f74f4ce4ee3a8c0c

Cette clé est partagée publiquement et permet de vérifier la signature.

Signature générée (hexadécimal) [512 bits] :

7dc6bdab403d33eff4e077d24c75b677fe6b2f9744d86d81c99858ee5df8abdef3d9581dd86c9d6c715e35f690c7407df40e6438e75836a6a63c420b14199707

4.2.4. Chiffrement

Un chiffrement est utilisé pour garantir la confidentialité, l'intégrité des données personnelles lors de l'envoi au sein notre application. En outre, même si la blockchain assure l'immutabilité et la traçabilité des transactions, le chiffrement reste nécessaire. En raison de RGPD (Règlement Général sur la Protection des Données), sécurité et bonnes pratiques professionnelles. L'algorithme de chiffrement utilisé : AES-256 avec le mode opératoire GCM avec une taille de clé 32 et un vecteur d'initialisation IV 12 et à l'aide d'un mot de passe.

La fonction ci-dessous permet de générer une clé dérivé d'un mot de passe pour l'utiliser dans le chiffrement :

```
// GENERE UNE CLE AES-256-GCM A PARTIR D'UN MOT DE PASSE
EXPORT ASYNC FUNCTION DERIVEKEY(PASSWORD: STRING : UINT8ARRAY): PROMISE<CRYPTOKEY> {
    CONST ENC = NEW TEXTENCODER();
    CONST KEYMATERIAL = AWAIT WINDOW.CRYPTO.SUBTLE.IMPORTKEY(
        "RAW",
        ENC.ENCODE(PASSWORD),
        "PBKDF2",
        FALSE,
        ["DERIVEKEY"]
    );
    RETURN WINDOW.CRYPTO.SUBTLE.DERIVEKEY(
        {
            NAME: "PBKDF2",
            ITERATIONS: 100_000,
            HASH: "SHA-256"
        },
        KEYMATERIAL,
        {
            NAME: "AES-GCM",
            LENGTH: 256
        },
        FALSE,
        ["ENCRYPT", "DECRYPT"]
    );
}
```

Figure III-33 Génération d'une clé AES-256-GCM à partir d'un mot de passe

Le code ci-dessous illustre la fonction `encryptData`, qui utilise l'algorithme AES-GCM pour chiffrer des données à l'aide d'un mot de passe. Le processus inclut la génération aléatoire d'un vecteur d'initialisation (`iv`), garantissant ainsi la robustesse et l'unicité du chiffrement pour chaque opération.

```
// CHIFFRE LES DONNEES ET RETOURNE UN PAQUET COMBINE : IV + CIPHER
EXPORT ASYNC FUNCTION ENCRYPTDATA (DATA: ANY, PASSWORD: STRING): PROMISE<STRING> {
    CONST ENC = NEW TEXTENCODER();
    CONST IV = WINDOW.CRYPTO.GETRANDOMVALUES(NEW UINT8ARRAY(IV_LENGTH));
    CONST KEY = AWAIT DERIVEKEY(PASSWORD);
    CONST ENCODED = ENC.ENCODE(JSON.STRINGIFY(DATA));
    CONST CIPHERBUFFER = AWAIT WINDOW.CRYPTO.SUBTLE.ENCRYPT(
        {NAME: "AES-GCM", IV },
        KEY,
        ENCODED
    );
```

La fonction de déchiffrement correspondante, représentée dans la figure III-34, permet de reconstituer les données originales en extrayant le vecteur d'initialisation (IV) et en appliquant l'algorithme AES-256-GCM à l'aide de la clé partagée. Elle garantit que seules les parties autorisées peuvent accéder aux informations sensibles.

```
// Déchiffre les données à partir du paquet
export async function decryptData(encryptedData: string, password: string): Promise<any> {
    const [ivHex, cipherHex] = encryptedData.split(":");
    const iv = new Uint8Array(ivHex.match(/.{1,2}/g)!.map(b => parseInt(b, 16)));
    const cipher = new Uint8Array(cipherHex.match(/.{1,2}/g)!.map(b => parseInt(b, 16)));
    const key = await deriveKey(password);
    const plainBuffer = await window.crypto.subtle.decrypt(
        { name: "AES-GCM", iv },
        key,
        cipher
    );
    const dec = new TextDecoder();
    return JSON.parse(dec.decode(plainBuffer));
}
```

Figure III-35 Fonction de déchiffrement des données chiffrées avec AES-256-GCM à l'aide d'un mot de passe

