


République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Akli Mohand Oulhadj de Bouira

Faculté des Sciences et des Sciences Appliquées

Département d'Informatique

Mémoire de Master

en Informatique

Spécialité : *Génie des Systèmes Informatiques*

Thème

Réseau social décentralisé basé sur la technologie blockchain

---

Encadré par

- Mr. BADIS Lyes
- Mr. LASLAA Noureddine

Réalisé par

- ZAIDI Yahia
- DADOUN Amayasse

2019/2020

Remerciements

Nous remercions tout d'abord ***ALLAH,*** *le tout puissant de nous avoir donné la patience, la santé et la volonté pour réaliser ce mémoire.*

Un grand merci à nos chers parents et nos familles qui par leurs prières et leurs encouragements, on a pu surmonter tous les obstacles.

Nous tenons à adresser nos sincères remerciements et le grand respect à notre encadreur ***Mr. BADIS Lyes et Mr. LASLAA Noureddine de l'université de Hamad Ben Khalifa de Qatar****, pour leur disponibilité, leur conseils, leur gentillesse et toute l'aide qu'ils nous ont rapportés.*

Nous adressons toutes nos sympathies à tous nos collègues et nos amis pour leurs encouragements et pour tous les moments agréables qu'on passés ensemble.

Nos plus vifs remerciements vont également à nos collègues de la spécialité Master Génie des Systèmes Informatiques et nos enseignants qui nous ont aidés durant notre parcours.

Un grand merci pour tous ceux qui ont contribuées de près ou de loin pour la réalisation de ce mémoire.

Merci à vous tous

Dédicace

*Je dédie ce travail à celle qui a été toujours la source de ma grande affection ... ma **MÈRE**, qu'ALLAH accueille son âme dans son vaste paradis*

*A celui qui a été toujours la source d'inspiration, de courage tout au long de ma vie... mon **PÈRE***

*A mon frère **GH Ilas** et mes sœurs*

A toute ma famille

*A Mon binôme **Yahia***

A tous mes amis sans exception, Moumouh, Assirem, Moumouh, Amar, Amro, Toufik, Youghorta, Brahim, Salah, Anis, Djamel, Hicham, Ayoub, Abdelhak, Lyes, Amine, Mohcin, Fares, Larbi …

A tous mes collègues que j'ai partagés avec eux mes bons moments tout au long de mes études

Amayasse

Dédicace

Je dédie ce travail à L'âme pure de ma tante, qu'ALLAH l'accueille dans son vaste paradis

A ceux qui ont été toujours la source d'inspiration, de courage tout au long de ma vie... mes parents

A mon cher frère Riadh

A mes grands parents

A ma chère Ahlem

A toute ma famille

A mon binôme Amayasse

A mes cousins Ali & Fouad

A tous mes amis sans exception, Cherif, Rabah, Slimane, Walid, Idir, Anis, Djamel, Hicham, Salaheddine, Lyes, Ayoub, Hakou, Brahim, Lilou, Omaro, Amine, Said…

A tous mes collègues que j'ai partagés avec eux mes bons moments tout au long de mes études

Yahia

Résumé

Au cours des dernières années, les réseaux sociaux ont devenus une partie essentielle dans notre vie, et avec cette énorme croissance, les utilisateurs ont commencés à s'inquiéter sur leur vie privée et la protection de leurs données, la majorité de ces réseaux sociaux gèrent et stockent les données des utilisateurs de manière centralisé, ce qui rend la vie privée menacée. Ces limites posées par ces architectures centralisées ont motivé la communauté de recherche pour développer des architectures de réseaux sociaux décentralisée dont leur but essentiel est de donner aux utilisateurs le contrôle total de leurs propres données et relations.

Ce travail a pour objectif de développer un nouveau réseau social décentralisé basé sur la technologie Blockchain, qui est une technique révolutionnaire sécurisée qui fonctionne sans entité centrale de contrôle, pour sécuriser l'accès aux données dans l'espace de stockage, les processus d'authentification et d'inscription et une parfaite gestion des identités. Afin de garantir aux utilisateurs de garder le contrôle total de leurs données. De ce fait, nous allons présenter aussi la plateforme Blockstack que nous avons choisi pour le développement de notre application profitant de ses fonctionnalités offertes en terme de sécurisation des interactions et des identités des utilisateurs, et un contrôle total sur leur espace de stockage de données.

Mots clés : Réseau social, Réseau social décentralisé, Blockchain, Blockstack. ...

Abstract

Over the last few years, social networks have become an essential part of our lives, and with this huge growth, users have started to worry about their privacy and data protection, the majority of these social networks manage and store user data centrally, which makes privacy threatened. These limitations posed by these centralized architectures have motivated the research community to develop decentralized social network architectures whose main goal is to give users full control over their own data and relationships.

The aim of this work is to develop a new decentralized social network based on Blockchain tchnology, which is a revolutionary secure technique that works without a central control entity, to secure data access in the storage space, authentication and registration

processes and perfect identity management, in order to guarantee that users retain total control over their data. Therefore, we will also present the Blockstack platform that we have chosen for the development of our application taking advantage of its functionalities offered in terms of securing users' interactions and identities, and a total control over their data storage space.

Key words : Social network, Decentralized social network, Blockchain, Blockstack. . . .

ملخص

في السنوات القليلة الماضية، أصبحت شبكات التواصل الاجتماعي تمثل جزءًا أساسيًا في حياتنا، ومع هذا النمو الهائل، بدأ المستخدمون يبدون قلقهم بشأن مسألة خصوصيتهم وحماية بياناتهم، حيث تدير غالبية هذه الشبكات بيانات مستعمليها وتخزنها بشكل مركزي، مما يعرض خصوصيتهم للخطر. وهذا ما أدى إلى تحفيز الباحثين على تطوير بنى شبكات تواصل اجتماعي لامركزية، هدفها الأساسي هو منح المستخدمين الحرية المطلقة في تسيير بياناتهم وعلاقاتهم الشخصية.

الهدف من هذه المذكرة هو تطوير شبكة تواصل اجتماعي جديدة لامركزية مبنية على تقنية Blockchain ، وهي تقنية أحدثت ثورة في مجال أمن المعلومات، تعمل بدون جهة تحكم مركزية، لتأمين الوصول إلى البيانات في مساحة التخزين الخاصة بالمستخدمين، وعمليات المصادقة والتسجيل وادارة الهوية بشكل مثالي. وذلك بهدف ضمان تمتع المستخدمين بالسيطرة والتحكم الكامل على بياناتهم. وسنعرض أيضا منصة Blockstack التي اخترناها لتطوير تطبيقنا عليها، والاستفادة من ميزاتها المقدمة من حيث تأمين التفاعلات وهويات المستخدمين، والتحكم الكامل في مساحة تخزين البيانات الخاصة بهم.

الكلمات المفتاحية: شبكات التواصل الاجتماعي، شبكات التواصل الاجتماعي اللامركزية، Blockchain، Blockstack.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

API	Application Programming Interface
BNS	Blockchain Naming System
BTC	Bitcoin
CSS	Cascading Style Sheets
DApp	Application Décentralisée
DDoS	Distributed Denial of Service
DHT	Table De Hachage Distribuée
DNS	Domain Name System
EC2	Amazon Elastic Compute Cloud
ENS	Ethereum Name Service
ETH	Ethereum
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ID	Identifiants
IP	Internet Protocol
JSON	JavaScript Object Notation
LDAP	Lightweight Directory Access Protocol
P2P	Peer to Peer
PHP	Hypertext Preprocessor
RS	Réseau Social
RSD	Reseau Social Décentralisé
SBD	Steem Dollars

SMS	Short Message Service
SIR	Social Information Retrieval
SP	Steem Power
UML	Unified Modeling Language
URL	Uniform Resource Locator
Web	World Wide Web
XML	Extensible Markup Language

Introduction générale

1. Contexte du travail

Au cours des dernières années, les réseaux sociaux (RS) ont réussi à imposer leur place parmi les services les plus populaires sur Internet. Ils sont en plein essor depuis leur apparition, et devenus une partie essentielle dans notre vie culturelle, sociale, économique et professionnelle. Ils sont très utilisés pour partager des données (vidéo, audio, texte,... etc.) entre des utilisateurs distants à travers un réseau étendu.

Ainsi, avec cette grande croissance des plateformes de RS, des inquiétudes croissantes des utilisateurs sur leur vie privée et la protection de leurs données sont augmentées.

La majorité des RS aujourd'huit gèrent et stockent les données des utilisateurs sur des serveurs distants. Ces services centralisés sont une cible de choix pour les pirates et souvent se piraté. En 2016, Yahoo ! a admis avoir perdu des informations pour 500 millions de personnes.

Comme la gestion et le contrôle des données de l'utilisateur sont généralement centralisées, les fournisseurs des RS ont aujourd'hui un privilège pour accéder aux données privées des utilisateurs, ils peuvent obtenir un aperçu en profondeur sur les intérêts personnels de leurs utilisateurs, les opinions et les relations sociales. La plupart de ces fournisseurs vendent les données des utilisateurs sauvegardées dans ses serveurs à des agences de publicité ou à des entreprises commerciales, ce qui rend la vie privée menacée, par exemple, LinkedIn a fui des millions de mots de passe de ses utilisateurs [1], et Facebook a passé des informations commerciales sensibles des utilisateurs au public sans leurs permissions [2].

Les limites posées par ces architectures centralisées des RS ont motivé la communauté de recherche pour développer des architectures de réseaux sociaux décentralisées (RSD).

L'objectif principal de ces RSD est de donner aux utilisateurs plus d'autonomie en termes de stockage et de contrôler les droits d'accès à leurs contenus. En conséquence, les utilisateurs peuvent avoir plus de contrôle sur leur contenu où seront stockés et comment ils seront accessibles. Ceci, à son tour, donne aux utilisateurs la liberté de participer à tout RSD sans se soucier de la sécurité de leur vie privée.

Pour cela nous essayons de rependre à ces besoins des utilisateurs par le développement d'un nouveau réseau social décentralisé basé sur la technologie Blockchain, qui est une technique révolutionnaire sécurisée, transparente et qui fonctionne sans entité centrale de contrôle, pour sécuriser l'accès aux données dans l'espace de stockage et les processus d'authentification et d'inscription, et assurer une distribution des noms décentralisée et une parfaite gestion des identités. Afin de garantir aux utilisateurs de garder le contrôle total de leurs données et leurs identités. De ce fait, nous allons présenter aussi la plateforme Blockstack que nous avons choisi pour le développement de notre application profitant de ses fonctionnalités offertes en terme de sécurisation des interactions et des identités des utilisateurs, et un contrôle total de leurs propres données.

2. Objectifs attendus

A travers le développement de cette application nous visons essentiellement à atteindre les objectifs suivants :

- Développer une application de réseau social décentralisé avec des performances comparables à celles des applications des RS traditionnels.
- Offrir toutes les fonctionnalités des RS traditionnels.
- Permettre aux utilisateurs de partager tout type de fichier avec leurs amis.
- Garantir la sécurité de la vie privée des utilisateurs en leur offrant le contrôle total de leurs propres données et de leurs identités.

3. Organisation du rapport

La suite de ce rapport est organisée comme suit :

Chapitre 1 : Etat de l'art sur les réseaux sociaux : Dans ce chapitre, nous allons présenter un premier concept parmi deux concepts de base en rapport avec notre thème qui est : Les réseaux sociaux. Nous parlerons tout d'abord des réseaux sociaux, puis des réseaux sociaux décentralisés.

Chapitre 2 : Les réseaux sociaux décentralisés à base de la technologie Blockchain : Dans ce chapitre, nous allons présenter le deuxième concept, qui est les RSD basé sur la technologie Blockchain, nous parlerons de la technologie Blockchain, puis des RSD basés sur la technologie Blockchain.

Chapitre 3 : Conception : il sera question dans ce chapitre de présenter notre propre analyse et conception du système, ainsi que les méthodes de développement utilisées pour la réalisation.

Chapitre 4 : Implémentation : il sera question dans ce chapitre de présenter les différents outils et langages utilisés ainsi que nos choix d'implantation.

Chapitre 1

Etat de l’art sur les réseaux sociaux

Chapitre 1

Etat de l'art sur les réseaux sociaux

1.1 Introduction

Avant d'entamer le développement de notre système nous avons commencé par faire une étude bibliographique pour cerner et comprendre les concepts en rapport avec notre domaine d'application. Dans ce cadre, nous avons identifié deux concepts essentiels qui sont : Le concept des réseaux sociaux et le concept des réseaux sociaux décentralisés basés sur la technologie Blockchain. Dans ce chapitre, nous allons aborder le premier concept. Nous allons étudier les réseaux sociaux en général. Ensuite, on va présenter les réseaux sociaux décentralisés qui constitue un domaine de recherche d'actualité.

1.2 Les réseaux sociaux (RS)

1.2.1 Historique des RS

Avec l'émergence du Web 2.0, les utilisateurs sont placés au cœur de diverses technologies Web. Dans ce contexte, les RS apparaissent comme un nouveau type d'application sur Internet, qui peut être considéré comme une extension naturelle des applications Web établissant et gérant des relations explicites entre les utilisateurs pour la majorité des RS. Les RS ont eu beaucoup de succès ces dernières années et le nombre d'utilisateurs connectés sur ces réseaux augmente d'une façon exponentielle. Aujourd'hui, nous témoignons la croissance rapide d'une grande variété de RS [3].

Le premier RS le plus connu, nommé SixDegrees.com, est lancé en 1997. L'origine de son nom provient de la théorie de six degrés de séparation. Cette théorie évoque

la possibilité que toute personne de l'univers peut être reliée à n'importe quelle autre personne, à travers une chaîne du réseau relationnel entre elles comprenant au plus cinq autres maillons intermédiaires. À travers SixDegrees.com, les utilisateurs peuvent créer leurs profils, avoir un groupe d'amis et fournir des informations à leur communauté. Bien que ce réseau ait attiré des millions d'utilisateurs, il n'a pas pu évoluer et a été stoppé en 2000. Le fondateur de SixDegrees.com estime qu'il était en avance sur son temps. Depuis 2003, nous avons assisté à une révolution et une assimilation de sites de RS qui ont établi la plupart des sites les plus populaires de nos jours. Cette révolution a apporté un changement radical sur l'activité des utilisateurs, leurs cultures et le domaine de la recherche sur le Web [4].

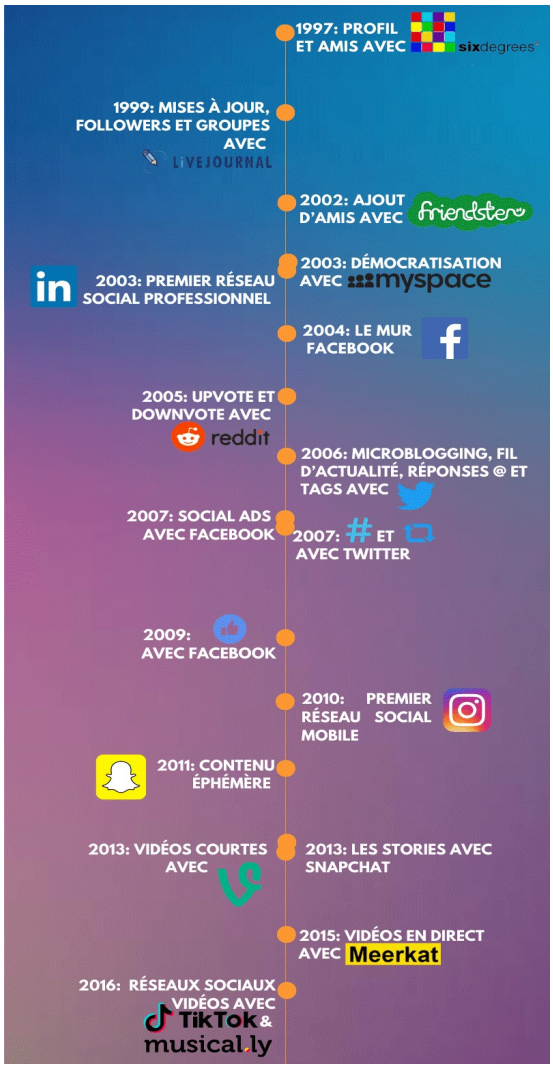

FIGURE 1.1 – Chronologie des RS de 1997 à 2016 [5].

Le progrès important apporté aux sites de RS comme Facebook [1], Flickr [2], LinkedIn [3] et YouTube [4], et la principale force motrice de leur succès, sont expliqués par le fait que les sites de RS mettent l'accent sur les intérêts des utilisateurs pour développer leurs services. Par conséquent, l'objectif principal des RS est de fournir des fonctionnalités de réseautage social comme un service de base pour une variété d'applications et de services de haut niveau [6].

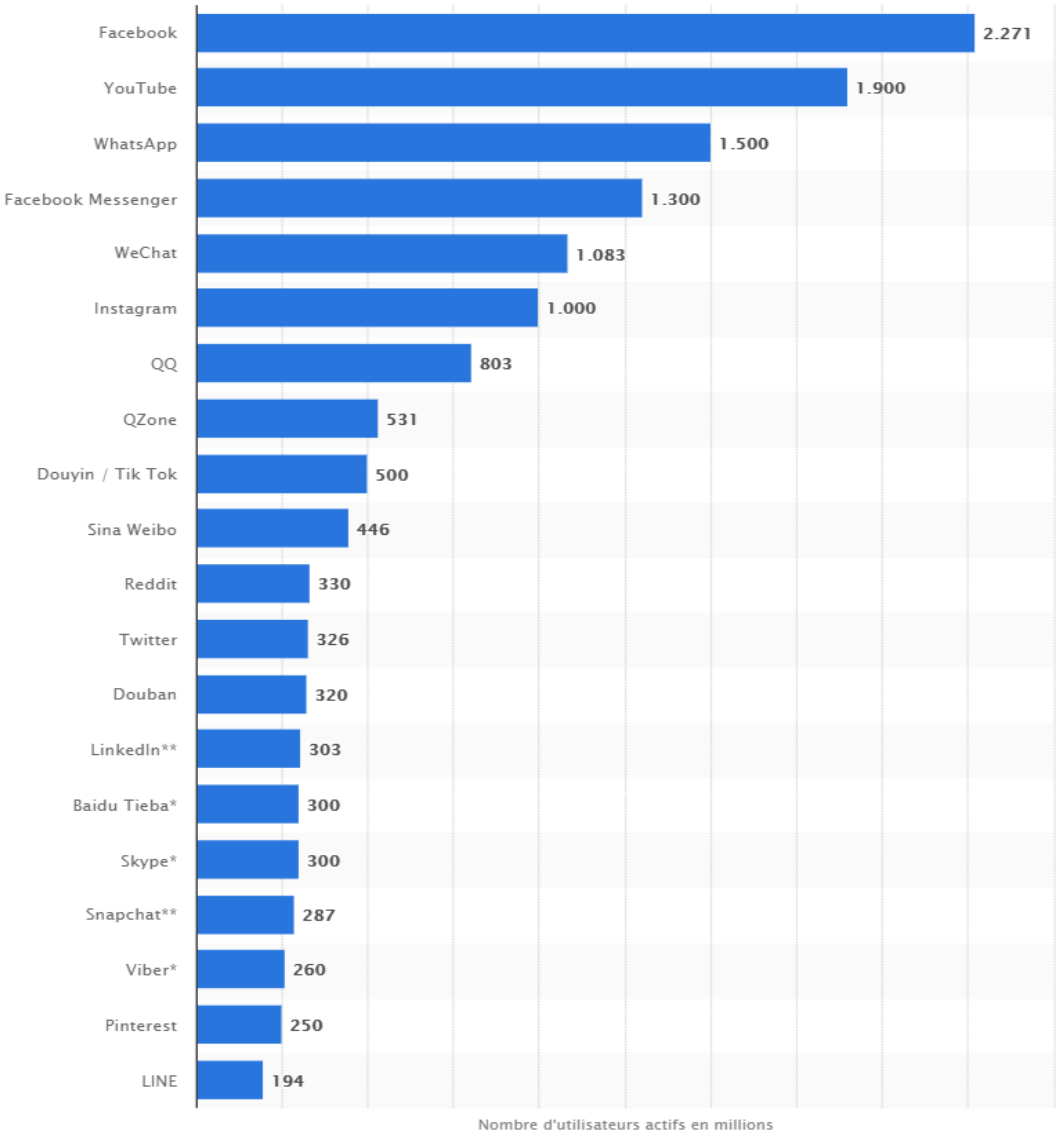

FIGURE 1.2 – Classement des RS les plus populaires dans le monde en janvier 2019, selon le nombre d'utilisateurs actifs [7].

1. https ://www.facebook.com
2. https ://www.flickr.com
3. https ://www.linkedin.com
4. https ://www.youtube.com

1.2.2 Définition d'un RS

"Un RS est un ensemble de relations entre un ensemble d'acteurs", selon Michel Forsé [8]. Ces acteurs sont des entités sociales (individu, groupe ou organisation), reliées par des relations sociales. Ces relations peuvent être de nature parentale, amicale et sentimentale ou d'autres natures (relation de travail, d'affaires, une simple connaissance,...etc.). Elles peuvent se former à travers des contacts directs ou d'intermédiaires. Sur Internet, un RS interconnecte les acteurs d'un RS via Internet selon des relations sociales entre eux.

FIGURE 1.3 – Représentation des réseaux sociaux [8].

Et selon la définition proposée par Boyd et Ellison en 2007 [4], les RS sont des espaces d'échange sur Internet qui permettent aux individus de construire des profils publics ou semi publics, associés à une liste de contacts inscrits sur le même site . Actuellement, Les RS intéressent différents domaines de recherche [9], tels que : sociologie, psychologie, communication, informatique, marketing,...etc.

1.2.3 Architecture de base des RS

Une architecture de base pour les RS est présentée dans la Figure 1.4. Le système entier est formé par les couches suivantes [3] :

- Data Storage Layer (couche de stockage des données) : cette couche est formée par deux composants principaux : le *Storage Manager* (gestionnaire de stockage) qui est chargé de stocker efficacement les informations des graphes sociaux. Ceci est habituellement réalisé en adoptant une mémoire cache distribuée. Le deuxième composant est appelé *Data Store* (espace de stockage des données). Il comprend les éléments de stockage qui mémorisent des informations sur le service du RS. Ces *Data Stores* peuvent être des bases de données Multimédia, des bases de données des Profils utilisateurs,...etc.

- Content Management Layer (couche de gestion de contenu) : cette couche est responsable de trois tâches principales. Tout d'abord, il s'agit de collecter des informations sociales provenant des autres RS pour les incorporer dans un *Content Aggregator* (agrégateur de contenu) qui collecte et organise les contenus reçus des RS et les distribue aussi vers les autres RS. Deuxièmement, il facilite la maintenance et la récupération du graphe présentant le contenu social à travers le *Data Manager* (gestionnaire de données). La troisième tâche *Access Control Manager* (gestionnaire du contrôle d'accès) consiste à contrôler l'accès des utilisateurs par la création et la maintenance du schéma de contrôle d'accès.

- Application Layer (couche d'application) : chaque site Web de RS supporte plusieurs services implantés par la couche application, comme la recherche, le flux d'actualités, les accès mobiles,...etc. Les services de cette couche communiquent avec le *Data Manager* et l'*Access Control Manager* dans le but d'analyser et de gérer le graphe du contenu social. Les applications sont fournies aux utilisateurs à travers l'*Application Manager* (gestionnaire d'applications). L'*Application Manager* facilite l'interaction de l'utilisateur à travers un ensemble d'API (Application Programming Interface). Ce composant contient aussi un Service *Framework* pour le développement de services évolutifs franchissant les obstacles linguistiques.

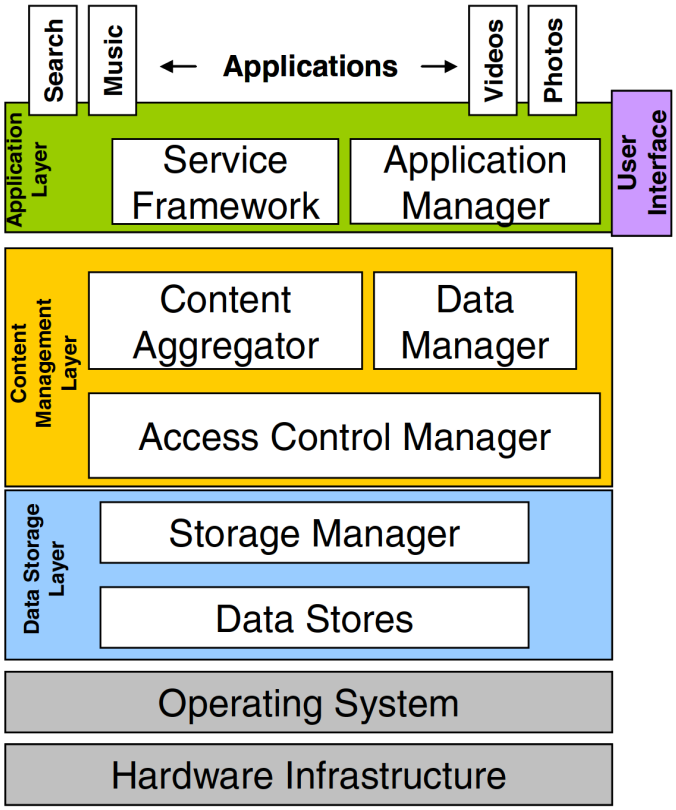

FIGURE 1.4 – Architecture de base d'un RS [3].

Les utilisateurs interagissent avec une plateforme de RS via des requêtes HTTP. Chaque utilisateur peut soit être inscrit comme utilisateur autorisé ou être anonyme. Les utilisateurs enregistrés ont généralement plus de droits que ceux qui sont anonymes (pour commenter des contenus publiés, télécharger des éléments multimédias,...etc.).

Le module de contrôle d'accès est chargé de traiter les paramètres de confidentialité et de sécurité des utilisateurs des RS.

1.2.4 Classification des RS

Une étude globale des aspects des RS existants [3] nous permet de déterminer la classification présentée dans la Figure 1.5 . Nous citons quatre grandes classes pour décrire un tel RS. La première classe couvre le champ d'application des RS en termes d'activités (Scope). La deuxième classe concerne le modèle de données des RS, puisque le modèle de données représente la façon dont les données sont stockées dans un RS. La troisième classe (Plateforme) catégorise les RS selon l'hébergement et la distribution de contenus des serveurs d'applications. La quatrième classe caractérise le réseau relationnel formé

entre les utilisateurs des RS.

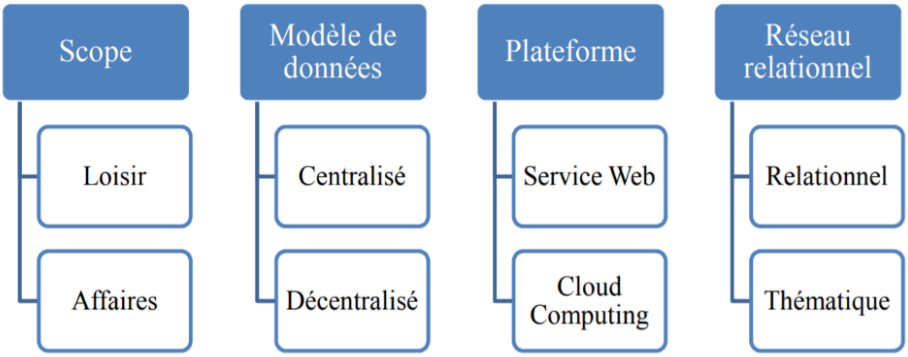

FIGURE 1.5 – Classification des RS [3].

Du point de vue champ d'application, les RS peuvent être classifiés en deux catégories :

- RS de loisirs : la plupart des RS sont des RS de loisirs. Leur rôle est le partage des loisirs qui peuvent être des jeux, des informations, des photos, des vidéos ou des savoirs dans tous les domaines de la vie courante entre des communautés d'amis. Les plus populaires de cette catégorie sont Facebook, Instagram [5] et Flickr.

- RS d'Affaires : leur rôle est de relier les professionnels du monde entier pour les rendre plus interactifs et productifs. Grâce au RS d'affaires, les utilisateurs enregistrés créent des profils qui résument leurs expertises et leurs activités professionnelles. Les RS indicatifs dans cette catégorie sont LinkedIn² et Xing [6].

La deuxième dimension présentée est le modèle de données (Data Model). Ici, nous identifions les catégories suivantes :

- Data Model centralisé (modèle de données centralisé) : les données des RS centralisés sont stockées entièrement sur une seule entité physique qui peut être un *Cluster* ou un *Data Center*. Ceci concentre les données des utilisateurs sous un seul domaine administratif. La plupart des RS s'appuient sur le stockage et la fonctionnalité centralisés vu que leur souci est de soulever des préoccupations concernant la protection de la vie privée des utilisateurs et leur capacité de supporter une base croissante d'utilisateurs et d'applications.

5. https ://www.instagram.com
6. https ://www.xing.com

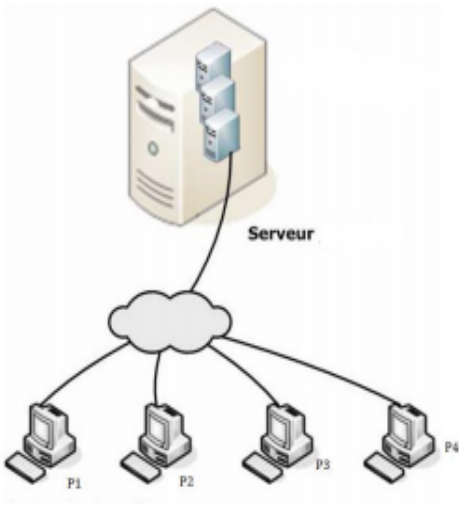

FIGURE 1.6 – Architecture centralisée [3].

- Data Model décentralisé (modèle de données décentralisé) : les données des RS décentralisés sont distribuées à travers multiples domaines administratifs [10,11]. Les serveurs d'applications s'exécutent sur les machines (les pairs) appartenant aux utilisateurs. En général, l'hébergement de données personnelles sur les pairs est plus sécurisé que de déléguer leur contrôle à un fournisseur de services tiers. En outre, ce modèle est moins cher que l'acquisition d'équipements centralisés dédiés. L'inconvénient majeur de cette approche est que les pairs ne sont pas disponibles en permanence [12]. Les pairs sont sujets à des défaillances, des redémarrages, des *power-offs*, des déconnexions de réseau,...etc. [13].

La troisième classe de cette classification présente la plateforme d'exécution des RS. Nous identifions les deux catégories suivantes :

- RS fondé sur une application Web : les serveurs d'applications hébergent les sites Web qui offrent un ensemble de services et d'API. Dans les système fondés sur le Web, la plupart des services de RS sont gratuits pour les utilisateurs.

- RS fondé sur le Cloud : les serveurs d'applications sont hébergés par une infrastructure informatique utilitaire telle qu'Amazon Elastic Compute Cloud (EC2). Dans un tel RS, chaque utilisateur stocke ses propres données sur une instance de machine virtuelle personnelle. Les principaux avantages de ce RS sont sa haute disponibilité et la sécurité des informations. L'utilisation de cette technique offre une vitesse de transfert très élevée. D'autre part, l'hébergement de données dans le Cloud augmente les coûts dus à l'utilisation d'une infrastructure commerciale.

Finalement, la classification couvre le réseau relationnel formé par le RS qui distingue ces deux classes :

- RS orienté utilisateur (relationnel) : les RS de cette catégorie mettent en valeur les relations sociales. Ainsi, le partage de contenus se fait principalement entre les utilisateurs qui appartiennent à la même communauté (exp : Facebook, LinkedIn).

- RS fondé sur le contenu (thématique) : le réseau des utilisateurs n'est pas déterminé par les relations sociales, mais par les intérêts communs entre les utilisateurs. Les RS de cette catégorie sont les *blogs*, les *forums* et les services de partage (exp : YouTube).

1.2.5 Fonctionnalités des RS

Les RS se partagent les fonctionnalités de base mais chaque réseau développe une ou plusieurs services comme il peut bloquer définitivement une fonctionnalité donnée. On résume dans ce paragraphe les différentes fonctionnalités des RS [14].

• **Création de profil**

C'est une étape indispensable que chaque utilisateur doit faire pour pouvoir participer dans les réseaux. Il s'agit de remplir une fiche d'identité contenant essentiellement :

— Un identifiant qui est en général une adresse mail ou un numéro téléphone.
— Le mot de passe qui sécurise l'accès au compte.
— Un pseudo nom qui représente l'utilisateur dans le réseau et qui peut être changé par la suite.
— Les informations personnelles tel que le nom, le prénom, la date de naissance. La vérification de la fiabilité n'est pas exigée dans la plupart des médias sociaux.

On peut ajouter d'autres informations telles que la photo du profil, une citation introductive.

• **Publication**

Une fois le profil est créé, l'utilisateur peut contribuer dans les réseaux en partagent différent types de publications. Certain réseaux limitent la taille de la publication comme Twitter [7] qui limite la publication a 280 caractères. D'autre n'acceptent qu'un seul type de données, c'est le cas des plateformes de partages : YouTube pour les vidéos, SlideShare pour les présentations,...etc.

7. https ://twitter.com

• **Navigation**

L'utilisateur peut consulter le contenu existant dans le réseau avec plusieurs manières :

— il peut visualiser son contenu personnel qui ne contient que les publications partagées par lui-même.

— Il peut visualiser le contenu des autres utilisateurs.

— Souvent, les réseaux proposent ce qu'on appelle un fil d'actualité qui contient un contenu varié extrait des profils des autres utilisateurs. Le choix de cet extrait est en général basé sur des algorithmes intelligents dont la notion d'apprentissage est nettement présente. Le système repose sur le comportement de l'utilisateur pour lui propose un contenu approprié.

• **Interactions avec les autres utilisateurs**

L'interactivité est une propriété de base dans les RS et dans les applications du web 2.0. L'utilisateur peut réagir aux publications des autres utilisateurs par :

— **L'appréciation :** il s'agit de donner un avis positive ou négative sur un contenu. Ceci est réalisé par les boutons de type « like /dislike » (le cas de YouTube) ou par un vote (+1) comme le cas de Google+. Ces mentions d'appréciation ont un effet important sur la propagation du contenu dans le réseau.

— **Commentaire :** L'avis sur un contenu peut être exprimé par un commentaire textuel ou image (fonctionnalité ajouté par Facebook en juin 2013).

— **Partage :** Le partage signifie que l'utilisateur peut publier une nouvelle publication, ou bien publier le contenu provenant d'un autre membre. Dans ce cas, la publication partagée aura la même propriété d'une nouvelle publication : elle apparait dans le profil personnel de l'utilisateur et elle peut être appréciée, commentée ou même partagée une nouvelle fois.

• **Connexion aux autres utilisateurs**

Les liaisons entre les membres d'un RS peuvent prendre plusieurs formes :

— **Liens symétrique (amitié) :** Dans ce type de liens, les deux parts ont le même rôle [9]. Si le membre Yahia envoie une demande d'ami à Amayasse, et que ce dernier accepte cette invitation, Yahia est l'ami d'Amayasse, et de même pour Amayasse.

— **Lien asymétrique (abonnement) :** Cette liaison a un seul sens. Il s'agit de la

fonctionnalité d' « abonner » ou « suivre » [9]. Si par exemple Yahia s'abonne à Amayasse, pour suivre ses publications, Amayasse n'est pas forcement abonné à Yahia.

— **Groupes d'utilisateurs :** Plusieurs utilisateurs regroupés autour d'une thématique ou d'un centre d'intérêt bien définit. Les membres ne sont pas forcément des amis mais ils peuvent publier librement dans l'espace réservé au groupe. En peut différentier une sous-catégorie de membres dans le groupe : les administrateurs ou les modérateurs, ils disposent de quelques privilèges par rapport aux autres membres. Il intervient pour modifier le contenu et les paramètres du groupe comme ils peuvent décider l'ajout ou la suppression des membres (le cas de Facebook).

• **Administration**

L'utilisateur est le maitre de son espace personnelle. Il dispose de quelque autorisation pour la gestion de son contenu. En plus de la publication, l'utilisateur peut :

— Supprimer une publication ou un commentaire.
— Définir la confidentialité d'une publication donnée (public, privée, spécifique).
— Autoriser ou interdire un autre membre à interagir avec lui.
— Autoriser ou interdire les invitations d'amitié.

• **Discussion**

La position de cette fonctionnalité diffère d'un RS à un autre. Il existe des RS qui ignorent ce service. D'autres mettent en place un service de messagerie entre les membres. La messagerie peut être « statique » ou instantané, elle peut supporter ou non la communication audiovisuelle. Il est a noté aussi qu'il existe des RS dont la messagerie constitue la fonctionnalité de base.

• **Jeux**

Les jeux en ligne est l'un des fonctionnalités les plus anciennes dans les RS. Aujourd'hui, il existe des réseaux dédiés uniquement aux jeux (Friendster [8]). Certains RS tel que Facebook intègrent des applications de jeux en ligne.

• **Diffusion en direct du contenu multimédia**

En plus du partage des vidéos, Il existe des RS qui offrent une fonctionnalité plus avancé, il s'agit de la diffusion des vidéos en direct. On cite par exemple : Facebook, Youtube.

8. http ://www.friendster.com

1.2.6 Application des réseaux sociaux

on présente dans cette optique les différentes application des RS [14] :

- **Diffusion d'information**

Les RS offrent au grand public un outil facile et puissant de diffusion d'information, sans avoir besoin d'un grands budget ou un contact avec les medias classiques (journaux, chaines télévisés, sites web informatif), les utilisateurs peuvent partager des publications en toute liberté, et leurs publications se propagent ensuite dans les profils des amis.

- **Communication**

Les RS ont permet aux utilisateurs de rester en contact avec leurs proches ou amis éloignés. En plus, ils ont permet de communiquer avec des personnes célèbres qui sont difficile à contacter dans la vie réelle.

- **Réseautage d'affaires**

Grâce aux liaisons qui existent entre les utilisateurs et la propagation rapide des informations. Les RS permettent de conduire des affaires avec succès, que ça soit la promotion d'un produit ou d'un service, ou même la recherche d'un emploi.

- **Recherche d'information**

En plus des moteurs de recherche, l'utilisateur repose sur les RS pour rechercher un contenu ou une personne, cette tendance a conduit vers l'apparition d'une nouvelle classe de systèmes de recherche d'information : Les systèmes de recherche d'information sociale SIR (Social Information Retrieval) [15]. Ces systèmes se basent sur la fonctionnalité d'indexation personnelle (dit folksonomie). Les utilisateurs jointent à leurs publications des marques « Tag ». Ces marques proposées par les utilisateurs constituent un index collaboratif qui peut servir aux outils de recherche de relier une requête avec un contenu adéquat. Cette technique a rendu la recherche plus efficace par rapport aux différents algorithmes d'indexation utilisés dans les systèmes de recherche d'information classiques. Les résultats de la recherche d'information sociale sont basés sur des propositions des utilisateurs.

- **Marketing**

Pour promouvoir leurs produits et services, les entreprises donnent plus d'importance à la publicité dans les RS. Selon une étude de Emarketer, Marketingland et Recode, « 70% des annonceurs ont augmenté leur budget publicitaire sur les RS en 2015 » [16].

1.2.7 Exemples des RS

1.2.7.1 Exemples des RS grands publics

• **Facebook :** c'est le RS le plus connu : dépasse deux milliard de membres d'inscrits (en janvier 2019) [7]. Le principe est d'échanger avec sa communauté d'amis des publications sur tout et n'importe quoi. L'inscription est obligatoire pour l'utiliser. Pour être amis sur Facebook avec une personne, il faut lui envoyer une demande et que cette dernière l'accepte. Facebook permet également de réagir sur les publications postées par ses amis via le « Like » ou J'aime. Facebook permet beaucoup d'autre chose : discussion instantanée, envoi de message direct, identifier des amis sur une photo,...etc.

• **Twitter :** Il s'agit d'une plateforme de *microblogging*. Comme Facebook, Twitter permet de partager avec d'autres. Le fonctionnement est toutefois différent de Facebook : une limitation à 140 caractères par message (passé à 280 caractères en 2017), la possibilité de suivre d'autres comptes, pas de demande d'invitation, le partage de photo, de vidéo ou d'article se fait par l'utilisation de lien. En gros, on peut publier des SMS avec des liens. Sur Twitter, contrairement à Facebook, nous n'avons pas un mur mais une *timeline* ou fil. Twitter possède un côté plus pro et relationnel. Avec Twitter, nous lisons les tweets de personnes que nous suivons et pouvons les Retweeter pour les partager avec nos *followers*.

• **Youtube :** Il peut aussi être classé dans les RS puisqu'il permet de partager ses vidéos et de commenter et réagir sur les vidéos postées. Youtube appartient à Google. Il n'est pas nécessaire d'être inscrit pour regarder les vidéos postées. Youtube permet également de suivre des thèmes pour être informé de leurs actualités. La grande majorité du contenu présent provient de particuliers [17].

1.2.7.2 Exemples des RS professionnels

• **Linkedin :** un réseau professionnel international permet la mise en relation entre des professionnels. Il offre un espace de présentation de ses compétences et expériences qui peut être consultable par le public. Très utile pour le recrutement. En janvier 2019, le site revendique plus de 303 millions de membres issus de 170 secteurs d'activités dans plus de 200 pays et territoires [7].

• **Xing :** C'est une plateforme allemande qui permet de construire et d'agréger son réseau professionnel. Il possède 3,5 millions d'utilisateurs répartis sur plus de 190 pays [18].

1.2.8 Synthèse

Selon la classification présentée dans la section 1.2.4, le Tableau 1.2 résume les principales caractéristiques des différentes plateformes de RS utilisées.

	Champ d'application		**Modèle de données**		**Plateforme**		**Réseau relationnel**	
Réseau Social	Loisir	Business	Centralisé	Décentralisé	Web	Cloud	Relationnel	Thématique
Facebook	✓		✓		✓	✓	✓	✓
MySpace	✓		✓		✓		✓	
Hi5	✓		✓		✓	✓	✓	
Flickr	✓		✓		✓	✓	✓	
LinkedIn		✓	✓		✓		✓	
Twitter	✓		✓		✓		✓	
YouTube	✓			✓	✓	✓		✓

TABLE 1.1 – Synthèse des caractéristiques de quelques RS [3].

Nous pouvons constater que la plupart des RS sont destinés au loisir et quelques-uns uniquement sont utilisés pour le business comme LinkedIn.

Comme montré par le tableau 1.2 précédent, la plupart des RS existants utilisent une infrastructure centralisée où une seule unité de contrôle pour des données distribuées sur un nombre limité de serveurs. Nous observons une déviation de l'infrastructure centralisée à l'infrastructure décentralisée [13]. En général, la décentralisation peut apporter des solutions aux problèmes soulevés par l'infrastructure centralisée, comme la confidentialité et l'intégrité des informations personnelles [19,20]. La décentralisation promet de meilleures performances, la tolérance aux pannes et la scalabilité en présence d'une base d'utilisateurs et d'applications en pleine extension. La décentralisation de RS a été identifiée comme un défi clé des recherches [13]. Cependant, le passage vers une architecture entièrement décentralisée de RS n'est pas trivial. Cette migration donne lieu à de nombreuses questions de recherche concernant la mise en réseau et l'analyse des RS et des réseaux sociaux décentralisés (RSD).

Nous constatons aussi dans ce tableau que la plupart des RS sont fondés sur les relations sociales entre les utilisateurs (RS relationnels).

1.3 Les réseaux sociaux décentralisés (RSD)

Les architectures centralisées des RS, qui stockent les données personnelles des utilisateurs de manière centralisée, ouvrent largement la possibilité de violation de la vie privée, un fait qui a suscité la demande d'alternatives ouvertes et décentralisées qui présentent actuellement une tendance.

1.3.1 Définition

Un RSD est un RS implémenté de manière distribuée et décentralisée. Les approches exploitées par les RS actuels pour assurer l'indépendance par rapport à un fournisseur centralisé sont généralement basées sur des architectures Peer to Peer (P2P) (telles qu'une table de hachage distribuée (DHT) ou un réseau de serveurs de confiance interconnectés). En effet, chaque utilisateur participant peut agir à la fois comme serveur et comme client [21].

1.3.2 Motivation pour le développement d'un RSD

A ce sujet, il est important de se demander, pourquoi utiliser une infrastructure décentralisée pour supporter les RS, quand l'autre architecture (centralisée) fonctionne bien.

Un des avantages immédiats du modèle décentralisé est assez trivial : il n'est pas centralisé et il n'appartient pas à une seule entité. Nous pouvons aussi donner l'argument traditionnel que le décentralisé s'adapte parfaitement au besoin de la scalabilité, car un nombre croissant d'utilisateurs entraîne naturellement une consommation croissante de ressources réseaux.