Cette Partie du code TypeScript permet de convertir les données en hexadécimal pour le transport

```
    // Encode en hex pour transport : iv:cipher
    const ivHex = Array.from(iv).map(b => b.toString(16).padStart(2, "0")).join("");
    const cipherHex = Array.from(new Uint8Array(cipherBuffer)).map(b => b.toString(16).padStart(2,
"0")).join("");

    return `${ivHex}:${cipherHex}`;
}
```

Figure III-36 Conversion des données (IV, texte chiffré) en chaîne hexadécimale pour l'export

5. Analyse et Critique

L'analyse, la conception et le développement de notre solution représentent une avancée significative par rapport aux systèmes de gestion traditionnels. Notre approche offre des avantages en termes de confidentialité, d'intégrité des données et de résilience face aux points de défaillance uniques SPOF. En outre, certaines limitations et améliorations doivent être identifiées afin d'améliorer davantage l'efficacité de la solution proposée.

5.1. Points forts

- L'utilisation de la blockchain garantit la décentralisation et l'immutabilité des données, ainsi que la résistance face aux pannes et aux attaques liées aux systèmes traditionnels.
- L'autonomie d'utilisateur qui bénéficie d'un contrôle accru sur ses données personnels et son identité numérique.
- L'intégration de Wallet (tel que MetaMask), et le concept des VC ainsi que la signature numérique multiple, améliore la sécurité utilisateur-système et facilite la vérification de l'identité.

5.2. Limites

Malgré les avancées apportées par notre Solution de la gestion décentralisée des identités, certaines limites subsistent :

- L'application repose sur des portefeuilles comme MetaMask et une connectivité stable au réseau blockchain, ce qui peut restreindre son accessibilité dans certains contextes.

- L'usage de concepts tels que les DIDs, les VC, ou la signature cryptographique peut représenter une barrière pour des utilisateurs sans formation technique.

- Bien que conforme à certaines normes (comme GDPR ou W3C DID), l'application nécessite d'être adaptée à tous les cadres légaux internationaux concernant l'identité numérique.

5.3. Point d'Amélioration

- Optimisation de la latence lors de des interactions fréquentes avec la blockchain.
- Renforcement de la sécurité, notamment par l'ajout de mécanisme de détection et de prévention des attaques.
- Architecture plus modulaire pour l'intégration de nouvelles fonctionnalités et adaptation aux différents cas d'utilisation.

5.4. Critique globale

Bien que notre solution repose sur une base technique et théorique solide, et que les différents tests réalisés valident sa performance et sa fiabilité, certaines fonctionnalités supplémentaires et améliorations en termes d'ergonomie pourraient être ajoutées pour enrichir l'expérience utilisateur vis-à-vis le système.

Dans l'ensemble, la solution proposée répond bien aux besoins de la gestion décentralisée de l'identité tout en offrant une marge pour de futures améliorations.

Le tableau ci-dessous synthétise les principales caractéristiques d'un système SSI et précise si celles-ci sont vérifiées dans notre solution.

Propriété du system SSI	Vérification
Décentralisation	Vérifié
Disponibilité	Vérifié
Souveraineté numérique	Vérifié
Transparence	Vérifié
Indépendance	Vérifié
Interopérabilité	Vérifié
Compatibilité	Vérifié
Consentement	Vérifié

Tableau III-2 propriétés SSI clés dans notre système

6. Conclusion

Ce chapitre a présenté l'analyse, la conception et l'implémentation pratique de notre solution de gestion décentralisée des identités. Dans un premier temps, nous avons présenté les divers outils utilisés pour répondre aux besoins identifiés. Dans un second temps, nous avons détaillé la conception logicielle, en mettant en lumière l'intégration entre le front-end et le back-end, le rôle des divers acteurs ainsi que les illustrations des résultats obtenus qui démontrent le bon fonctionnement de la solution proposée.

L'implémentation de différentes fonctionnalités fondamentales, telles que la création des identités, la modification d'une identité existante, le partagedes identités ainsi que la vérification de leur validité, témoigne de la facilité d'usage, la fiabilité et de l'efficacité de notre solution.

Ainsi, les tests réalisés assurent que la solution répond bien aux objectifs fixé préalablement ce qui démontre sa pertinence face aux problèmes des systèmes traditionnels.