Alors que la décentralisation permet d'accroître la confidentialité des utilisateurs par rapport au fournisseur de RS, plusieurs études montrent que la confidentialité est une préoccupation croissante pour les utilisateurs des RSD [22,23]. En effet, quelle que soit leur architecture, l'une des principales caractéristiques des RSD actuels est la possibilité donnée aux utilisateurs de définir leurs préférences en matière de confidentialité sur le contenu de leurs profils, c'est-à-dire de définir quels autres utilisateurs sont autorisés à voir ce contenu.

Étant donné que le nombre de contacts des utilisateurs, ainsi que le nombre et le type de contenus partagés sur les RS, sont en constante augmentation, les membres des RS ont besoin d'un moyen efficace pour définir les autorisations de protection de leurs contenus. Les contenus des utilisateurs doivent être protégés par l'infrastructure des RS conformément aux politiques de confidentialité des utilisateurs contre tout accès non autorisé.

1.3.3 Architectures des RSD

Les approches utilisées par les RSD actuels pour assurer l'indépendance par rapport à une entité centralisée combinent plusieurs niveaux d'architecture, chacun ayant ses propres caractéristiques.

Les RSD peuvent être classés en fonction de la structure des composants du système. Il existe deux principaux composants du système : le contrôle et le stockage. Le contrôle concerne les services de consultation (utilisateur et contenu) et de gestion de l'identité, et le stockage concerne le stockage et la disponibilité des données.

Les architectures décentralisées proposées pour implémenter les fonctionnalités des RS sont classé en trois classes [24] : Architecture P2P, Architecture fédérée et Architecture hybride.

1.3.3.1 Architecture P2P

La première approche adoptée pour décentraliser les RS était les architectures Pair à Pair qui ont connu une réussite dans d'autres domaines d'applications tels que : partage de fichiers, calcul distribué, communication, collaborations.

L'idée clé est d'utiliser les machines des utilisateurs pour stocker le contenu et les informations sociales (liste d'amis, messages,...etc.). La communication entre les utilisateurs et l'interaction avec leurs contenus peuvent être implémentés en utilisant les protocoles de routage P2P (DHT, flooding,...etc.).

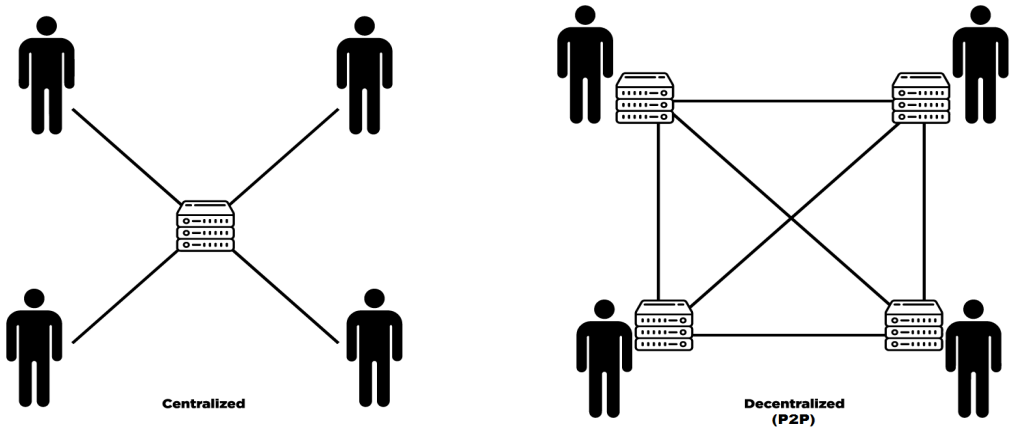

FIGURE 1.7 – Aperçu des réseaux sociaux P2P [24].

Selon l'architecture P2P utilisée, les RS-P2P sont classés en 3 classes [25] :

- Structuré : Dans les architectures P2P structurées, les pairs sont organisés selon une topologie spécifique qui garantit de bonnes performances pour des tâches spécifiques du système, telles que le routage. Cette architecture exploite le hachage pour associer un identifiant au pair et pour apparier les contenus avec les pairs, définissant de ce fait une DHT. La plupart des RSD récentes utilisent une architecture structurée et des DHT pour le service de recherche. Synereo [26], Prpl [27], PeerSon [10], Safebook [28] et Cachet [29] utilisent tous une architecture de contrôle structurée. Vis-a-vis [30] a conçu sa propre structure des arbres de localisation distribués, qui permettent un partage efficace et évolutif.

- Semi-structuré : le RSD semi-structuré utilise des super pairs, qui sont un sous-ensemble de tous les utilisateurs chargés de stocker l'index et de gérer les autres utilisateurs, comme le propose le système Supernova [31]. Une telle structure peut inclure des services de recherche et le suivi du temps de disponibilité des utilisateurs afin de trouver les meilleurs endroits pour la réplication.

- Non structuré : Cette architecture n'impose aucune structure particulière et les utilisateurs sont connectés en fonction de leurs besoins. Aucun utilisateur du système ne stocke d'index, et les opérations du système sont simplement effectuées par l'utilisation des flux ou des communications basées sur les échanges entre les utilisateurs [32]. Ce type de gestion n'a pratiquement pas de frais généraux.

1.3.3.2 Architecture Fédéré

Les architectures fédérées se basent sur des serveurs avec un petit niveau de décentralisation. Dans cette approche le RS est hébergé sur plusieurs serveurs (instances). L'utilisateur peut choisir l'instance d'hébergement ou même créer sa propre instance. Cette approche est proposée dans des travaux académiques tels que Prpl [27], SoNet [33] et Diaspora [34] et implémentée par plusieurs systèmes : PeerTube [9], Mastadon [10], Diaspora [11].

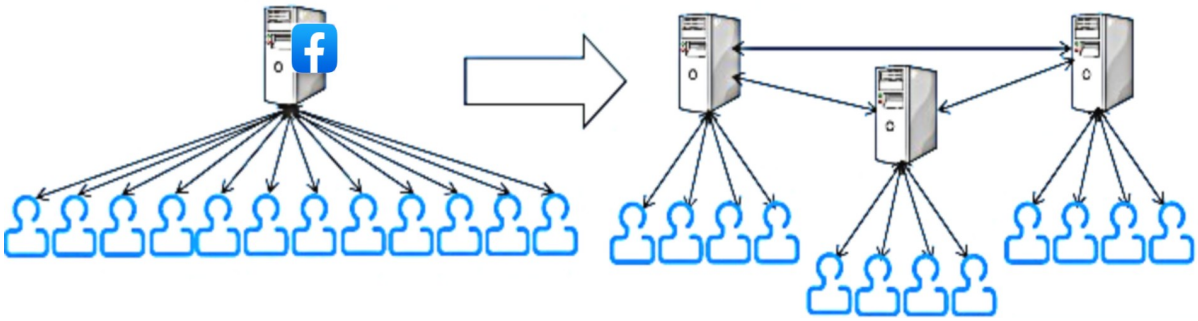

FIGURE 1.8 – Aperçu des réseaux sociaux fédérés [24].

1.3.3.3 Architecture Hybride

Cette architecture exploite l'approche P2P,mais s'appuie également sur un certain service externe fourni par une entité centralisée (comme les Cloudes, les serveurs privés, Dropbox..). ce service permet aux utilisateurs d'exploiter des ressources disponibles en permanence qui garantissent que leur sontenu est toujours accessible, mais cela implique également un coût pour les utilisateurs du RSD.Des architectures hybrides sont proposées dans Vis-a-Vis [11], Vegas [35] et Confidant [36].

9. https ://joinpeertube.org
10. https ://mastodon.social
11. https ://diasporafoundation.org

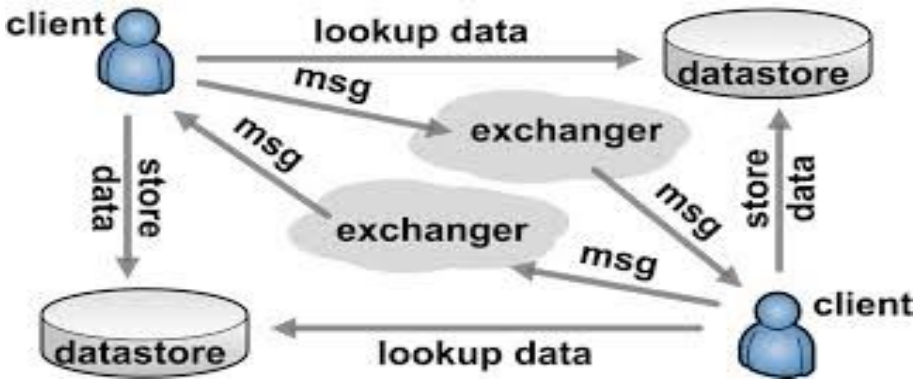

FIGURE 1.9 – Architecture hybride Vegas [35].

1.3.4 Exemples des RSD

1.3.4.1 PeerSon

PeerSon [25] proposent une architecture à deux niveaux dans laquelle le premier niveau est un hachage distribué (DHT), fournit des services de recherche (exp : trouver des amis, trouver du contenu), stocke les métadonnées des utilisateurs et conserve les mises à jour pour un utilisateur pendant sa période hors ligne et le deuxième niveau est constitué des nœuds représentant les utilisateurs. L'idée est d'utiliser la DHT pour trouver les informations nécessaires pour les utilisateurs se connectant directement aux nœuds cibles. Tout le contenu de l'utilisateur est chiffré.

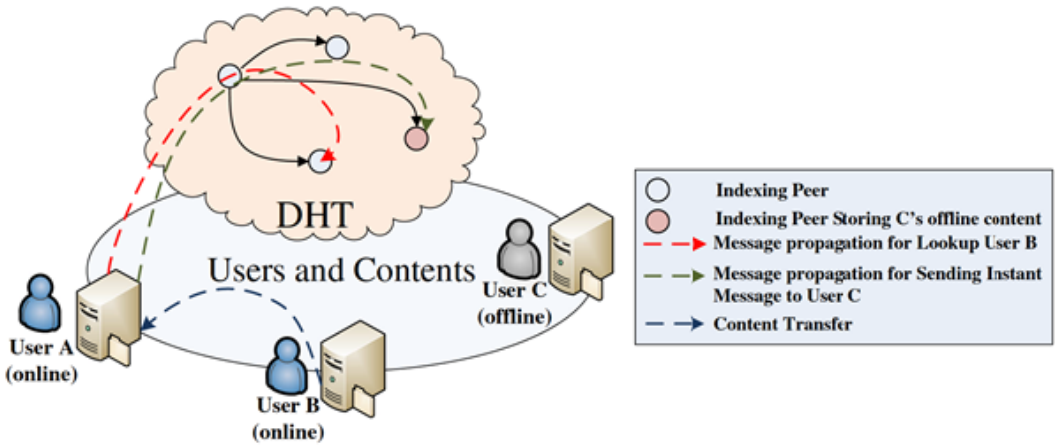

FIGURE 1.10 – Architecture de PeerSon [25].

1.3.4.2 SuperNova

SuperNova [25] propose une architecture RSD basée sur le concept de super pairs. SuperNova permet aux utilisateurs de faire partie du RSD et de partager leur contenu tout en conservant la pleine propriété du contenu. De plus, les utilisateurs peuvent imposer un accès public (accessible à tous), privé (accessible à personne) ou protégé (accessible à un sous-ensemble d'amis) à tous leurs contenus. Dans SuperNova, les super pairs participent activement à la formation de l'infrastructure de contrôle de RS. Les utilisateurs de SuperNova peuvent stocker leurs contenus sur leurs propres machines.

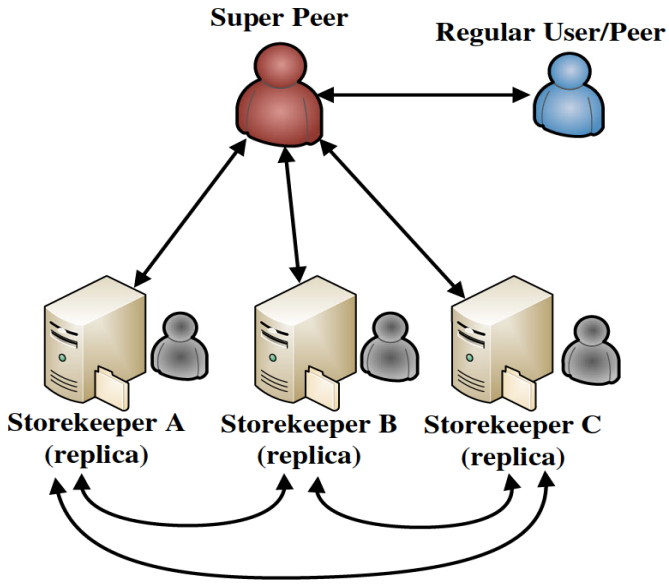

FIGURE 1.11 – Architecture de SuperNova [25].

1.3.4.3 Safebook

Safebook [25] propose une architecture à trois niveaux de RSD qui se concentre principalement sur la vie privée, l'intégrité et la disponibilité (Figure 1.12). Le niveau le plus bas est constitué par les utilisateurs et leurs relations sociales. Ce niveau gère le stockage des données, la disponibilité du contenu et la confidentialité des communications. Une couche P2P, située au-dessus du niveau des RS, fournit les services d'application (par exemple, le service de consultation). L'internet se situe au niveau supérieur, fournissant des services de communication et de transport.

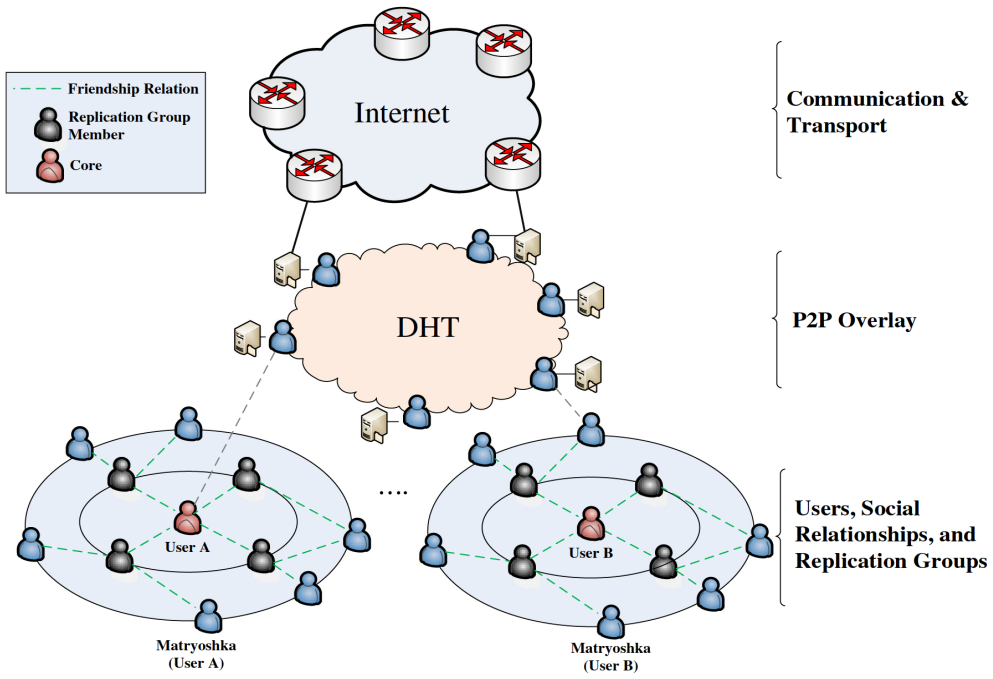

FIGURE 1.12 – Architecture de Safebook [25].

Le tableau comparatif suivant résume les différentes caractéristiques pour quelques RSD :

Système Caractéristiques	**PeerSon**	**Safebook**	**Cachet**	**SuperNova**
Architecture	structurée	structurée	structurée	Semi-structurée
Stockage des données	DHT	Les amis de confiance	DHT	Super peer
Service DHT	- Routage - Stockage pendant une durée de 7 jours	- Routage	- Routage - Stockage des données	-
Mécanisme de disponibilité	Non abordé	Réplication	Réplication et Mise en cache	Réplication
Visibilité de contenu	Non abordé	Groupe	Tous	Tous
Mode hors ligne	Oui si Friendstore en ligne	Oui si miroir en ligne	Oui toujours	Oui toujours
Placement de répliques	Non aborde	Par 1'utilisateur (sur les nœuds amis)	Par le système (place par la DHT)	Par le système (place par le super Peer)
Scalabilité pour le stockage	Non abordé	Moins scalable avec 1'augmentation du contenu	Scalable	Non scalable pour les superpeers quand le nombre d'utilisateurs augmente
Mode connexion	Avec ou sans Internet	Connexion Internet requise		

TABLE 1.2 – Synthèse des architectures des RSD [37].

1.3.5 Défis pour les RSD

Dans cette section, nous listons quelques défis pour les RSD. Si plusieurs d'entre eux sont de nature technique, certains sont des compromis qui dépendent de la préférence, comme la priorité à accorder à la protection de la vie privée ou à la recherche. Ces compromis sont directement liés aux questions techniques et sont donc inclus [38].

- Stockage : Où le contenu doit-il être stocké ? Doit-il être stocké exclusivement dans des nœuds gérés par des amis, ou être crypté et stocké dans des nœudsaléatoires, ou les nœuds doivent-ils être choisis en utilisant d'autres méthodes telles que la DHT ou en fonction de l'historique de disponibilité ? Comme pour le partage de fichiers, il y aura plusieurs réponses à cette question. L'exigence de redondance pour assurer la disponibilité des données dépend dans une large mesure de la durée et de la répartition du temps pendant lequel les pairs sont en ligne. Ces modèles d'activité sont également

influencés par la distribution géographique des pairs et décalés par les fuseaux horaires.

- Mises à jour : Comment gérer les mises à jour, par exemple les mises à jour du statut des amis ? Dans les systèmes de collaboration entre pairs, les mises à jour, par exemple d'un lieu de travail, sont envoyées à un petit groupe de pairs via un mécanisme de synchronisation décentralisé. Dans les réseaux sociaux P2P, avec un stockage et une réplication distribués, et un besoin potentiel d'extensibilité, les exigences changent. Les mécanismes de publication/abonnement P2P sont une possibilité, mais leur sécurité en termes de contrôle d'accès devra être développée par la suite.

- Topologie : Les nœuds doivent-ils être connectés en fonction de leur lien social ? Cela permettrait de regrouper les amis dans le réseau, ce qui faciliterait les mises à jour. L'inconvénient, étant donné la possibilité d'avoir un ensemble d'amis relativement limité, est que cela limiterait la disponibilité et la sécurité de l'accès aux données.

- Découverte, adressage : Comment les utilisateurs peuvent-ils découvrir de nouveaux amis en fonction d'intérêts communs ?. Au cours de plusieurs sessions, les pairs peuvent changer leur adresse physique. Dans un réseau de partage de fichiers typique, ce n'est pas un problème. Il suffit de trouver un pair avec le contenu qu'il recherche. Cependant, les amis et les liens de confiance d'un RS sont essentiels, et il est donc crucial de pouvoir retrouver des amis même s'ils ont changé d'adresse physique, et aussi d'authentifier leur identité.

- Sécurité : Garder le contrôle de ses données par l'utilisateur implique la nécessité d'un support de sécurité. Les principales questions relatives au contrôle de l'utilisateur se situent dans le domaine du contrôle d'accès, par exemple comment pouvons-nous garantir que seuls les amis autorisés peuvent accéder au contenu. Pour le stockage distribué avec d'autres pairs auxquels l'utilisateur ne veut pas nécessairement accéder, le contenu doit être crypté. Pour gérer l'accès aux données cryptées, la distribution et la maintenance des clés doivent être effectuées de manière à ce que le groupe de RS puisse accéder aux données.

1.4 Conclusion

Nous avons présenté dans ce chapitre les différents éléments du premier concept afin de bien comprendre le domaine des RS et des RSD, ainsi que les motivations pour le la transition des RS vers des RSD, et en donnant des exemples pour bien illustrer les avantages et les inconvénients de chacun d'eux. Le prochain chapitre sera consacré aux réseaux sociaux basés sur la technologie Blockchain.

Chapitre 2

Les réseaux sociaux décentralisés à base de la technologie Blockchain

Chapitre 2

Les réseaux sociaux décentralisés à base de la technologie Blockchain

2.1 Introduction

Nous allons aborder dans le présent chapitre, un deuxième concept, qui est les réseaux sociaux décentralisés basés sur la technologie Blockchain, nous allons présenter la technologie Blockchain, ces caractéristiques et son fonctionnement, et comment elle a pu émerger dans les RS et les rendent des RSD, et ce qu'elle a apportée comme avantages pour ces derniers, en illustrant avec des différents exemples des RS qui utilise cette techniques pour plusieurs raisons.

2.2 La technologie Blockchain

2.2.1 Historique

En 2008, un individu (ou groupe) écrivant sous le nom de Satoshi Nakamoto a publié un article intitulé "Bitcoin : A Peer-To-Peer Electronic Cash System". Ce document décrivait une version P2P de la monnaie électronique qui permettrait d'envoyer des paiements en ligne directement d'une partie à une autre sans passer par une institution financière. Bitcoin a été la première réalisation de ce concept. Aujourd'hui, "crypto-monnaie" est le terme utilisé pour décrire tous les réseaux et les moyens d'échange qui utilisent la cryptographie pour sécuriser les transactions, par opposition aux systèmes où les transactions sont acheminées par une entité centrale de confiance [39].

L'auteur du premier article voulait rester anonyme et donc personne ne connaît Satoshi Nakamoto à ce jour. Quelques mois plus tard, un programme open source implémentant le nouveau protocole a été lancé, en commençant avec le bloc *Genesis* de 50 pièces. Tout le monde peut installer ce programme open source et faire partie du réseau P2P de Bitcoin. Sa popularité s'est accrue depuis cela.

La popularité du Bitcoin n'a cessé de croître depuis ce moment. De plus, la technologie Blockchain se trouve maintenant dans une nouvelle gamme d'applications au-delà de la finance.

2.2.2 Définition de Blockchain

Il n'existe actuellement aucune définition stricte et universellement approuvée de la Blockchain, car la technologie de la Blockchain continue à se développer rapidement [40]. Nous comparons principalement l'interprétation de la Blockchain à partir des définitions suivantes :

La Blockchain peut être définie comme un registre numérique distribué immuable (infalsifiable), qui est sécurisé par une cryptographie avancée, répliqué entre les nœuds du réseau P2P, et utilise un mécanisme de consensus pour s'accorder sur le journal des transactions, alors que le contrôle est décentralisé [39].