Conclusion générale

L'évolution constante du monde numérique fait de la gestion sécurisé et autonome de l'identité un enjeu majeur. L'avènement de la blockchain et du concept SSI offre une alternative idéale aux systèmes centralisés traditionnels et leurs vulnérabilités critiques ainsi que le manque de transparence. La gestion décentralisée des identités permet aux individus de reprendre le contrôle total de leurs données personnelles tout en garantissant la confidentialité, la transparence et l'intégrité.

Dans ce travail, nous avons exploré les fondements de l'identité décentralisée et de la blockchain, tout en présentant l'analyse, la conception et l'implémentation de notre solution applicative qui permet aux utilisateurs de gérer, partager et vérifier leurs identités de façon autonome.

Notre solution renforce la confidentialité et réduit les vulnérabilités et les risques SPOF, tout en garantissant la confidentialité grâce aux caractéristiques de la blockchain et aux techniques cryptographiques.

Cette étude ouvre la voie à de futures perspectives et explorations visant a améliorer l'interopérabilité de la gestion décentralisée de l'identité, son adoption à grande échelle et son intégration dans des divers domaines qui représente désormais un objectif atteignable grâce à la gestion décentralisée de l'identité (DID). Parmi les perspectives possibles :

Chiffrement des fichiers stockés sur IPFS : Actuellement, les fichiers partagés via IPFS sont stockés de manière décentralisée, mais sans chiffrement natif (en clair). Il serait pertinent d'intégrer un chiffrement symétrique avant la mise en ligne, afin d'assurer la confidentialité des données partagées, même en environnement public.

Attribution d'autorités spécifiques selon le type d'identité : Une perspective essentielle consiste à assigner à chaque type d'identité une autorité de vérification dédiée. Par exemple : l'identité nationale pourrait être délivrée et signée par l'autorité chargé de l'identité nationale.

Bibliographie

[1] R. Pinto, Decentralized identity explained "embrace decentralization for a more secure and empowering digital experience", 11 St Paul's Square, Birmingham,B3 1RB, UK: Packt Publishing Ltd., 2024.

[2] K. T. Elisa Bertino, Identity Management Concepts, Technologies, and Systems, 685 Canton Street, Norwood, MA 02062: ARTECH HOUSE, 2011 .

[3] F. Fossi, «Comprendre l'identité numérique, ses enjeux et les tendances,» 07 04 2025. [En ligne]. Available: https://www.digicommunicate.com/comprendre-l-identite-numerique-ses-enjeux-et-les-tendances. [Accès le 07 05 2025].

[4] e. Makaay, «Blockchain Identity Management: Beginner's Guide 2025,» 28 Février 2024. [En ligne]. Available: https://www.dock.io/post/blockchain-identity-management#what-is-identity-management. [Accès le 05 02 2025].

[5] «https://www.pingidentity.com/,» [En ligne]. Available: https://www.pingidentity.com/en/resources/identity-fundamentals/centralized-identity-management/what-is-centralized-identity-management.html. [Accès le 29 01 2025].

[6] A. team, «en.archipels.io/,» [En ligne]. Available: https://en.archipels.io/faq/digital-identity. [Accès le 30 01 2025].

[7] LE CLUSIF (Club de la Sécurité de l'Information Français), «Gestion des Identités,» PARIS, 2007.

[8] ISO, «iso.org/fr/securite-de-l-information/gestion-de-l-identite,» ISO, [En ligne]. Available: https://www.iso.org/fr/securite-de-l-information/gestion-de-l-identite. [Accès le 30 01 2025].

[9] Youzer, «Fédération d'identité - Glossaire IAM,» [En ligne]. Available: https://en.youzer.net/glossaire-iam/federation-identite. [Accès le 05 02 2025].

[10] SailPoint, «Qu'est-ce que la fédération d'identité,» [En ligne]. Available: https://www.sailpoint.com/fr/identity-library/what-is-federated-identity. [Accès le 05 02 2025].

[11] I. International, «Fédération d'identité,» [En ligne]. Available: https://www.ilex-international.com/fr/strategie-iam/federation-identite. [Accès le 05 02 2025].

[12] B. e. H. Boumriga, «La protection de la vie privée dans un système de gestion d'identité,» 2012. [En ligne]. Available: https://www.memoireonline.com/01/14/8693/m_La-protection-de-la-vie-privee-dans-un-systeme-de-gestion-d-identite19.html. [Accès le 05 02 2025].