Ou par [41], La Blockchain (chaîne de blocs) est une technologie de stockage et de transmission d'informations qui est sécurisé, transparente et qui fonctionne sans organe centrale de contrôle.

L'article de Nakamoto considère la Blockchain comme une base de données distribuée décentralisée, dépourvue de confiance et gérée de manière collective [42], où distribué fait référence non seulement au stockage distribué, mais aussi aux enregistrements de données distribués [43].

D'un point de vue technique, la Blockchain est un nouveau mode de structure, de stockage et d'expression des données qui est intégré aux nouvelles technologies [44, 45, 46].

La technologie à la base de Bitcoin et d'autres crypto-monnaies, est une base de données de registre distribuée (ledger) pour l'enregistrement des transactions, permettant ainsi aux utilisateurs de partager leur registre de transactions.

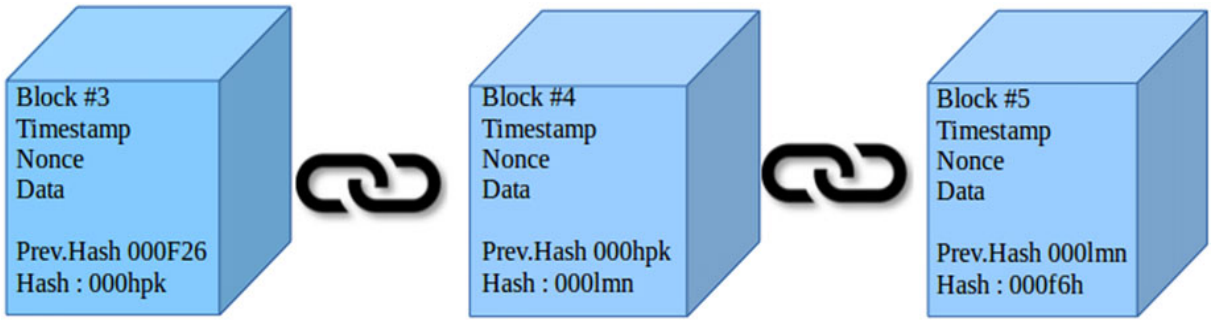

FIGURE 2.1 – Aperçu de Blockchain [41].

2.2.3 Caractéristiques de Blockchain

Nous allons maintenant aborder les caractéristiques suivantes de la Blockchain [47] :

— **Sécurisé :** Il est vraiment difficile pour quiconque de falsifier les transactions ou les registres présents dans la Blockchain, ce qui la rend plus sécurisée, elle est donc considérée comme une source d'information fiable.

— **Portée mondiale :** La Blockchain a été adoptée dans le monde entier et bénéficie du soutien de nombreux investisseurs des secteurs bancaire et non bancaire, ce qui en fait une base technologique acceptée dans le monde entier.

— **Opérations automatisées :** Les opérations sont entièrement automatisées par des logiciels. Il n'est pas nécessaire de faire appel à des sociétés privées pour effectuer les opérations, c'est pour cela qu'aucune intermédiaire n'est nécessaire pour effectuer les transactions et que la confiance est assurée, de sorte que les gens peuvent effectuer leurs propres transactions.

— **Open source :** La Blockchain est une technologie à open source. Toutes les opérations sont effectuées par la communauté open source.

— **Décentralisé :** Blockchain fonctionne en mode décentralisé, dans lequel les données sont stockées dans tous les nœuds du réseau. Si un nœud tombe en panne, cela n'a aucun impact sur les autres nœuds ou sur les autres données, car elles sont distribuées globalement sur tous les nœuds.

— **Flexible :** La Blockchain est programmable, utilisant des concepts de programmation de base et une sémantique de programmation, ce qui la rend très flexible.

2.2.4 La cryptographie et la crypto-monnaie

La technologie de la Blockchain est la base de la crypto-monnaie et elle utilise la cryptographie pour sécuriser les données. Dans cette section, nous allons aborder la cryptographie, la crypto-monnaie et la manière dont la cryptographie est utilisée dans l'implémentation de la crypto-monnaie [47].

2.2.4.1 La crypto-monnaie et ses utilisations

La crypto-monnaie est un sujet très intéressant et fascinant qui est devenu un phénomène mondial. Elle est différente de toutes les monnaies que l'on trouve dans le monde entier. Depuis l'apparition de la crypte-monnaie, tous ceux qui la possède ou la contrôle, ses avantages et ses limites, son utilisation et bien plus encore, sont tous basés sur des idées et des processus nouveaux. Voyons maintenant un bref aperçu des différentes caractéristiques de la crypto-monnaie.

Les principales caractéristiques de la crypto-monnaie sont les suivantes :

— En 2008, la première monnaie cryptographique décentralisée a été conçue.
— C'est un élément numérique utilisé comme moyen d'échange.
— Il sécurise les transactions et contrôle l'approvisionnement en utilisant la cryptographie.
— Il s'agit d'un sous-ensemble de monnaies alternatives.
— La crypto-monnaie est une monnaie numérique, tandis que la technologie de base qui permet le transfert de pièces de monnaie numériques entre les individus est la Blockchain.

2.2.4.2 Les principales techniques de cryptographie utilisées dans la crypto-monnaie

Les principales techniques de cryptographie utilisées dans la crypto-monnaie sont une fonction de hachage et une signature numérique. Examinons brièvement chacune d'entre elles :

A. Fonction de hachage

La fonction de hachage est une fonction mathématique ayant les propriétés suivantes :

— Toute entrée que nous fournissons, que ce soit une chaîne de caractères, un nombre, un nombre flottant ou autre, peut-être de n'importe quelle taille.

— Il produit un résultat de taille fixe, comme un résultat de hachage de 128 bits ou même de 256 bits.

— Il est résistant aux collisions.

— Il cache les données dedans.

Voici un exemple de fonction de hachage dans la figure suivante :

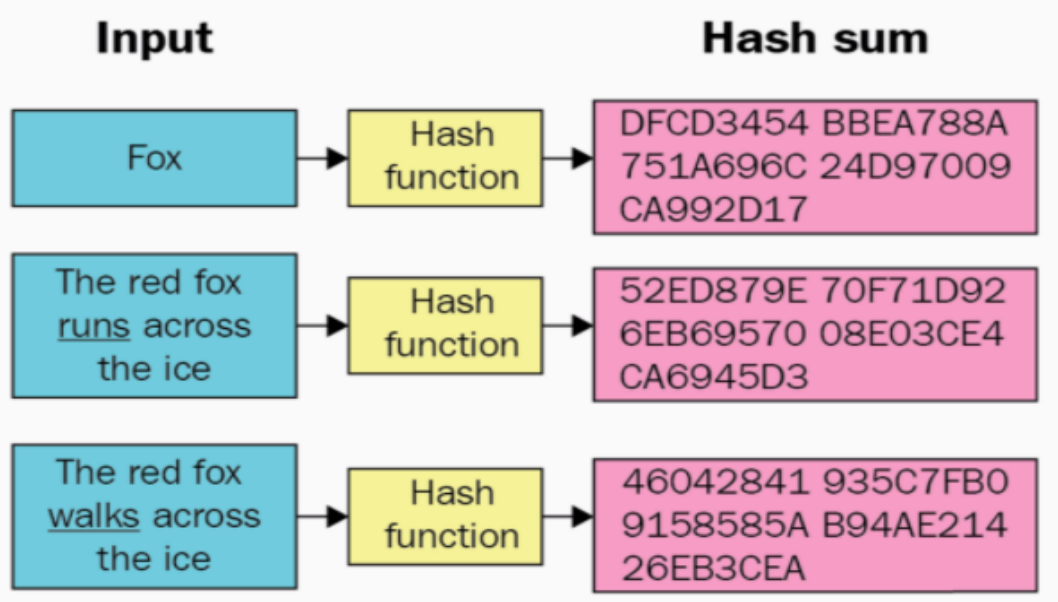

FIGURE 2.2 – Exemple de fonction de hachage [47].

Dans le diagramme précédent, nous avons donné à Fox, une chaîne de caractères, comme entrée au mécanisme de cryptage et nous lui avons appliqué une fonction de hachage. Elle donne un résultat de hachage fixe spécifique, qui est également connu sous le nom de somme de hachage. Supposons que nous passions une autre déclaration qui dit "*The red fox runs across the ice*" et que nous lui appliquions une fonction de hachage. Il nous donne une somme de hachage spécifique. Ensuite, nous ajoutons un autre énoncé, "*The red fox walks across the ice*", et nous voyons qu'il modifie la fonction de hachage au fur et à mesure que l'entrée a été modifiée. C'est la caractéristique clé que la fonction de hachage apporte.

B. Signature numérique

C'est le deuxième élément de base de cryptographie utilisé dans la crypto-monnaie et il est numériquement analogue aux signatures manuscrites que nous faisons habituellement.

Les propriétés de la signature numérique sont les suivantes :

— Vous pouvez créer vos propres signatures, mais elles peuvent aussi être vérifiées

par une autre personne

— La signature est liée à un document ou à un message particulier, de sorte qu'elle ne peut être utilisée à plusieurs reprises pour différents documents et messages

Voici un exemple de signature numérique dans la figure suivante :

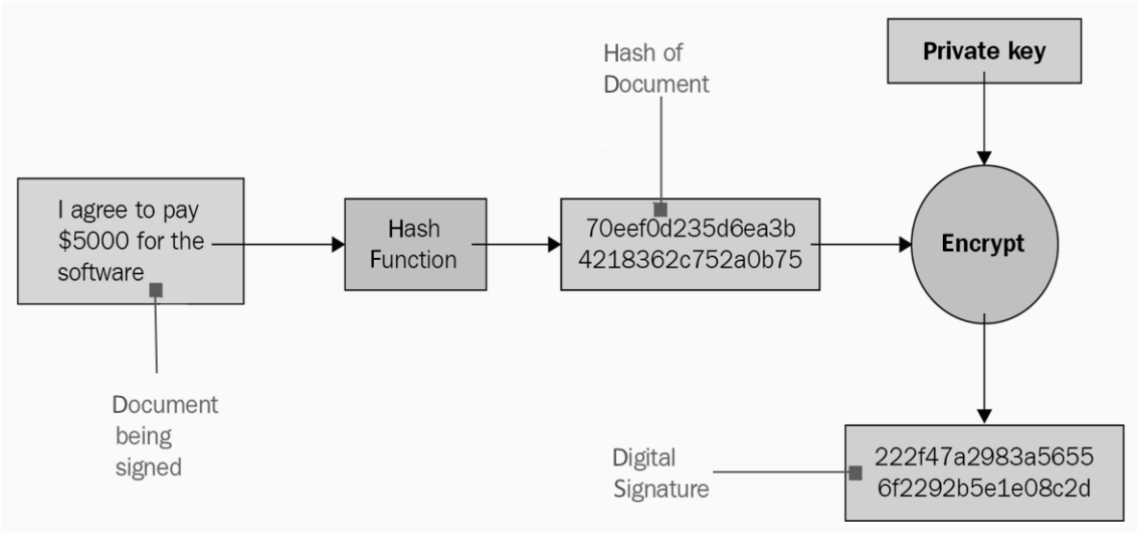

FIGURE 2.3 – Exemple de signature numérique [47].

Prenons un exemple et disons qu'il y a une chaîne de caractères dans le document qui dit : "J'accepte de payer 5 000$ pour le logiciel". Comme il s'agit du document qui est signé, nous lui appliquons d'abord une fonction de hachage. Cela nous donne un hachage ou un résumé du document, dont la taille est à nouveau fixe. Ensuite, nous appliquons notre clé privée, ou une clé spécifique, pour le crypter et nous obtenons le résultat sous forme de signature numérique. Cette signature numérique est signée spécifiquement pour cette entrée, qui indique que "j'ai accepté de payer 5 000$ pour le logiciel".

Là, nous avons vu comment la cryptographie et la crypto-monnaie sont des éléments numériques largement utilisés. Nous avons également vu comment la crypto-monnaie utilise la cryptographie et ses techniques, telles que les fonctions de hachage et les signatures numériques.

2.2.5 Structure de Blockchain

Dans cette section, nous allons découvrir les aspects suivants de la Blockchain : sa structure, ses éléments de base et ses parties essentielles qui la rendent disruptive, robuste, solide et inviolable [47].

Examinons la structure de la Blockchain. La structure de la Blockchain est très similaire à celle des listes chaînées ou des arbres binaires. Les listes chaînées ou les arbres binaires sont reliés entre eux par des pointeurs qui pointent vers les éléments de la liste précédente ou suivante sur les nœuds de la liste chaînée. La structure de la Blockchain n'est pas vraiment différente de celle des arbres binaires, mais la principale différence est que la Blockchain est inviolable et qu'il est également très facile de savoir si une falsification a eu lieu.

Dans le diagramme suivant, nous allons voir une représentation de la façon dont la Blockchain est construite et comment il s'agit d'une liste chaînée :

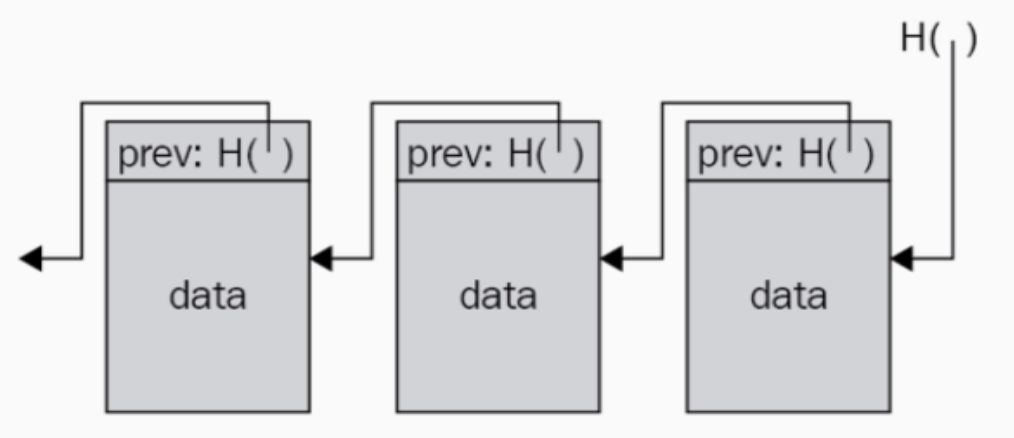

FIGURE 2.4 – Construction de la Blockchain [47].

Examinons maintenant la structure et les éléments de la Blockchain.

La Blockchain est une liste chaînée qui est construite avec des pointeurs de hachage au lieu de pointeurs. C'est la raison exacte pour laquelle la Blockchain ressemble à une liste chaînée, mais elle est différente car, dans la liste chaînée, nous avons utilisé les pointeurs pour pointer vers les nœuds précédents pour les éléments des listes, mais dans le cas de la Blockchain, les pointeurs sont des pointeurs de hachage et pas seulement des pointeurs simples.

Ainsi, en règle générale, tout bloc de la Blockchain se compose de trois parties, ou pieds, telles qu'un en-tête, un merkle et une liste des identifiants (ID) de transaction. Il s'agit d'un bloc récemment créé.

Nous pouvons voir la structure de la Blockchain dans la figure suivante :

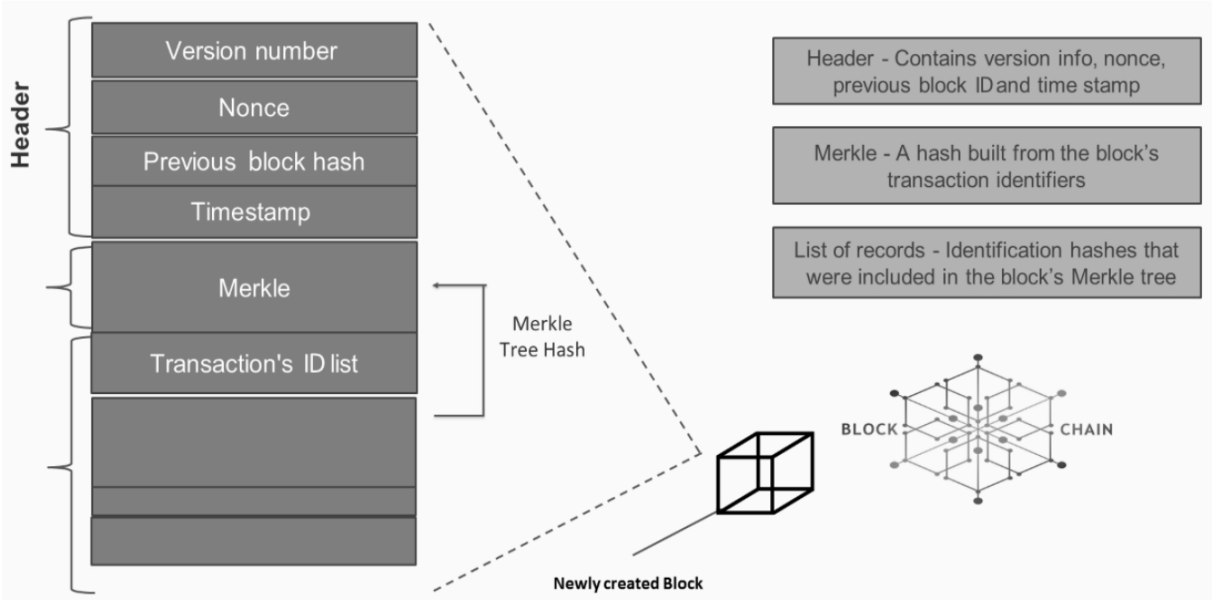

FIGURE 2.5 – Structure de la Blockchain [47].

Nous examinerons maintenant les blocs suivants et leurs éléments :

— **En-tête :** Ce bloc contient les informations de version du bloc, le nonce, l'ID du bloc précédent et le horodatage « *timestamp* » qui est à nouveau haché au moment de la création du bloc.

— **Merkle :** Ce bloc est un hachage construit à partir des identifiants de transaction du bloc.

— **Liste des ID de transaction :** Ce bloc représente les transactions elles-mêmes. C'est une liste d'enregistrements, de hachages des identifiants, qui sont inclus dans le bloc de l'arbre Merkle. Le bloc est ensuite créé avec tous les détails précédents. Ce bloc nouvellement créé est ajouté à la Blockchain.

2.2.6 Fonctionnement

La figure 2.5 illustre le mécanisme de fonctionnement des transactions dans le réseau Blockchain [48].

Les étapes de ce mécanisme sont les suivantes :

— Quelqu'un demande une transaction.

— La transaction est diffusée sur un réseau P2P public (réseau Blockchain) composé de plusieurs nœuds.

— Le réseau de nœuds valide la transaction en utilisant les algorithmes de hachage.

— Une fois vérifiée, la transaction est combinée avec d'autres transactions pour créer un nouveau bloc de données pour le registre.

— Le nouveau bloc est ajouté à la Blockchain existante, sous une forme qui est permanente et inaltérable.

— Enfin la transaction sera effectuée avec succès.

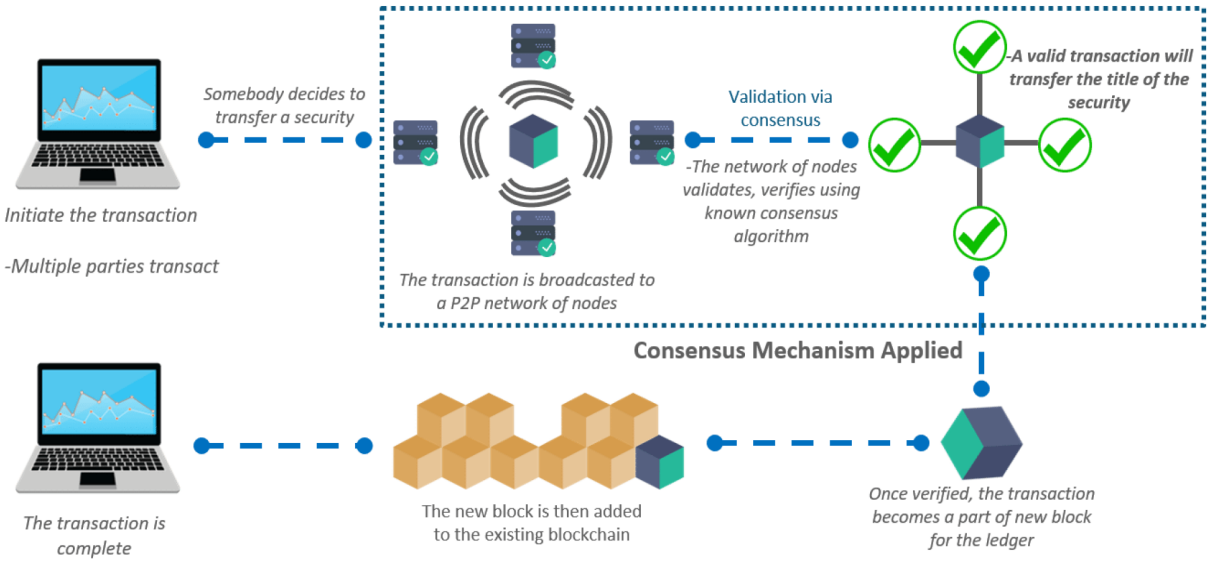

FIGURE 2.6 – Les étapes de mécanisme de fonctionnement des transactions dans le réseau Blockchain [48].

2.2.7 Types de Blockchain

Voici les différents types de Blockchain [47] :

• **Public :** Les Blockchains publiques ont des registres visibles par tous sur l'internet et tout le monde peut vérifier et ajouter un bloc de transactions au Blockchain. Par exemple, Bitcoins,et les centaines d'Altcoins disponibles sur le marché aujourd'hui.

• **Privé :** Le deuxième type de Blockchain est la Blockchain privée. Toutes les autorisations sont centralisées au sein d'une organisation. Les Blockchains privées ne permettent qu'à des personnes spécifiques ayant des rôles spécifiques dans l'organisation de vérifier les blocs de transactions, mais tout le monde sur l'internet est autorisé à les consulter. Cela dépend également de la décision de l'organisation. Citons, par exemple, MultiChain et Blockstack.

• **Consortium :** Le troisième type de Blockchain, et le plus populaire, c'est le consortium. Il est contrôlé par un groupe de membres. Ces membres sont issus d'entreprises de

haut niveau qui se sont présentées pour apporter des modifications à la Blockchain à des fins spécifiques. Ainsi, seul un ensemble prédéfini de nœuds a accès à l'écriture de données ou de blocs dans la Blockchain. On peut citer par exemple Ripple, R3, Hyperledger 1.0 et Hyperledger 2.0.

Propriétés	**Blockchains publiques**	**Blockchains consortiums**	**Blockchains privées**
Accès aux données en lecture	Aucune restriction	Avec ou sans restriction	Avec ou sans restriction
Accès aux données en écriture	Aucune restriction	Aucune restriction ou uniquement des entités présélectionnées	Une entité unique
Participer aux processus de validation	Aucune restriction	Uniquement les entités présélectionnées	Pas de validation de données car une seule entité peut ajouter des données
Complexité du consensus de validation	Consensus de validation complexe pour être tolérant aux fautes byzantines	Consensus de validation facilité	Pas de consensus nécessaire

TABLE 2.1 – Comparaison des Blockchains publiques, consortiums et privées [49].

2.2.8 Identification des utilisateurs

Les Blockchains sont potentiellement anonymes, mais pas nécessairement de cette façon. L'avantage est que cela permet de créer une confiance en utilisant le cryptage (valeurs binaires). Toute personne possédant le matériel et les logiciels appropriés peut participer au réseau [50].

Il faut également faire la distinction entre les Blockchains publiques et privées. Dans le premier cas, il n'existe pas d'autorité de ce type qui contrôle les autorisations ou la

vérification des identités, et dans le cas des Blockchains privées, il y a une autorisation à donner et il est probable que l'autorité qui a créé la Blockchain exige l'identification des utilisateurs.

Les utilisateurs effectuent des actions sur la Blockchain et celles-ci sont marquées par des signatures numériques. La signature utilisée dans la plate-forme n'est pas électronique mais numérique. Les signatures électroniques font référence à tout type de données qui sont transférées électroniquement pour signer un document ou un contrat, comme une signature manuscrite ou une image numérisée. Elles sont susceptibles d'être falsifiées et contrefaites. Les signatures numériques sont conçues comme des enregistrements électroniques basés sur la cryptographie pour certifier un document d'un espace numérique à un autre. Le facteur intéressant est qu'elle préserve l'intégrité d'un document, indique les signataires, mais ne révèle pas leur identité personnelle. Dans la Blockchain, les utilisateurs signent un hachage qui constitue une représentation du document principal.

2.2.9 Exemples d'applications de Blockchain

Les Blockchains les plus connus sont Bitcoin et Ethereum. Présentant brièvement chacune d'entre elles :

2.2.9.1 Bitcoin

Les billets de banque et les pièces métalliques sont ennuyeux et peu pratiques, et nous avons maintenant l'Internet. L'argent numérique semble donc être une idée utile.

La solution à laquelle les pays développés sont arrivés est d'utiliser les banques : vous avez un compte et vous pouvez transférer de l'argent vers les comptes d'autres personnes, par carte de débit, carte de crédit, PayPal ou autre. Grâce à l'autorité centrale, la réglementation est judicieuse, les erreurs et les vols peuvent être évités,...etc. C'est aussi une bonne solution par rapport à la monnaie de papier [51].

Mais ce n'est pas une solution complète, le lecteur de carte d'un magasin peut être en panne, votre passerelle de paiement peut facturer des frais, vous pouvez vouloir envoyer de l'argent à quelqu'un qui n'est pas sur le même réseau bancaire, vous tenez à votre vie privée, vérifier avec votre banque chaque fois que cela devient ennuyeux. Une forme d'argent liquide numérique serait donc une bonne idée [51].

Le Bitcoin est la première monnaie numérique décentralisée, vous pouvez donc envoyer

de l'argent sans avoir à passer par une chambre de compensation centrale, et à ce titre, c'est une invention technologique révolutionnaire. Elle a changé la façon dont nous calculons les choses et la façon dont nous utilisons les logiciels et les ordinateurs. Le Bitcoin et la Blockchain sont considérés comme la prochaine grande vague de changement après l'Internet [47].

Le registre des transactions de Bitcoin, la " Blockchain ", est présenté comme étant immuable (personne ne peut le modifier sans qu'il soit évident qu'il a été falsifié). L'idée est qu'il n'y a pas de contrôle central, que tout le monde peut faire fonctionner un nœud Bitcoin et faire partie du réseau, que personne ne peut bloquer ou annuler vos transactions et que vous n'avez à croire personne sur le système [51].

2.2.9.2 Ethereum

Ethereum est un réseau P2P de machines virtuelles que tout développeur peut utiliser pour exécuter des applications décentralisées. Ces programmes informatiques peuvent être n'importe quoi, mais le réseau est conçu pour exécuter des instructions qui s'exécutent automatiquement lorsque certaines conditions sont atteintes, comme un contrat. Ethereum utilise sa propre Blockchain publique décentralisée pour stocker, exécuter et protéger ces contrats de manière cryptée. Chaque ordinateur du réseau télécharge une petite machine virtuelle pour se synchroniser avec la Blockchain d'Ethereum et reste disponible pour exécuter les contrats. Ce réseau décentralisé d'ordinateurs offre la sécurité, la fiabilité et la puissance de calcul nécessaires à l'exécution des contrats. Bien évidemment, ce réseau de consensus n'est ni gratuit ni privé, de sorte que les développeurs ne l'utilisent que pour obtenir un consensus sur les résultats et lorsque leurs données peuvent être publiques. La Blockchain Ethereum est consultable par le public [52].

Le but d'Ethereum est de créer un protocole alternatif pour la construction d'applications décentralisées, en fournissant un ensemble différent de solutions que nous pensons être très utiles pour une large gamme d'applications décentralisées, avec un intérêt particulier pour les situations où le temps de développement rapide, la sécurité des petites applications rarement utilisées et la capacité des différentes applications à interagir très efficacement, sont importants. Ethereum y parvient en construisant ce qui est essentiellement la couche de base abstraite finale : une Blockchain avec un langage de programmation intégré et complet [52].

• **Smart contract**

Un contrat intelligent est un accord conclu entre deux entités qui utilisent la technologie Blockchain pour obliger les parties à en respecter les termes, plutôt que de recourir aux méthodes traditionnelles telles que la confiance en un intermédiaire ou le recours aux lois pour régler les différends [53].

Grâce à la Blockchain Ethereum, vous pouvez créer des contrats intelligents avec le langage Solidity (entre autres). Ethereum n'est pas la seule plateforme qui permet de créer des contacts intelligents, mais c'est le choix le plus populaire, car il a été conçu dès le départ pour aider à les établir [53].

2.3 Les RSD basés sur Blockchain

La réussite de la technologie Blockchain dans le domaine du commerce électronique a déclenché plusieurs initiatives pour l'adopter à d'autre domaine d'application. La motivation principale de Blockchain et l'élimination d'une partie centrale qui domine un système d'échange. L'idée révolutionnaire est de donner aux utilisateurs du système le contrôle total de leurs données. Cette idée coïncide exactement avec l'objectif de décentralisation des RS.

Ainsi, Blockchain offre d'autres possibilités pour enrichir les performances des RS tel que : la récompense des utilisateurs, la détection des rumeurs (*fake news*) et l'élimination de censure centralisé (suppression de censure ou bien une censure basé communauté).

Stocker les données sur la Blockchain n'est pas une bonne idée. Elle peut causer l'explosion de la taille de la Blockchain (qui est un registre dupliqué dans les machines de tous les utilisateurs du réseau). En plus, les données stockées sur une Blockchain sont immuables (non modifiable), au contraire des données sociaux (post, commentaire...) qui sont très dynamique. Cette idée a été implémentée dans Peepeth [1] mais avec plusieurs restrictions (limiter le nombre de publications et réactions par jour, stocker temporairement *offchain* (hors de la Blockchain) et l'impossibilité de modifier ou supprimer un contenu une fois stocké sur la Blockchain).

1. https ://peepeth.com

Plusieurs travaux académiques ont proposé l'utilisation partielle de Blockchain dans les architectures des RSD. On peut citer Tawki [54] qui s'appuie sur la Blockchain pour gérer les identités des utilisateurs et le routage des messages.

Les auteurs de [55] et [56] ont proposé des modèles de contrôle d'accès basé sur la technologie Blockchain. Dans ces deux proposition le stockage et géré dans des nœuds indépendant de la Blockchain et les informations relatives aux droit d'accès (liste de contrôle d'accès, liste des rôles,...etc.) sont stockées dans la Blockchain. Les nœuds de stockage, une fois sollicités par un utilisateur, consultent la Blockchain pour décider de livrer un contenu ou bloquer l'accès.

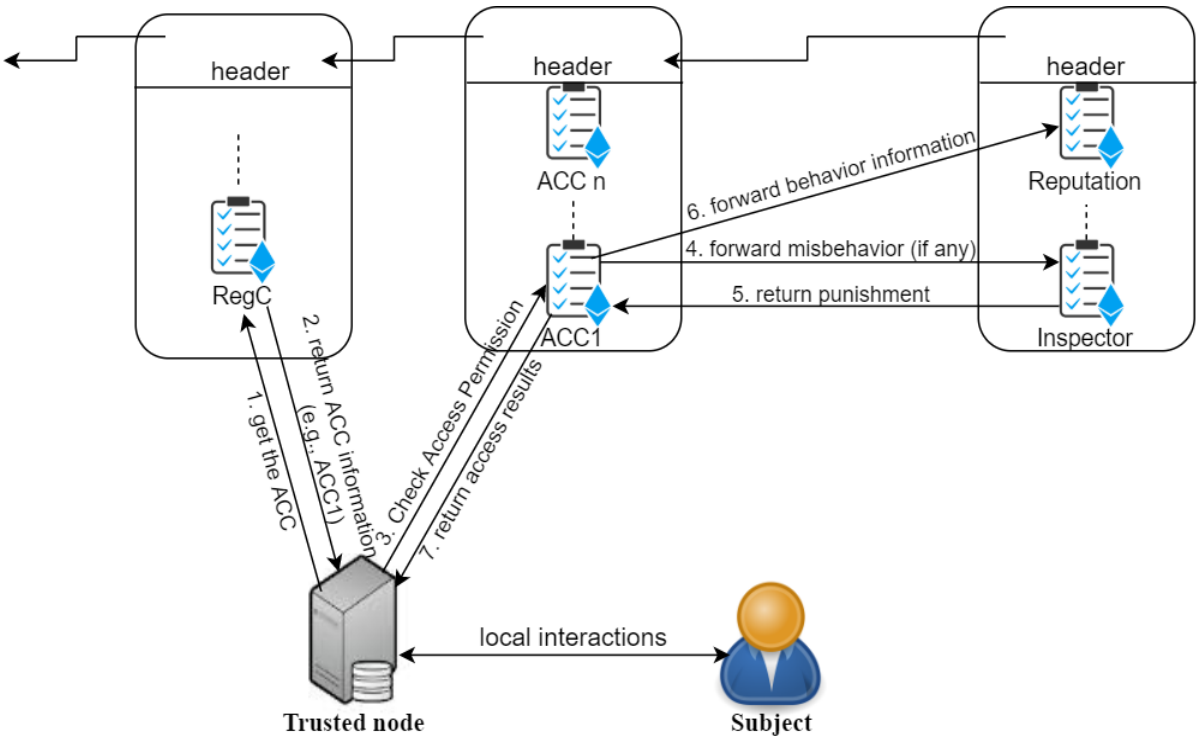

FIGURE 2.7 – Architecture d'un RSD basé sur Blockchain [55].

Dans les systèmes commerciaux, l'apport le plus populaire de Blockchain aux RS est la récompense des utilisateurs. Des RS comme SteemIt [2], memo.cash [3]et Minds [4] sont des exemples de cette tendance.

2. https ://steemit.com
3. https ://memo.cash
4. https ://www.minds.com

2.3.1 Tawki

Dans cette section, nous présentons Tawki [54], une architecture décentralisée pour la communication sociale. Grâce à cette architecture, les utilisateurs restent entièrement souscontrôle de leurs données personnelles, qui sont stockées et gérées par des systèmes de stockage de données personnelles. Chaque espace de stockage de données est accessible via une API Tawki unique, qui permet aux utilisateurs d'envoyer et de recevoir des données à partir des stockages de données personnelles d'autres utilisateurs. Selon cette approche, la communication sociale est de type P2P, donc aucune entité tiers ne contrôle ou surveille le processus. Tawki utilise l'Ethereum Name Service (ENS) pour la gestion des identités des utilisateurs et pour la résolution des identifiants vers le répertoire de stockage personnel de l'utilisateur respectif (la découverte du contenu des autres utilisateurs). Grâce à la fonction d'immuabilité de la Blockchain Ethereum (impossibilité de modification), la gestion des identités et la découverte des stockages de données personnelles sont protégées contre la censure et le contrôle par un tiers.

Afin de découvrir les contenus des différents utilisateurs, Tawki s'appuie sur la nature décentralisée de la Blockchain Ethereum. En utilisant le service ENS, les utilisateurs peuvent enregistrer un nom d'utilisateur, qui peut ensuite être résolu au stockage de données du propriétaire respectif. De cette façon, l'identité et la gestion des données sont contrôlées par les utilisateurs eux-mêmes sans qu'il soit nécessaire de recourir à une entité ou à une organisation centrale.

• **Stockage des données personnelles**

Pour assurer la protection des données, le contenu généré par l'utilisateur est uniquement stocké dans une base de données personnelle afin de garantir un contrôle total sur le contenu créé. Ce serveur central n'offre pas seulement un stockage de données mais fournit également une API pour interagir avec le client et échanger des messages avec les stockages de données personnelles des autres utilisateurs. Les utilisateurs peuvent soit déployer une instance sur leur propre appareil, soit utiliser le service d'une entité central de confiance. Comme le contenu personnel est uniquement stocké sur un appareil contrôlé par l'utilisateur et distribué à d'autres utilisateurs lorsque cela est voulu, il peut être supprimé ou modifié par son propriétaire à tout moment, tout en permettant à ce dernier de définir des politiques de contrôle d'accès précises pour les autres utilisateurs.

- **Demande d'ami**

Comme la découverte et la communication avec des amis et d'autres utilisateurs est un aspect clé des RS, ce processus doit être transparent pour les utilisateurs. Tawki conserve les caractéristiques décentralisées des RS dans sa technologie tout en fournissant des noms lisibles par l'homme pour identifier les utilisateurs. La procédure de création de liens dans Tawki est illustrée pour deux utilisateurs, Alice et Bob, dans la figure suivante :

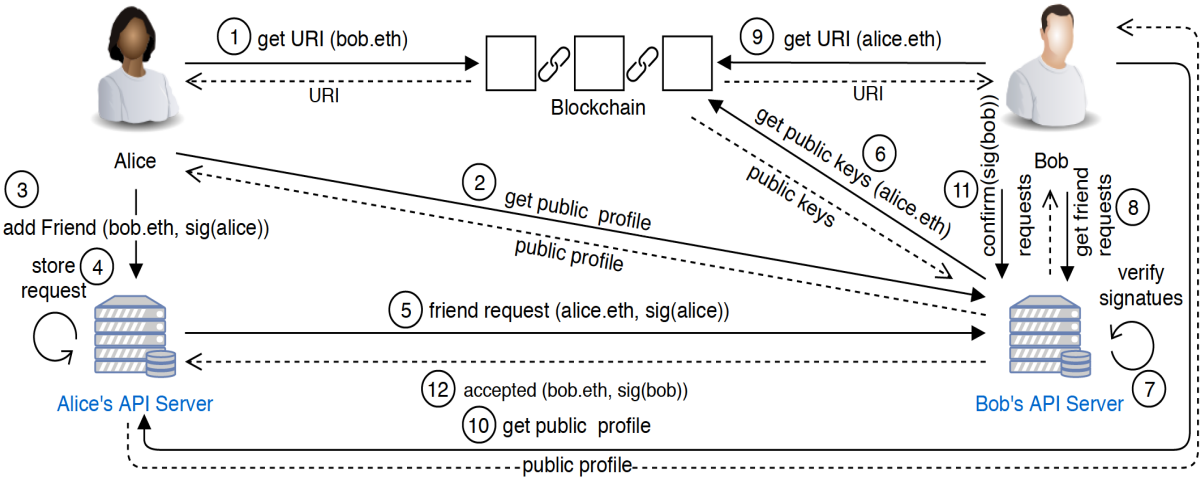

FIGURE 2.8 – Envoi d'une demande d'ami avec Tawki [54].

2.3.2 SteemIt

SteemIt est une plateforme de réseau social basé sur la technologie Blockchain [57], avec plus de 1,4 million d'utilisateurs [58], où chacun peut recevoir une récompense pour la création et la conservation de contenu, sous la forme de la crypto-monnaie Steem, et cela se fait par l'interaction des utilisateurs sur le contenu créé par des votes.

Les utilisateurs de SteemIt peuvent créer et partager des contenus sous forme de blogs. Les autres utilisateurs peuvent répondre à une publication, la partager ou voter. En fonction de la valeur des votes obtenus, les publications sont classées, et les publications les mieux classées sont placées en première page.

SteemIt vise à soutenir les réseaux sociaux et les communautés en ligne en offrant une grande partie de sa valeur aux personnes qui apportent des contributions de qualité, en les récompensant avec une crypto-monnaie, et grâce à ce processus, SteemIt a pu créer une monnaie capable d'atteindre un large public, y compris les personnes qui n'ont pas encore participé à une économie de crypto-monnaie [59].

Parmi les principes clés utilisés pour guider la conception de SteemIt, le plus important est que chaque personne qui contribue à une entreprise doit recevoir une part équitable de la valeur de la propriété, du paiement ou de la dette de l'entreprise. Le deuxième principe est que toutes les formes de capital ont la même valeur. Cela signifie que ceux qui consacrent leur temps et leur attention à la production et à la conservation de contenus pour d'autres ont la même valeur que ceux qui apportent leurs faibles ressources financières. Le troisième principe est que la communauté crée de la valeur pour servir ses membres. C'est-à-dire qu'elle serve ses membres plutôt que de vendre des produits ou des services à des personnes extérieures à la communauté [59].

La différence entre SteemIt et les autres plateformes est qu'il existe trois types d'unités monétaires différentes [57] : Steem, Steem Power (SP) et Steem Dollars (SBD).

— **Le Steem :** est l'unité qui est achetée et vendue pour de l'argent réel sur les marchés ouverts. Elle représente la principale crypto-monnaie du réseau.

— **Le Steem Power :** est une sorte d'investissement à long terme car les gens ne peuvent pas vendre cette unité pendant deux ans. Celui qui possède les unités de Steem Power est également propriétaire du réseau. En effet, 90% de la nouvelle unité Steem produite chaque jour est distribuée parmi ceux qui détiennent déjà des unités Steem Power. En outre, plus un utilisateur possède des unités de Steem Power, plus son vote comptera.

— **Le Steem Dollar :** est une monnaie stable qui ne perd jamais sa valeur, et les gens peuvent la vendre à tout moment. Le concept principal est que la communauté doit être récompensée pour la valeur qu'elle apporte.

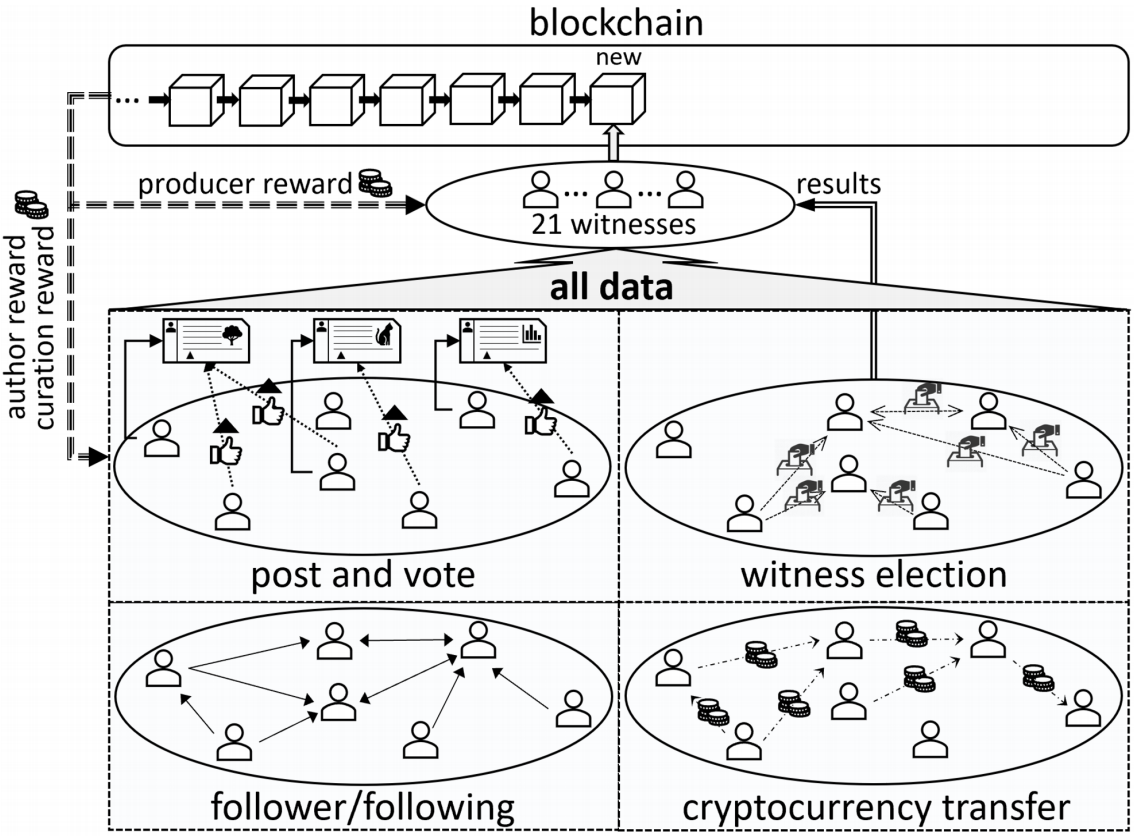

FIGURE 2.9 – Aperçu de la Blockchain Steem [60].

SteemIt utilise la Blockchain Steem pour stocker les données de base de la plateforme sous forme de Blockchains. Toutes les trois secondes, un nouveau bloc est produit, qui comprend toutes les opérations confirmées effectuées par les utilisateurs au cours des trois dernières secondes. SteemIt permet à ses utilisateurs d'effectuer plus de trente types d'opérations différentes. Dans la figure 2.9, nous présentons quatre catégories d'opérations qui sont les plus pertinentes pour le système. Tandis que les fonctions "poster/voter" et "suiveur/suivre" sont des fonctionnalités communes offertes par les RS, les opérations telles que le vote et le transfert de crypto-monnaies sont des fonctionnalités spécifiques aux Blockchains [60].