[13] Ping Identity, «Decentralized Identity 101,» [En ligne]. Available: https://www.pingidentity.com/fr/lp/ac/pingone-neo/decentralized-identity-101.html. [Accès le 07 02 2025].

[14] Archipels, «On the Road to User-Centricity: Digital Identity in E-Wallet Era,» [En ligne]. Available: https://en.archipels.io/post/on-the-road-to-user-centricity-digital-identity-in-e-wallet-era. [Accès le 07 02 2025].

[15] Plisio, «Decentralized Identity (DID): The Pillar of Web3,» 13 02 2024. [En ligne]. Available: https://plisio.net/fr/blog/decentralized-identity-did-the-pillar-of-web3. [Accès le 07 02 2025].

[16] D. B. Jean Langlois-Berthelot, «Pour une identité numérique décentralisée,» 24 05 2023. [En ligne]. Available: https://www.journaldunet.com/web3/1521839-pour-une-identite-numerique-decentralisee/. [Accès le 07 02 2025].

[17] Archipels, «What is Decentralised Identity?,» [En ligne]. Available: https://en.archipels.io/faq/what-is-decentralised-identity. [Accès le 07 02 2025].

[18] IN Groupe, «Blockchain et souveraineté : les prémices d'une révolution de l'identité numérique,» [En ligne]. Available: https://ingroupe.com/fr/observatoire/blockchain-souverainete-premices-revolution-identite-numerique/. [Accès le 07 02 2025].

[19] Okta, «What is Decentralized Identity?,» 28 10 2024. [En ligne]. Available: https://www.okta.com/fr/blog/2021/01/what-is-decentralized-identity/. [Accès le 01 2025 2025].

[20] J. BERNARD, «Avantages de l'Identité Numérique Décentralisée. Épisode 3/5,» 24 07 2024. [En ligne]. Available: https://www.linkedin.com/pulse/avantages-de-lidentit%C3%A9-num%C3%A9rique-d%C3%A9centralis%C3%A9e-%C3%A9pisode-j%C3%A9r%C3%B4me-bernard-9dkye/. [Accès le 08 02 2025].

[21] D. R. Alex Preukschat, Self-Sovereign Identity, DECENTRALIZED DIGITAL IDENTITY, Shelter Island, NY 11964: Manning Publications Co. 20 Baldwin Road , 2021.

[22] I. Bashir, Mastering Blockchain, 2. Packt Publishing, Éd., 2017, p. 540 .

[23] A. I. Sankaa, M. Irfana, IanHuangb et R. C.C.Cheung, «A survey of break through in blockchain technology: Adoptions, applications, challenges and future research,» *ComputerCommunications*, n° %1169, p. 181, 2021.

[24] D. Lajeunesse, Buenos Aires, Argentina, 2025.

[25] J. Smith, Blockchain Development-Blockchain Applications, Kindle Edition, 2022, p. 243.

[26] B. Becher, «What Is a Blockchain Node and How Does It Work?,» 30 05 2024. [En ligne]. Available: https://builtin.com/blockchain/blockchain-node.

[27] Ledger, «Nœuds crypto : que sont les nœuds et comment fonctionnent-ils ?,» 12 05 2024. [En ligne]. Available: https://www.ledger.com/fr/academy/f-le-b-a-ba-de-la-blockchain/quest-ce-quun-noeud-pourquoi-en-faire-tourner-un. [Accès le 09 03 2025].

[28] A. M. Antonopoulos, Mastering Bitcoin, 2e éd., T. McGovern, Éd., O'Reilly Media, Inc., 1005 Gravenstein Highway North,CA, 2017, p. 374.

[29] E. Rasure, « Merkle Root (Cryptocurrency),» 24 08 2024. [En ligne]. Available: https://www.investopedia.com/terms/m/merkle-root-cryptocurrency.asp#:~:text=A%20Merkle%20root%20is%20the%20result%20of%20hashing%20transactions%20in,transactions%20stored%20in%20the%20block.. [Accès le 02 2025].

[30] J'aime les Startups, « La signature numérique et ce que les startups y gagnent,» 15 07 2020. [En ligne]. Available: https://www.jaimelesstartups.fr/la-signature-numerique-et-ce-que-les-startups-y-gagnent/. [Accès le 08 02 2025].

[31] Coin Academy, «Comprendre le fonctionnement des transactions sur la blockchain,» [En ligne]. Available: https://coinacademy.fr/academie/transaction-blockchain/?. [Accès le 03 03 2024].