2.4 Conclusion

Nous avons présenté dans ce chapitre le concept des réseaux sociaux décentralisés à la base de la technologie Blockchain, ainsi que l'apport révolutionnaire de cette technologie sur les RS, dans ce cadre, on a illustré avec des exemples pertinents. Le chapitre suivant va présenter une conception et une analyse de notre système cible.

Chapitre 3

Conception

Chapitre 3

Conception

3.1 Introduction

Notre objectif et de proposer une nouvelle architecture d'un réseau social basé sur la technologie Blockchain. Notre idée clé est l'utilisation cette technologie pour sécuriser les données des utilisateurs qui sont stockés dans un espace choisit par l'utilisateur. Notre recherche a conduit vers une plateforme qui coïncide avec notre idée. C'est la plateforme Blockstack qui constitue une infrastructure fiable pour développer des différentes classes d'applications décentralisées. Notre projet s'est transformé à la conception et l'implémentation d'un réseau social décentralisé dans l'environnement Blockstack. Dans la première partie de ce chapitre, on va présenter la plateforme Blockstack. Ensuite, on va modéliser notre réseau social décentralisé qui utilise Blockstack comme une infrastructure de base.

3.2 Objectifs de l'application proposée

Afin de faciliter la communication et la collaboration sur le web sans qu'il soit nécessaire qu'une entité centrale contrôle ou surveille nos données, nous proposons notre nouvelle application de réseau social, une architecture de service décentralisée pour la communication sociale basée sur la technologie Blockchain, qui permet aux utilisateurs de partager tout type de fichier avec leurs amis et en gardant le contrôle total de leurs propres données et de leurs identités. La majorité des RS existants utilisent une infrastructure centralisée où les données partagées sont sous le contrôle du fournisseur de services sur un nombre

limité de serveurs comme Facebook , Twitter ...

Bien que les architectures de services P2P facilitent les applications et les services sans entités de contrôle centrales, la fiabilité et l'authenticité des informations gérées sont difficiles à garantir à cause des faibles performances des machines des utilisateurs. C'est pour cela que nous avons opté pour l'utilisation d'un autre moyen pour le stockage des données, qui est le Cloud, pour profiter de ses performances.

En terme général, notre application se présente comme un nouveau réseau social décentralisé basé sur la technologie Blockchain, qui a pour objectif de permettre à l'utilisateur de partager tout type de fichier avec ses amis tout en gardant le contrôle sur ses données et ses relations, et qui utilise le Cloud et ses performances comme un moyen de stockage des données. Elle utilise la technique Blockchain pour sécuriser l'accès aux données dans l'espace de stockage et les processus d'authentification et d'inscription, et assurer une distribution des noms décentralisée et une parfaite gestion des identités.

Il existe des travaux connexes, des RSD basés sur la technologie Blockchain qui offrent des services similaires à ceux de notre application comme Tawki [54], SteemIt...

3.3 Architecture globale de l'application

L'idée de notre application est de concevoir un réseau social décentralisé basé sur la technologie Blockchain, afin de permettre à l'utilisateur de partager tout type de fichier tout en garantissant la sécurisation de ses propres donnés et ses relations, et aussi son identité. Avec des performances comparables à celles des applications centralisées.

Dans cette section, nous présentons Blockstack, une plateforme qui offre une nouvelle génération d'applications où les développeurs et les utilisateurs peuvent interagir de manière fiable et sécurisée, l'architecture de Blockstack coïncide avec notre système cible, elle nous offre un environnement de développement souple qui nous permet de réaliser notre application avec ses caractéristiques souhaitées.

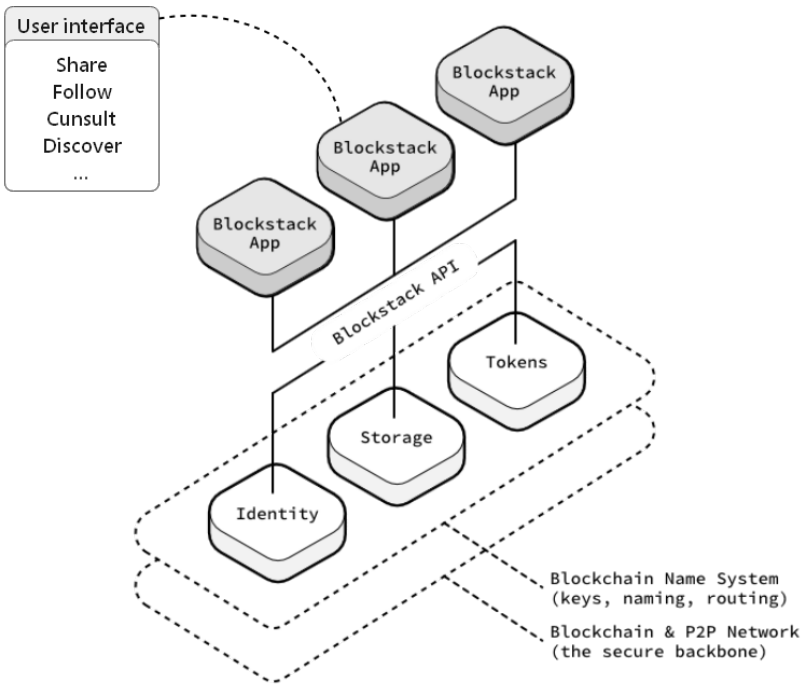

FIGURE 3.1 – Superposition de l'application sur la plateforme Blockstack.

Blockstack gère tout, de l'identité et de l'authentification au stockage de données, il offre un API pour connecter et pour lire et écrire dans le système de stockage Gaia de Blockstack.

3.4 Présentation de Blockstack

Lancé en 2017, Blockstack est une plateforme informatique décentralisée qui permet aux utilisateurs de garder le contrôle de leurs données et de leur identité [61]. Les bibliothèques Blockstack permettent aux développeurs de créer des applications décentralisées [62]. Blockstack propose une nouvelle infrastructure pour l'internet où les utilisateurs choisissent les données à partager et avec qui les partager [61]. Blockstack fournit des protocoles décentralisés pour l'authentification et le stockage des données[63]. Grâce à la conception de leur plateforme, les développeurs d'applications ne peuvent pas accéder aux données des utilisateurs, les utilisateurs peuvent choisir qui stocke leurs données, et les droits de lecture ou d'écriture sur les données sont décidés aussi par l'utilisateur [61].

Blockstack utilise la Blockchain pour l'enregistrement des noms pour sécuriser le stockage dans le Cloud. Le stockage des données se fait hors de la Blockchain dans un fournisseur de stockage exécutant le système de stockage Gaia de Blockstack. Gaia est simple et fiable et utilise les infrastructures Cloud existantes. Cette architecture d'application de base signifie que toute application peut fonctionner et évoluer comme elle le fait sans une Blockchain [64], elle permet aux applications d'offrir une expérience d'utilisateur similaire aux applications traditionnelles. Elle permet de bénéficier d'une performance élevée grâce au système de stockage Cloud et elle garantit la sécurité des données grâce à la technologie Blockchain.

Afin de donner à l'utilisateur le contrôle de ses données et d'associer strictement ses données à l'identité de l'utilisateur, Blockstack a fourni le système de stockage décentralisé (Gaia) et le système de dénomination de la Blockchain (BNS) ou « Blockchain Naming System ». L'utilisateur peut se connecter à une application de Blockstack en utilisant l'identité numérique fournie par le système BNS.

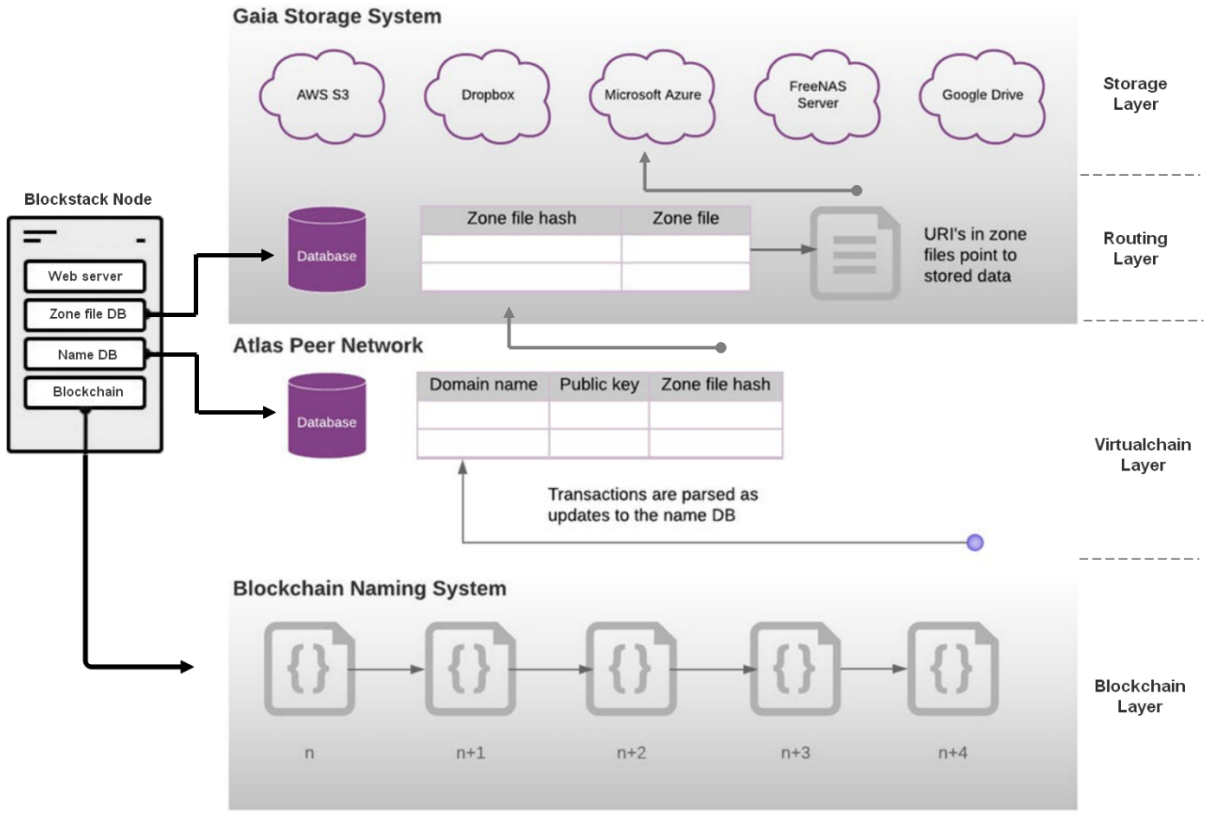

FIGURE 3.2 – Aperçu de l'architecture de Blockstack.

3.4.1 Stockage

Le réseau Blockstack stocke les données des applications à l'aide d'un système de stockage appelé Gaia [64], qui offre des performances comparables à celles des fournisseurs de services de Cloud centralisés, et qui utilise des fournisseurs de stockage dans le Cloud (comme Dropbox, Amazon S3 et Google Drive). Les métadonnées des transactions sont stockées sur la Blockchain Blockstack et les données d'application utilisateur sont stockées dans le stockage Gaia [64] . Les fournisseurs de stockage dans le Cloud, comme Dropbox, n'ont aucune visibilité sur les données des utilisateurs, ils ne voient que des blocs de données cryptées. Le stockage des données en dehors de la Blockchain garantit que les applications du réseau Blockstack peuvent fournir aux utilisateurs des performances élevées et une grande disponibilité pour la lecture et l'écriture des données sans introduire de parties centrales de contrôle.

L'approche de Gaia en matière de décentralisation se concentre sur le contrôle des données par les utilisateurs et leur stockage. Les utilisateurs peuvent choisir un fournisseur de hub Gaia.

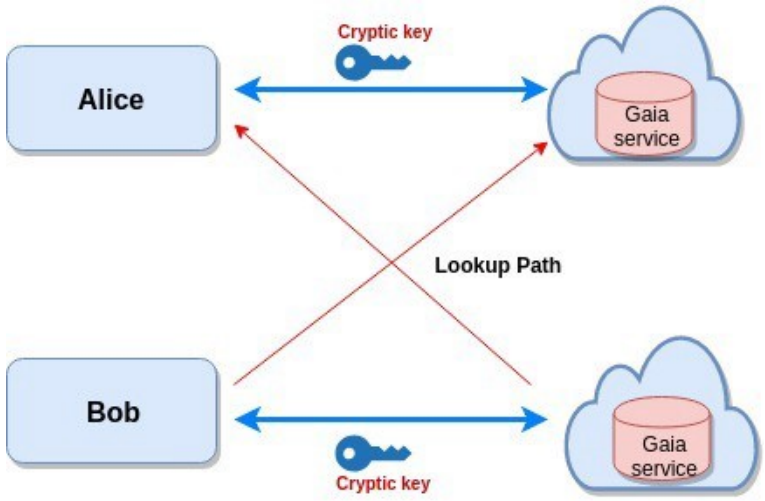

FIGURE 3.3 – Comment un client interagit avec le stockage Gaia.

Les données de l'utilisateur seront fortement couplées avec la clé publique de l'utilisateur. Les applications liront et écriront les données au Gaia hub pour le compte d'un utilisateur (si et seulement si l'utilisateur le permet). Toutes les données de l'utilisateur seront transférées à son Gaia hub. Le Gaia hub de l'utilisateur peut être la propriété de l'utilisateur lui-même ou il peut utiliser l'espace de stockage par défaut fourni par Blocks-

tack. Par défaut, un hub est utilisé pour stocker les données de l'utilisateur cryptées par la clé publique de l'utilisateur. De cette façon, les fournisseurs de stockage ne voient que des blobs de données.

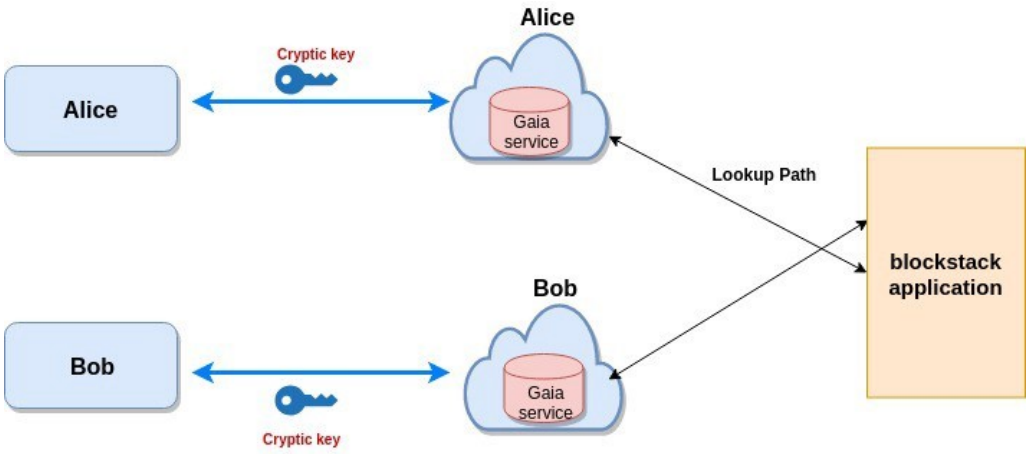

FIGURE 3.4 – Interaction entre une application Blockstack et le service Gaia.

3.4.2 Authentification et identité

3.4.2.1 Authentification

Blockstack fournit une connexion et une authentification uniques sans tiers ni serveurs distants. L'authentification Blockstack est un système d'authentification par jeton. Du point de vue d'un utilisateur d'application, la connexion est similaire aux techniques d'authentification habituelles. Pour un développeur d'applications, le processus est différent du processus client-serveur traditionnel des services de connexion centralisés. Avec Blockstack, le processus d'authentification se déroule entièrement du côté client.

Une application décentralisée (DApp) et l'authentificateur Blockstack communiquent pendant le processus d'authentification en faisant passer deux jetons dans les deux sens. L'application requérante envoie à l'authentificateur Blockstack un jeton *authRequest*. Une fois que l'utilisateur a approuvé la connexion, l'authentificateur Blockstack répond à l'application avec un jeton *authResponse*. Ces jetons sont des jetons Web JSON et sont transmis via des "strings" de requête URL. Lorsqu'un utilisateur choisit de se "connecter avec Blockstack" sur la DApp, la méthode *redirectToSignIn()* envoie l'utilisateur à l'authentificateur Blockstack. Le navigateur répond avec un identifiant de connexion et une clé privée de l'application[64].

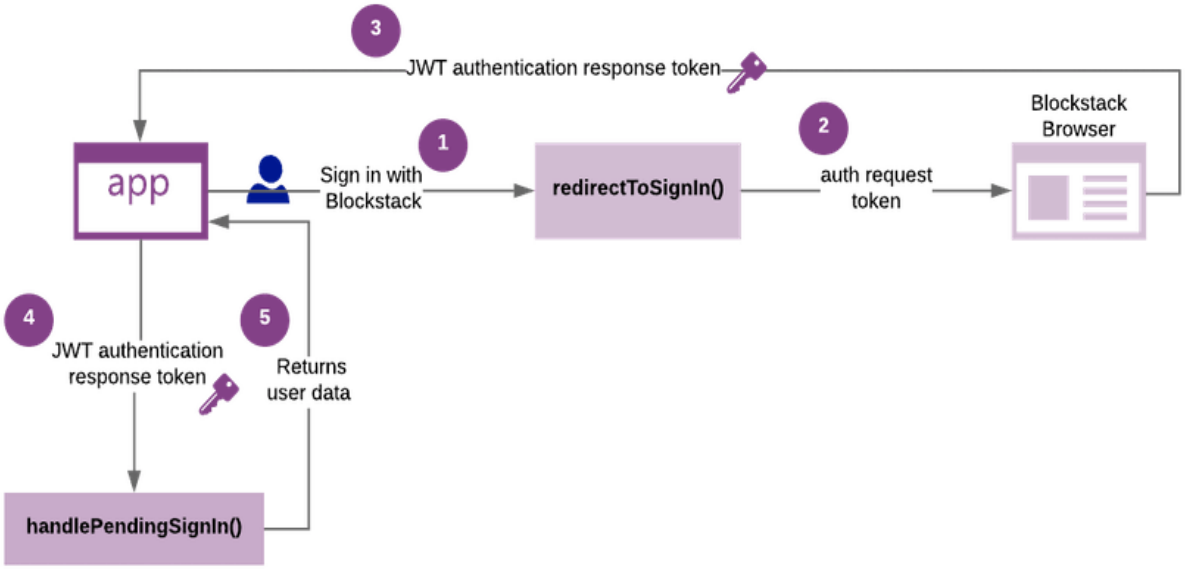

FIGURE 3.5 – Processus d'authentification d'une application avec Blockstack (géré côté client)[64].

La clé privée de l'application est spécifique à l'application. Elle est générée à partir de la clé privée de l'adresse d'identité de l'utilisateur en utilisant le domaine de l'application comme entrée. Cette clé est déterministe, ce qui signifie que pour un ID de Blockstack et un nom de domaine donnés, la même clé privée est générée à chaque fois. La clé privée de l'application est partagée de manière sécurisée avec l'application à chaque authentification et est cryptée par l'authentificateur du Blockstack. La clé sert trois fonctions [64] :

- Elle est utilisée pour créer les références qui permettent à une application d'accéder au stockage du Gaia hub pour cette application spécifique
- Elle est utilisée pour le cryptage de bout en bout des fichiers stockés pour l'application sur le Gaia hub de l'utilisateur
- Elle sert comme une clé secrète cryptographique que les applications peuvent utiliser pour effectuer d'autres fonctions cryptographiques

Lorsqu'une application écrit sur un hub Gaia, le jeton d'authentification, la clé et les données sont transmis au Gaia hub.

3.4.2.2 Identité

Un nom ou une identité, ou plus simplement ID, est le domicile d'une personne sur Blockstack. Une identité est unique, comme un numéro de passeport, l'utilisateur seul la

possède. Pour utiliser une application, l'utilisateur doit inscrire dans l'application avec son identité et une clé secrète que lui est seul à connaître. Les données et l'endroit où elles sont stockées sont liés à l'identité de l'utilisateur. L'utilisateur peut dire aux gens son identité tout comme il leur dit son nom. Ce qu'il doit sécuriser et protéger, c'est sa clé secrète.[64]

L'utilisateur peut obtenir une identité gratuite de Blockstack dans l'espace de noms *id.blockstack*. Un espace de noms est similaire à un domaine (wikipedia.com, par exemple) sur l'ancien internet. Ainsi, une identité gratuite a un suffixe Blockstack dans son nom, par exemple, *Riadh.id.blockstack* est une identité Blockstack gratuite.[64]

Si un utilisateur oublie son identité ou s'il perd sa clé secrète, aucune autre personne, ni logiciel, ne peut l'aider à récupérer son identité. Cette restriction protège son identité et sa clé secrète qui contrôlent ensemble l'accès à ses données. Personne d'autre que lui ne peut conserver et stocker ses données, pas même Blockstack.[64]

Lorsqu'un utilisateur crée une identité, son identifiant et sa clé privée sont hachés (cryptés) et enregistrés dans la Blockchain de Blockstack. Les données qu'il crée par le biais de son identité sont cryptées et conservées hors de la Blockchain dans son espace de stockage de données.[64]

3.4.3 Résolution des noms

L'Internet traditionnel utilise le système de noms de domaine (DNS) pour mapper les noms lisibles par l'homme aux adresses IP (qui donnent l'emplacement des nœuds et du contenu). DNS utilise des serveurs centralisés, ils peuvent être mis hors ligne par des attaques DDoS [65].

Blockstack a créé un remplacement décentralisé du DNS, c'est-à-dire d'un système qui lie les noms lisibles par l'homme aux données de découverte (adresses IP), mais qui ne possède aucun point central de contrôle, un nouveau système de dénomination décentralisé basé sur la technologie Blockchain appelé *Blockstack Naming Service* (BNS).

BNS peut être utilisé pour résoudre les noms en adresses IP, mais ce n'est pas son cas d'utilisation par défaut. Le BNS se comporte plutôt comme un système LDAP (*Light-weight Directory Access Protocol*) décentralisé pour résoudre les noms des utilisateurs en données d'utilisateurs [64].