[32] Ethereum.org, «Consensus Mechanisms,» 14 06 2024. [En ligne]. Available: https://ethereum.org/en/developers/docs/consensus-mechanisms/.

[33] crypto.com, «Qu'est-ce que la blockchain ?,» [En ligne]. Available: https://crypto.com/fr/university/what-is-blockchain-consensus. [Accès le 10 05 2025].

[34] J.-G. Dumas, T. Ariane et S. Varrette, «Qu'est-ce qu'un consensus ?,» *Les blockchains en 50 questions*, pp. 15-22, 07 06 2022.

[35] journalducoin.com, «Proof of Stake (PoS),» [En ligne]. Available: https://journalducoin.com/lexique/proof-of-stake/. [Accès le 08 05 2025].

[36] L. Lars, «Qu'est-ce que la preuve d'enjeu ou proof-of-stake (PoS) ?,» 18 02 2024. [En ligne]. Available: https://cryptoast.fr/qu-est-ce-que-pos-proof-of-stake/. [Accès le 08 05 2025].

[37] coinacademy, «Proof of Work (PoW) vs Proof of Stake (PoS) en crypto,» [En ligne]. Available: https://coinacademy.fr/academie/proof-of-work-pow-vs-proof-of-stake-pos/. [Accès le 10 05 2025].

[38] Binance Academy, «Proof of Authority Explained,» 17 08 2023. [En ligne]. Available: https://academy.binance.com/en/articles/proof-of-authority-explained.

[39] M. Clark, «blockchainreporter,» 05 05 2024. [En ligne]. Available: https://blockchainreporter.net/understanding-proof-of-authority-a-reliable-consensus-mechanism-in-blockchain-technology/. [Accès le 10 05 2025].

[40] N. E. Rharbi, H. H. Abdelali et A. O. Izem, «A Comparative Study of the Recent Blockchain,» 2023.

[41] Coinbase, «Qu'est-ce que le minage ?,» [En ligne]. Available: https://www.coinbase.com/fr-ca/learn/crypto-basics/what-is-mining. [Accès le 22 02 2025].

[42] Bitpanda Academy, «En quoi consiste le mining de bitcoins ?,» [En ligne]. Available: https://www.bitpanda.com/academy/fr/lecons/en-quoi-consiste-le-mining-de-bitcoins/. [Accès le 21 02 2025].

[43] A. Gayte, «Qu'est-ce que le minage de cryptomonnaie, exactement ?,» 27 10 2022. [En ligne]. Available: https://www.numerama.com/tech/791931-quest-ce-que-le-minage-de-cryptomonnaie-exactement.html.

[44] É. Brémond, «Introduction au minage de crypto-monnaies,» 09 05 2024. [En ligne]. Available: https://cryptoast.fr/introduction-au-minage-crypto-monnaies/.

[45] R. Pandey, S. Goundar et S. Fatima, Distributed Computing to Blockchain Architecture, Technology, and Applications, 1e éd., London, 2023, p. 37.

[46] R3, «Introduction to Corda,» [En ligne]. Available: https://docs.r3.com/en/platform/corda/4.1/open-source0/introduction.html. [Accès le 10 03 2025].

[47] C. Simon, «La Blockchain Based Identity, c'est quoi ? On vous explique,» 14 06 2024. [En ligne]. Available: https://fr.beincrypto.com/technologie/334570/blockchain-based-identity-ssi-explications/. [Accès le 10 03 2025].

[48] P. J. Nitin Naik, «Self-Sovereign Identity Specifications: Govern Your Identity Through Your Digital Wallet using Blockchain Technology».

[49] J. A. Kassem, S. Sayeed, H. Marco-Gisbert, Z. Pervez et K. Dahal, «A Blockchain Identity Management System to Secure Personal Data Sharing in a Network,» *applied Sciences*, pp. 7-8, 24 07 2019.

[50] Visual Studio Code, «https://code.visualstudio.com/docs,» [En ligne]. Available: https://code.visualstudio.com/docs. [Accès le 21 04 2025].

[51] Truffle Suite, «What is Truffle for VSCode?,» [En ligne]. Available: https://archive.trufflesuite.com/docs/vscode-ext/. [Accès le 21 04 2025].

[52] TruffleSuite, «What is Truffle?,» [En ligne]. Available: https://archive.trufflesuite.com/docs/truffle/. [Accès le 21 04 2025].

[53] metamask, «MetaMask developer documentation,» [En ligne]. Available: https://docs.metamask.io/. [Accès le 21 04 2025].

[54] Solidity, «Solidity,» [En ligne]. Available: https://docs.soliditylang.org/en/v0.8.29/#solidity. [Accès le 21 04 2025].

[55] React.dev, «Learn React,» [En ligne]. Available: https://react.dev/learn/describing-the-ui. [Accès le 21 05 2025].

[56] nodejs, «Introduction to Node.js,» [En ligne]. Available: https://nodejs.org/en/learn/getting-started/introduction-to-nodejs. [Accès le 28 05 2025].

[57] Silverfort, «Silverfort Glossaire,» Silverfort, [En ligne]. Available: https://www.silverfort.com/fr/glossary/identity-infrastructure/. [Accès le 30 01 2025].

[58] M. Beschokov, «What is SAML Authentication?,» [En ligne]. Available: https://www.wallarm.com/what/what-is-saml-authentication. [Accès le 03 03 2025].

[59] C. Vestergaard, Blockchain for International Security, The Potential of Distributed Ledger Technology for Nonproliferation and Export Controls, Washington, DC, USA: Stimson Center, 2021.

[60] T. Laurence, Blockchain For Dummies, 3e éd., 111 River Street, Hoboken, NJ: John Wiley & Sons, Inc.,, 2023, p. 30.

[61] É. A. Caprioli, « La blockchain ou la confi ance dans une technologie,» *LA SEMAINE DU PRATICIEN EN QUESTIONS*, p. 1162, Jun,2016.

[62] Coinbase, «What Is a Node in Cryptocurrency?,» [En ligne]. Available: https://www.coinbase.com/fr/learn/crypto-glossary/what-is-a-node-in-cryptocurrency. [Accès le 09 03 2025].

[63] TruffleSuite, «What is Ganache?,» [En ligne]. Available: https://archive.trufflesuite.com/docs/ganache/. [Accès le 21 04 2025].

[64] NextJS, «What is Next.js?,» [En ligne]. Available: https://nextjs.org/docs. [Accès le 21 04 2025].

[65] M. Mededjel, G. Belalem, F. Z. N. Benadda et S. Kadakelloucha, «A Blockchain Application Prototype for the Internet,» *JOURNAL OF COMMUNICATIONS SOFTWARE AND SYSTEMS*, vol. IV, p. 5, 2022.

Abstract

Decentralized identity management is a type of identity management that follows the SSI (Self Sovereign Identity) concept, based on blockchain technology and does not depend on a central authority, which ensures transparency, more privacy, and security compared to the traditional identity management system.

We offer a decentralized identity management system based on blockchain that allows the user to manage (add, modify and delete), share and verify the validity of an identity while using Web Application, Smart Contract, IPFS and blockchain.

Tags: decentralized identity DID, SSI, identity management, smart contract

Résumé

La gestion de l'identité décentralisée est un type de gestion d'identité qui suit le concept SSI, basé sur la technologie blockchain et ne dépend pas d'autorité centrale ce qui assure la transparence, plus de confidentialité et de sécurité par rapport aux systèmes de gestion d'identité traditionnels.

Nous proposons ici un système de gestion d'identité décentralisé basé sur la blockchain qui permet à l'utilisateur de gérer (ajouter, modifier et supprimer), partager et vérifier la validité d'une identité par le biais d'une application Web intégrée aux contrat intelligent, à l'IPFS et à la blockchain.

Mots Clés : Identité Décentralisé DID, SSI, Blockchain, Contrat intelligent

مُلَخَّص

إدارة الهوية اللامركزية هي نوع من إدارة الهوية يستند إلى مفهوم الهوية الذاتية السيادية، ويعتمد على تكنولوجيا سلسلة الكتل دون الحاجة إلى سلطة مركزية، مما يضمن الشفافية، ويعزز الخصوصية والأمان مقارنة بأنظمة إدارة الهوية التقليدية. نقترح نظامًا لإدارة الهوية اللامركزية يعتمد على سلسلة الكتل، يتيح للمستخدم إدارة الهوية (إضافة، تعديل، حذف)، مشاركتها والتحقق من صحتها، وذلك باستخدام تطبيق ويب، العقود الذكية، وتكنولوجيا سلسلة الكتل.

الكلمات المفتاحية :**الهوية اللامركزية، الهوية الذاتية السيادية، إدارة الهوية، العقد الذكي**.