Avant les Blockchains, les systèmes de dénomination ne permettaient que deux de ces

trois propriétés (le nom est unique, lisible par l'homme et décentralisé)[66] , jamais les trois à la fois. Cette limitation est appelée *Triangle de Zooko* [67], le BNS garantit ces trois propriétés à la fois.

BNS intègre un journal de ses messages de plan de contrôle dans une Blockchain publique, comme Bitcoin. Chaque pair BNS détermine l'état de chaque nom en indexant ces transactions spécialement conçues. Ce qui fait que chaque pair calcule indépendamment le même état de nom global.

En interne, un nœud BNS implémente une base de données de noms répliqués. Chaque nœud BNS se synchronise avec tous les autres nœuds du monde, de sorte que les requêtes sur un nœud BNS seront les mêmes sur les autres nœuds. Les nœuds BNS permettent au propriétaire d'un nom de lier à son nom jusqu'à 40Kb de données d'état en dehors de la chaîne, qui seront répliquées à tous les nœuds BNS via un réseau P2P appelé Atlas [69].

3.4.4 Développements des applications en dessus du système

Blockstack est une plateforme de développement d'applications et n'est pas une application, il offre un environnement souple qui permet de développer tout type d'application avec des caractéristiques importantes (décentralisation, performance et sécurité élevés. . .).

Découvrir ici des exemples d'applications développées avec Blockstack dans le store qui similaire au Google Play Store [1] d'Android et App Store [2] d'Apple : https ://www.app.co/

Le développement des applications décentralisées sur Blockstack est réalisé grâce aux API qui permettent aux développeurs de communiquer avec Blockchain pour l'authentification des utilisateurs et avec Gaia hub pour la lecture et l'écriture sur leurs données. Les applications sont indépendantes (même si un utilisateur utiles plusieurs applications, une application donnée ne peut lire et écrire que sur les données qui sont générées par elle-même).

3.5 Fonctionnalités de l'application

Notre application se présente comme un réseau social décentralisé, qui offre plusieurs fonctionnalités, permettant à l'utilisateur de bénéficier d'une bonne expérience tout en

1. https ://play.google.com/store/apps
2. https ://www.apple.com/app-store

gardant le control total sur ses données et en garantissant la sécurité de ses propres données et ses relations.

Dans cette section, nous présentons les fonctionnalités de base de notre application (le partage, la découverte et l'bonnement).

3.5.1 Partage

C'est la fonctionnalité principale de notre application, consiste à permettre aux utilisateurs de partager tout type de ficher (texte, image...) à travers l'application dans leur profil personnel.

Le contenu partagé se divise en trois catégories principales en matière de lisibilité (privé, publique et spécifique) :

- **Privé :** le contenu partagé est lisible par l'utilisateur seulement.
- **Publique :** le contenu partagé est lisible par tous les utilisateurs.
- **Spécifique :** le contenu est lisible par un nombre fini d'utilisateurs seulement (un groupe d'utilisateurs qui possèdent la clé du groupe).

3.5.2 Découverte

• **Découverte de contenu**

C'est la navigation ou la consultation d'un contenu partagé d'un ou de plusieurs utilisateurs par d'autres utilisateurs, un utilisateur a la possibilité de consulter un contenu publique d'un autre utilisateur ou son contenu spécifique s'il lui permet de le faire, mais ne peut pas consulter son contenu privé.

• **Découverte des profils**

Un utilisateur a la possibilité de découvrir d'autres profils des utilisateurs que les utilisateurs qu'il connaît ou bien la liste des utilisateurs qu'il suit, il peut avoir des suggestions présentées dans son profil pour suivre (abonner) d'autres personnes qui ont des relations ou des intérêts communs.

3.5.3 Abonnement

L'utilisateur peut suivre ou s'abonner (follow) d'autres utilisateurs et peut annuler cet abonnement à tout moment. La liste des abonnements est considérée comme une donnée

privé de l'utilisateur, elle est stockée dans son propre espace de stockage décentralisé, c'est à dire qu'elle n'est pas stockée dans un serveur central. L'utilisateur a le choix de partager cette liste ou pas, il peut aussi partager juste une partie de cette liste, le partage de la liste des abonnements permet de contribuer à l'opération de découverte ou dans l'algorithme de recommandation d'amis.

3.6 Structures de données et droits d'accès au contenu partagé

Le stockage des données dans le système Blockstack se fait sous forme de fichiers. L'interface du système permet aux utilisateurs de consulter les fichiers des autres utilisateurs mais ne permet pas de lister les fichiers. On a donc besoin d'ajouter d'autres fichiers index qui peuvent être des fichiers TXT, XML ou JSON... et qui possèdent des noms prédéfinit. Ces fichiers sont utilisés pour permettre la découverte du contenu.

On peut trouver des différents types de fichiers dans l'espace de stockage de chaque utilisateur tels que :

- Un fichier pour le contenu public d'un utilisateur qui est accessible par tous les utilisateurs, sous le nom Pubindex, et qui contient la liste de toutes les publications publiques partagées par l'utilisateur (exp : pub1, pub2...).
- Un fichier pour le contenu privé d'un utilisateur qui est crypté et accessible seulement par l'utilisateur lui-même, sous le nom Privindex, et il contient la liste de toutes les publications privées partagées par l'utilisateur (exp : priv1, priv2,...etc.).
- Un fichier LocalGroupindex qui contient les noms et les clés de cryptage des groupes créés par l'utilisateur. La structure de ce fichier est sous la forme « nom de groupe/clé de groupe » (exp : G1-key, G2-key...), la clé est générée lors de leur création est communiqué aux membres du groupe par une voie sécurisée (mail,...etc.). Le contenu des fichiers partagés dans chaque groupe est crypté par sa clé spécifique. Seuls les utilisateurs qui possèdent cette clé peuvent accéder à ces données.
- Un fichier NomGroupindex pour chaque groupe (exp : G1index, G2index,...). Ces fichiers contiennent chacun la liste de toutes les publications d'un groupe donné. Exemple : G1index contient la liste des publications du groupe G1 (G1file1,

G1file2,. . .).

— Un autre fichier ExternGroupindex qui contient la liste des groupes crées par d'autres utilisateurs dont l'utilisateur actuel est membre, sous la forme « nom de groupe/ID de créateur/clé de groupe » (exp : G1-hostid-key, G2-hostid-key,...etc.). Ce fichier permet à l'utilisateur de consulter le contenu de ces groupes sans être obligé de demander la clé secrète à chaque fois.

— Un fichier pour la liste des amis abonnés par l'utilisateur (followed), c'est un fichier privé accessible seulement par l'utilisateur lui-même, donc il est crypté par sa clé publique, sous le nom myFollowed.txt, il contient une liste des identifiants (ID) des personnes qu'il suit.

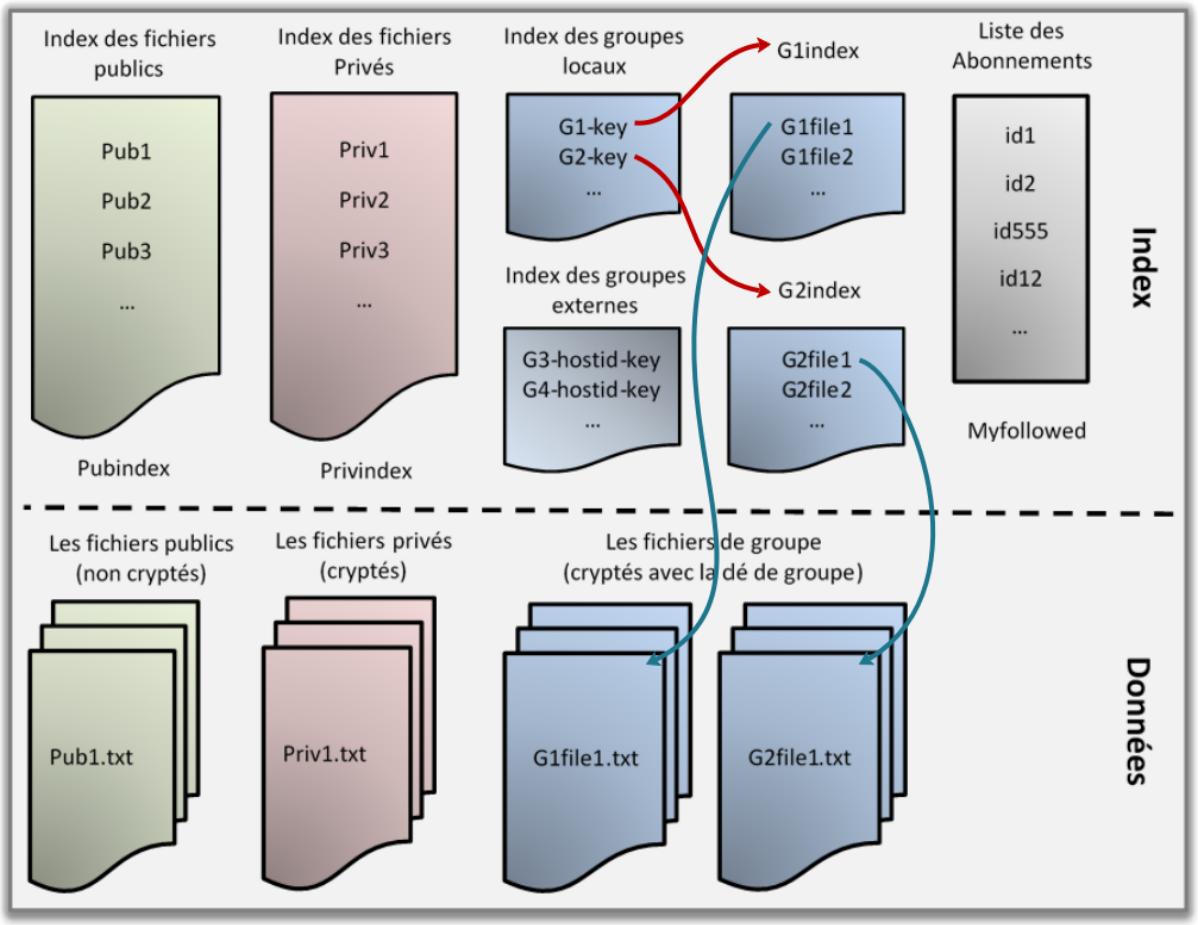

FIGURE 3.6 – Structure des données sur l'application.

3.7 Modélisation

Cette section sera consacrée à la modélisation de notre application, pour cela, nous avons choisi le langage de modélisation UML (Unified Modeling Language) que nous avons choisi pour ses divers avantages (modularité, abstraction, structuration cohérente des fonctionnalités et des données, diversité des diagrammes...).

3.7.1 Vue fonctionnelle

Après avoir identifié les acteurs qui interagissent avec le système, nous y développons un premier modèle UML de haut niveau, pour pouvoir établir précisément les frontières du système. Dans cette optique, nous allons identifier les cas d'utilisation et nous allons construire un diagramme général reliant les acteurs et les cas d'utilisation. Ensuite, nous précisons le point de vue fonctionnel en détaillant les différentes façons dont les acteurs peuvent utiliser le système.

3.7.1.1 Diagramme de cas d'utilisation

Il montre les interactions fonctionnelles entre les acteurs et le système à l'étude.

• **Identification des acteurs**

— **Internaute :** Un simple utilisateur qui fait juste la consultation de l'application et il n'a pas accès aux services contenues dessus.

— **Utilisateur :** C'est le consommateur qui bénéfice des services offerts par l'application (découverte des profils, partage, navigation, abonner/désabonner, réaction sur les publications, définir la confidentialité d'une publication).

— **Administrateur :** C'est la personne responsable de la gestion de système (gestion des comptes et des services, mise à jour de l'application).

— **Blockstack :** c'est le système Blockstack, qui est responsable de la gestion de l'inscription et l'authentification et la sécurisation de stockage.

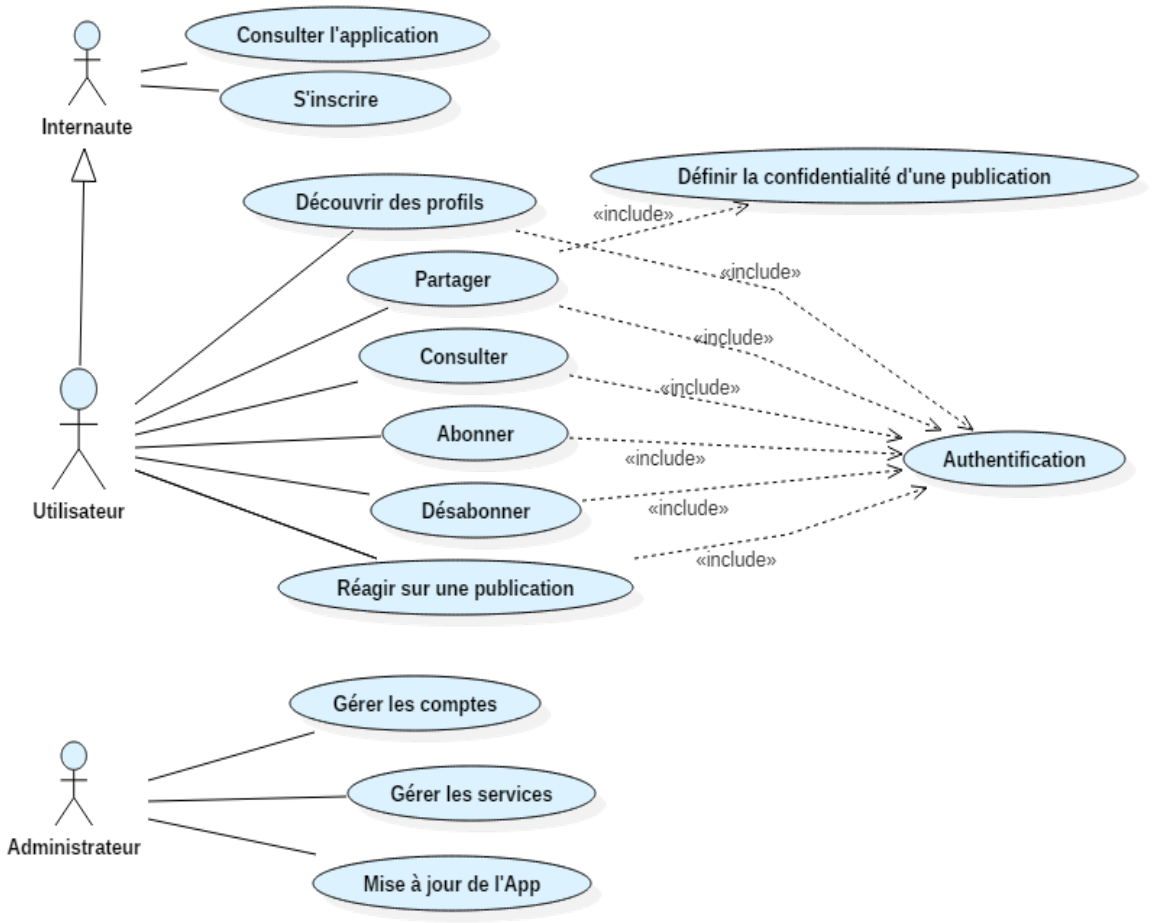

FIGURE 3.7 – Diagramme de cas d'utilisation général.

3.7.2 Vue dynamique

Cette partie va nous permettre d'illustrer pas à pas, à partir d'une nouvelle étude de cas, les principaux concepts et diagrammes UML pour le point de vue dynamique.

3.7.2.1 Diagrammes de séquence

Il représente séquentiellement le déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs. Il permet également de montrer les interactions d'un système avec son environnement ainsi qu'il modélise un système de manière dynamique et ils s'attachent principalement à montrer la circulation et l'ordre chronologique des messages, autrement dit, ils décrivent la circulation de l'information[68].

3.7.2.2 Réalisation des diagrammes de séquences

Nous allons maintenant présenter quelques diagrammes de séquence qui sont les plus utiles pour la suite de notre travail.

A. Diagramme de séquence pour la création d'un compte (Inscription)

La création d'un compte est la 1ère étape à réaliser par les acteurs précisément par l'internaute afin d'atteindre son but et cela se fait par les étapes suivantes :

— L'internaute demande le formulaire d'inscription.
— L'application envoi une requête d'inscription au système Blockstack et ce dernier renvoi une requête de réponse.
— L'application affiche un formulaire d'inscription.
— L'internaute demande une clé secrète, l'application envoi une requête au système Blockstack pour générer une clé secrète.
— Le système Blockstack répond à l'application avec une clé secrète générée, et qu'elle l'affiche à son tour à l'internaute et lui demande de la copier.
— L'application demande à l'internaute de choisir un nom d'utilisateur et envoi ce nom au système Blockstack pour la vérification.
— Si ce nom existe déjà, l'internaute doit choisir un autre nom d'utilisateur, sinon, le système enregistre les données d'inscription et l'internaute peut accéder à son compte utilisateur.

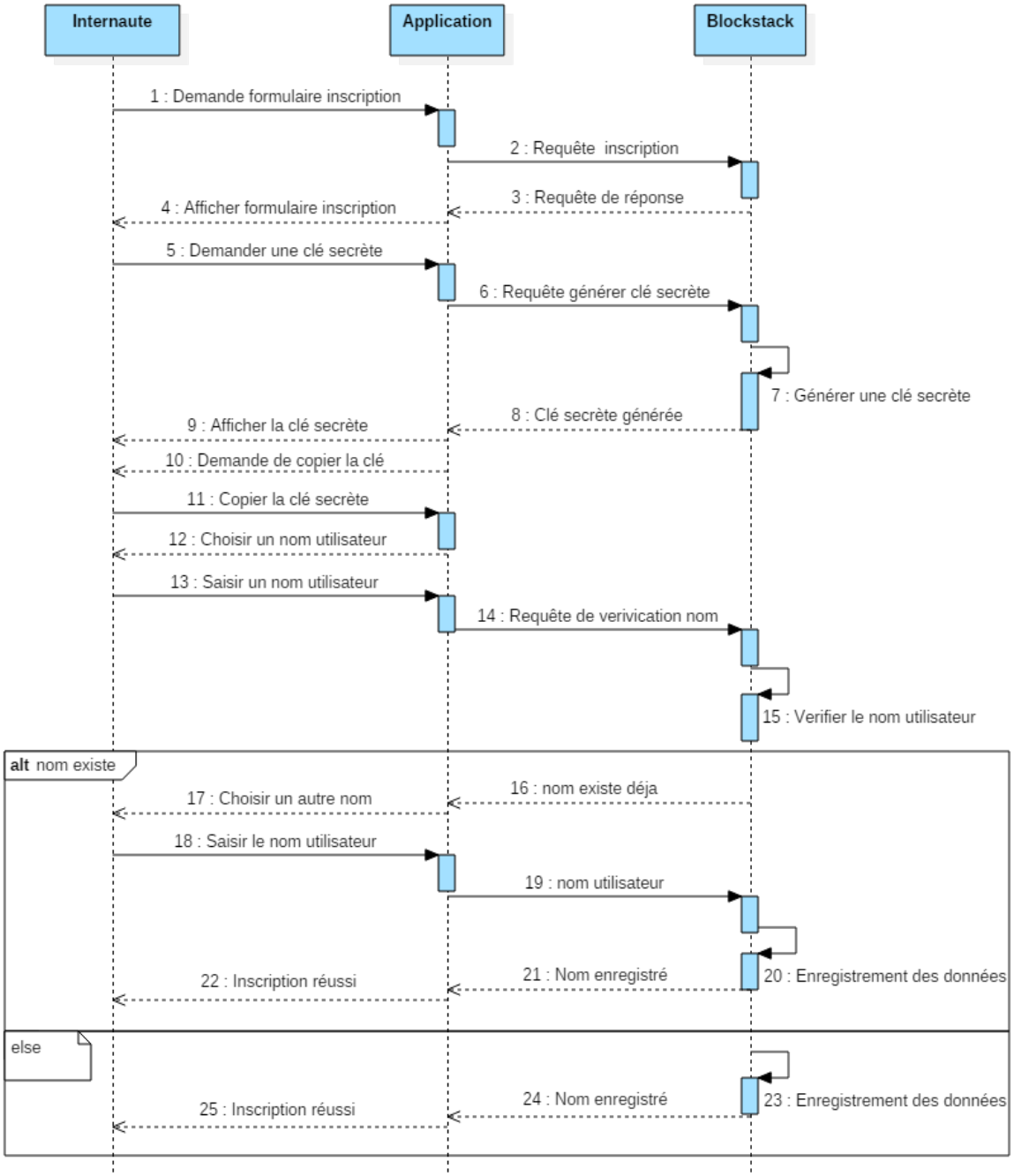

FIGURE 3.8 – Diagramme de séquence pour la création d'un compte (inscription).

B. Diagramme de séquence pour l'authentification.

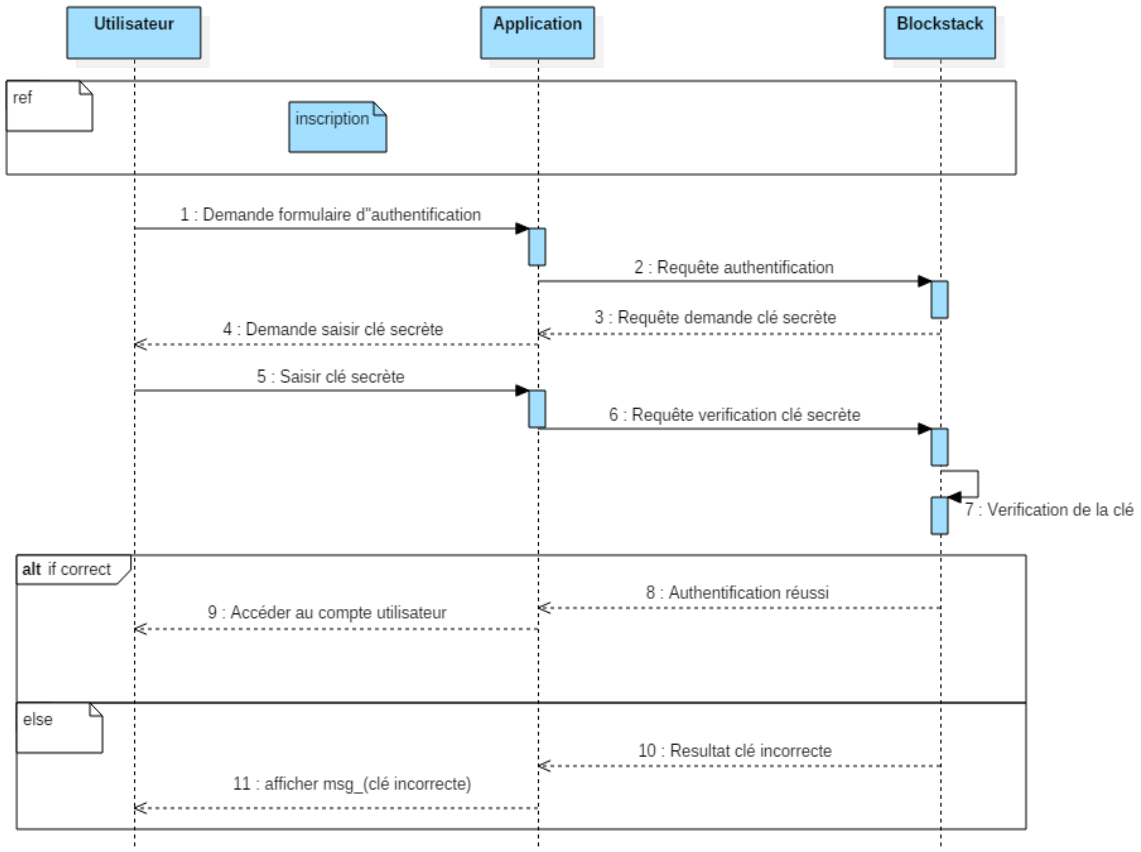

FIGURE 3.9 – Diagramme de séquence pour l'authentification.

C. Diagramme de séquence pour le partage.

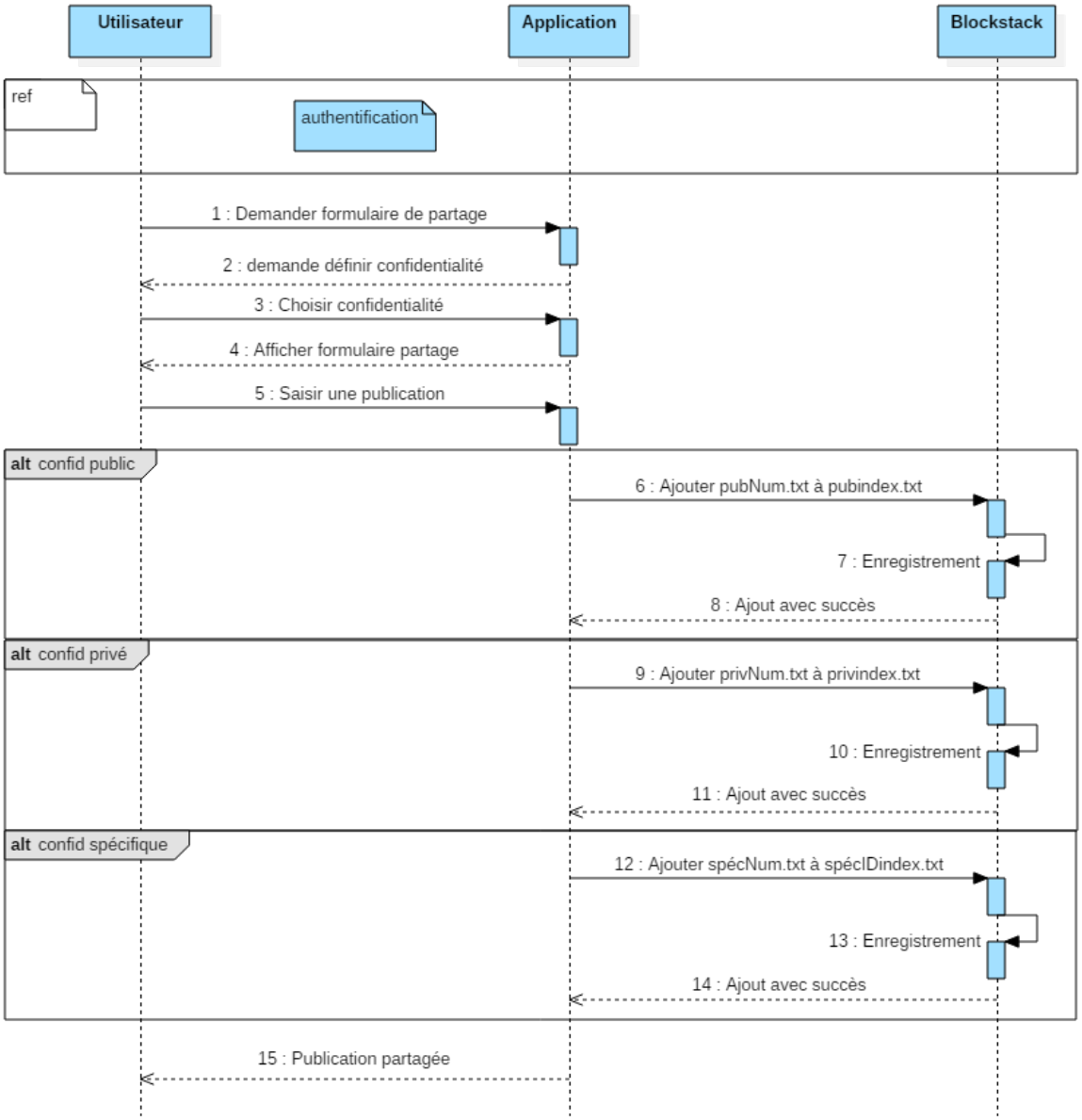

FIGURE 3.10 – Diagramme de séquence pour le partage.

D. Diagramme de séquence pour l'abonnement.

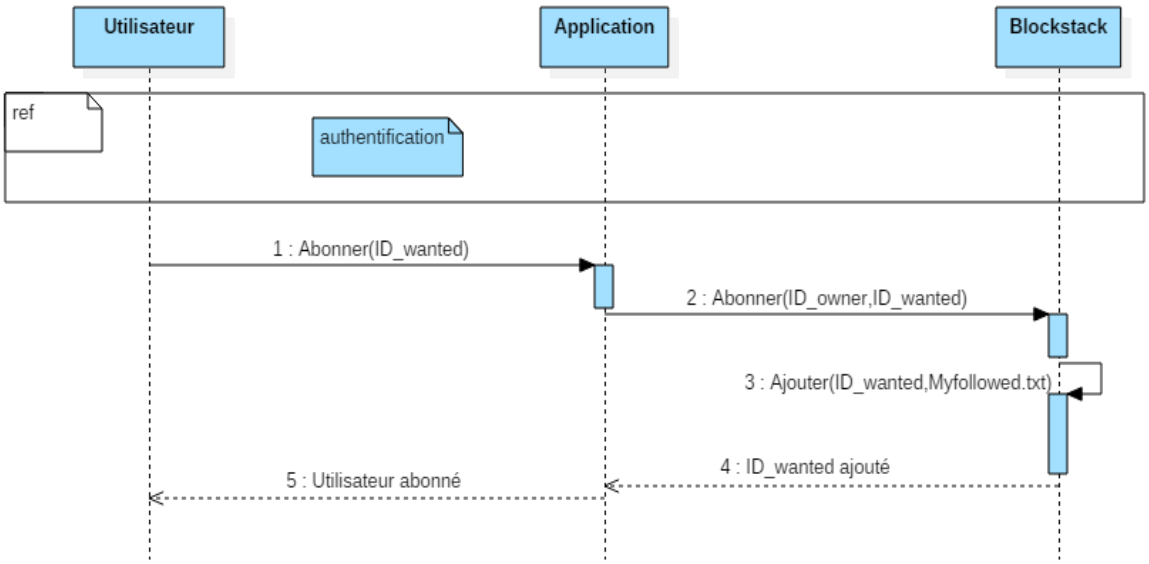

FIGURE 3.11 – Diagramme de séquence pour l'abonnement.

3.8 Conclusion

On a détaillé dans ce chapitre l'architecture proposée. Nous avons modélisé une application de partage social dans le cadre de la plateforme Blockstack (une plateforme qui profite de la performance du Cloud et la sécurité solide du Blockchain). Les structures de données et les interactions entre les différents acteurs sont présentées en détails. On va présenter dans le dernier chapitre les détails d'implémentation.

Chapitre 4

Implémentation

Chapitre 4

Implémentation

4.1 Introduction

Dans ce quatrième et dernier chapitre, on s'intéressera à la réalisation de notre application de réseau social basé sur la technologie Blockchain. Nous allons tout d'abord présenter les différents outils et langages utilisés pour la réalisation. Ensuite, nous présenterons les différentes fonctionnalités offertes par le système et cela sera illustré par la présentation des différentes interfaces de l'application.

4.2 Environnements de travail

4.2.1 Environnement matériel

Pour réaliser ce travail, on a utilisé un ordinateur DELL Latitude E5530, Intel Core i5-5300U, RAM 8 Go, Système d'exploitation Windows 7 Pro 64 bits.

4.2.2 Environnement logiciel

4.2.2.1 Visual Studio Code

Visual Studio Code est un éditeur de texte qui intègre la coloration syntaxique du code source pour les langages et fichiers C, C++, JAVA, C#, JavaScript, HTML, PHP, CSS,...etc. comme il nous offre :

— Plusieurs onglets pour éditer plusieurs fichiers dans la même fenêtre.

— Choix du Codage de caractères (ANSI, UTF-8, UCS-2).

— Glisser-déposer.

— Numérotation des lignes,...etc.

4.2.2.2 StarUML

StarUML est un logiciel de modélisation UML rapide, flexible, disponible en open source. Objectif du projet StartUML est de construire un outil de modélisation de logiciels et aussi la plate- forme qui est un remplacement convaincante d'outils UML commerciaux.

4.2.2.3 Github

GitHub est une plateforme de contrôle de version et de collaboration basée sur le web pour les développeurs de logiciels. De type SaaS (Software as a Service) [69], ce service en ligne est utilisé pour stocker le code source d'un projet et suivre l'historique complet de toutes les modifications apportées à ce code. Il permet aux développeurs de collaborer plus efficacement sur un projet en fournissant des outils pour gérer les changements éventuellement contradictoires de plusieurs développeurs.

4.2.3 Langages de programmation utilisés

4.2.3.1 Langage HTML

HTML (HyperText Markup Language) [70] a fait son apparition des 1991 lors du lancement du Web. Son rôle est de gérer et organiser le contenu des pages web. C'est donc en HTML que vous écrirez ce que vous souhaitez que la page affiche du texte, des liens, des images. Nous l'utiliserons pour l'élaboration des pages statiques qui constituent l'interface utilisateur.

4.2.3.2 Langage CSS

CSS (Cascading Style Sheets, aussi appelées Feuilles de style) [71] joue un rôle important dans la gestion et l'apparence de la page web (agencement, positionnement, décoration, couleur et taille du texte). Nous utiliserons ce langage pour rendre les pages programmées en HTML plus agréables et leurs donner un style spécifique.

4.2.3.3 Langage JavaScript

JavaScript [72] est un langage de programmation de scripts principalement employé dans les pages web. Par exemple, JavaScript permet de tester que les champs obligatoires sont bien remplis sans avoir besoin d'envoyer le formulaire au serveur pour qu'il le vérifie. L'avantage dans ce cas là est de limiter le nombre de requêtes envoyées au serveur.

4.2.3.4 Langage JSON

JSON (JavaScript Objet Notation) est un langage léger d'échange de données textuelles. Pour les ordinateurs, ce format se génère et s'analyse facilement [73]. Pour les humains, il est pratique à écrire et à lire grâce à une syntaxe simple et à une structure en arborescence. JSON permet de représenter des données structurées (comme XML par exemple).

4.2.4 Bibliothèques utilisées

4.2.4.1 Blockstack.js

La bibliothèque blockstack.js est l'interface de programmation fournie par Blockstack qui permet de développer des applications décentralisés fiables de manière souple et agile. On l'utilise dans l'application pour faire ce qui suit :

— Authentifier les utilisateurs et gérer les profils et les identités.

— Charger et stocker les données des utilisateurs.

— Consulter les données partagées par les utilisateurs dans l'application.

Voici quelques fonctions de base de la bibliothèque blockstack.js qu'on a utilisé dans notre application :

• **Authentification/inscription**

Pour authentifier les utilisateurs sur l'application, on utilise la fonction *redirectToSignIn()* qui envoie l'utilisateur à l'authentificateur Blockstack (la Blockchain) qui lui permet d'inscrire (créer un nouveau compte) ou de se connecter à son compte.

```
function SignIn() {
    blockstack.redirectToSignIn(
        redirectURI?: string,
        manifestURI?: string,["store_write", "publish_data"]);
 }
```

FIGURE 4.1 – Fonction d'authentification/inscription d'un utilisateur.

• **Connexion** Pour déconnecter de l'application, on utilise la fonction *signUserOut()*, qui prend un paramètre de type *string* qui présente le lien de direction après la déconnexion.

```
function signUserOut() {
  blockstack.signUserOut(redirectURL?: string)
}
```

FIGURE 4.2 – Fonction de déconnexion d'un utilisateur.

• **Lire/écrire sur le stockage Gaia hub**

Gaia est construit sur un modèle qui prend en charge de nombreux services de stockage. Ainsi, avec très peu de lignes de code, vous pouvez interagir avec des fournisseurs sur Amazon S3, Dropbox,...etc. Blockstack utilise les fonctions *putFile()* et *getFile()* pour interagir avec le stockage Gaia. *getFile()* est utilisée pour récupérer un contenu à partir de stockage Gaia et *putFile()* est utilisée pour écrire sur l'espace de stockage.

Par défaut, *putFile()* crypte les informations tandis que *getFile()* les décrypte par défaut. Les données stockées dans un format crypté signifient que seul l'utilisateur qui les a stockées peut les consulter. Pour les applications qui souhaitent que d'autres utilisateurs puissent visualiser les données, l'application doit définir l'option de cryptage sur false. De même, l'option de décryptage de *getFile()* doit également être false.

```
async function getFile() {
   try{
     blockstack.getFile(path: string, options: {decrypt: true | false | string});
   }
    catch(err){
       alert(err.message);
    }
 }
```

FIGURE 4.3 – Fonction de récupération d'un fichier (Lire).

```
async function putFile() {
  try{
      blockstack.putFile(
          path: string,
          content: string | Buffer | ArrayBufferView | Blob,
          options?: {decrypt: true | false | string});
  }
   catch(err){
      alert(err.message);
   }
}
```

FIGURE 4.4 – Fonction d'écriture sur un fichier (Ecrire).

Les interfaces *getFile()* et *putFile()* sont simples car Blockstack suppose et veut encourager une communauté de bibliothèques de gestion de données open source.

• **Récupération de tous les fichiers existants**

Pour récupérer la liste de tous les fichiers existants dans le stockage Gaia hub d'un utilisateur, on utilise la fonction *listFiles()*.

```
function listFiles(){
  var listoffiles=' ';
  blockstack.listFiles((filename)=>{listoffiles=listoffiles+'<br/>'+filename;
     document.getElementById("showfiles").innerHTML=listoffiles
         return true}      )
  }
```

FIGURE 4.5 – Exemple d'affichage de tous les fichiers existants

4.3 Scenario de fonctionnement et présentation graphique

Après qu'un internaute désir rentrer sur notre application, la page d'accueil qui est présentée dans la figure suivante est le seul point d'entrer pour chaque personne.

FIGURE 4.6 – Page d'accueil de l'application.

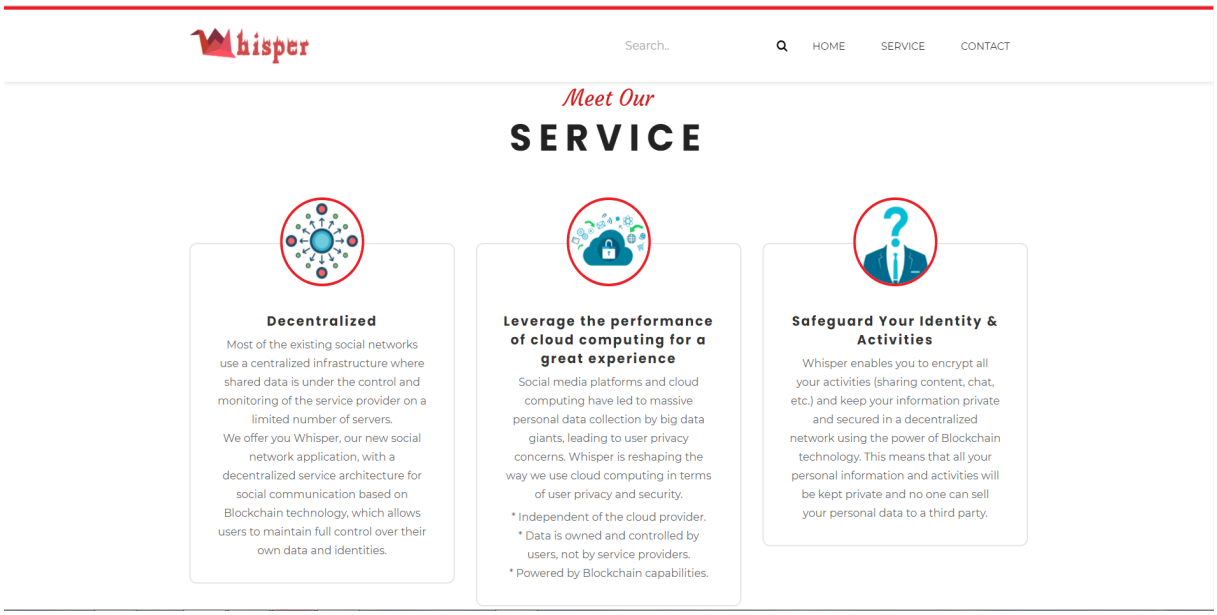

FIGURE 4.7 – Présentation des services.

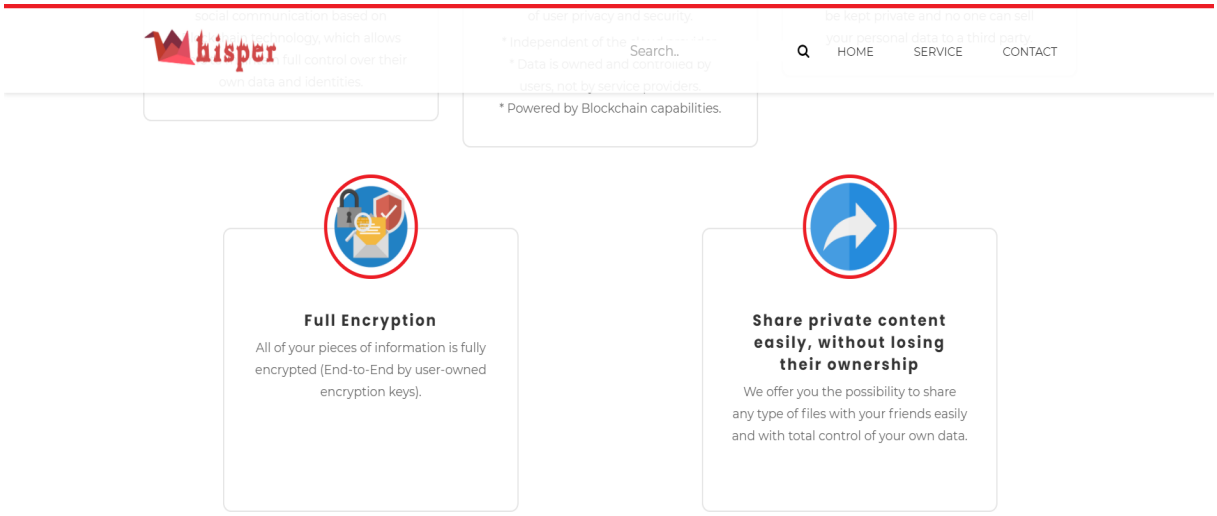

FIGURE 4.8 – Présentation des services 2.

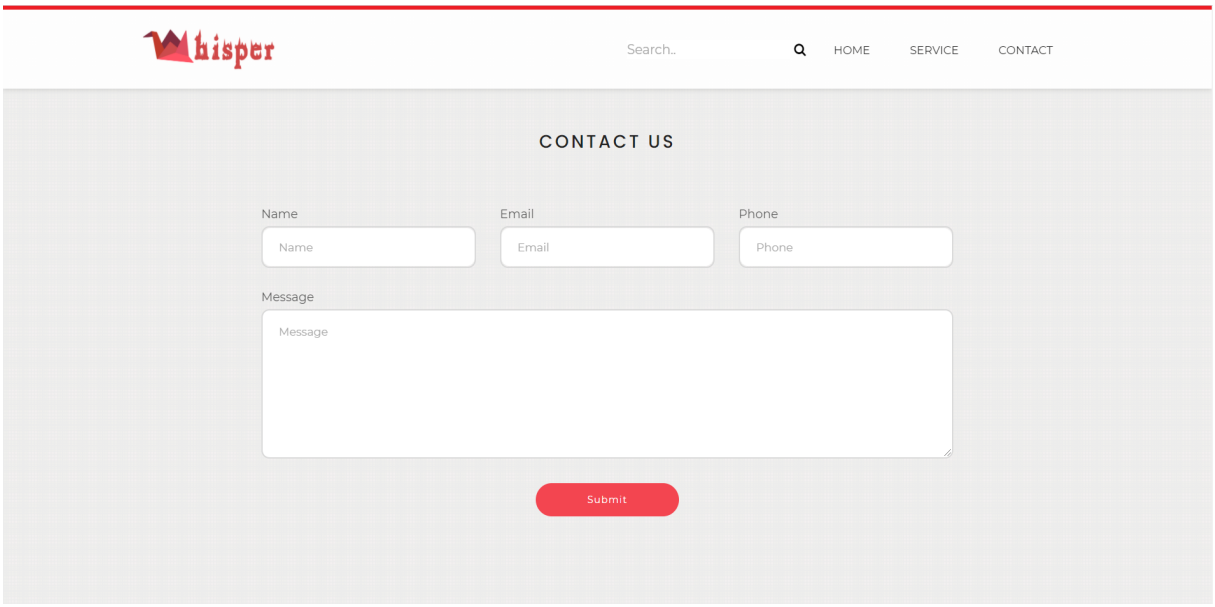

FIGURE 4.9 – Contacter les développeurs.

Si l'internaute décide de créer un compte personnel (s'inscrire) ou de se connecter, il clique sur le bouton SIGN IN présenté dans la page d'accueil (Figure Accueil). Ensuite il va être dirigé vers la page d'authentification fourni par Blockstack présenté dans la figure suivante. Ensuite, l'utilisateur choisi de créer un nouveau compte (Create new ID) ou de se connecter avec un compte existant (Sign in with an existing ID).

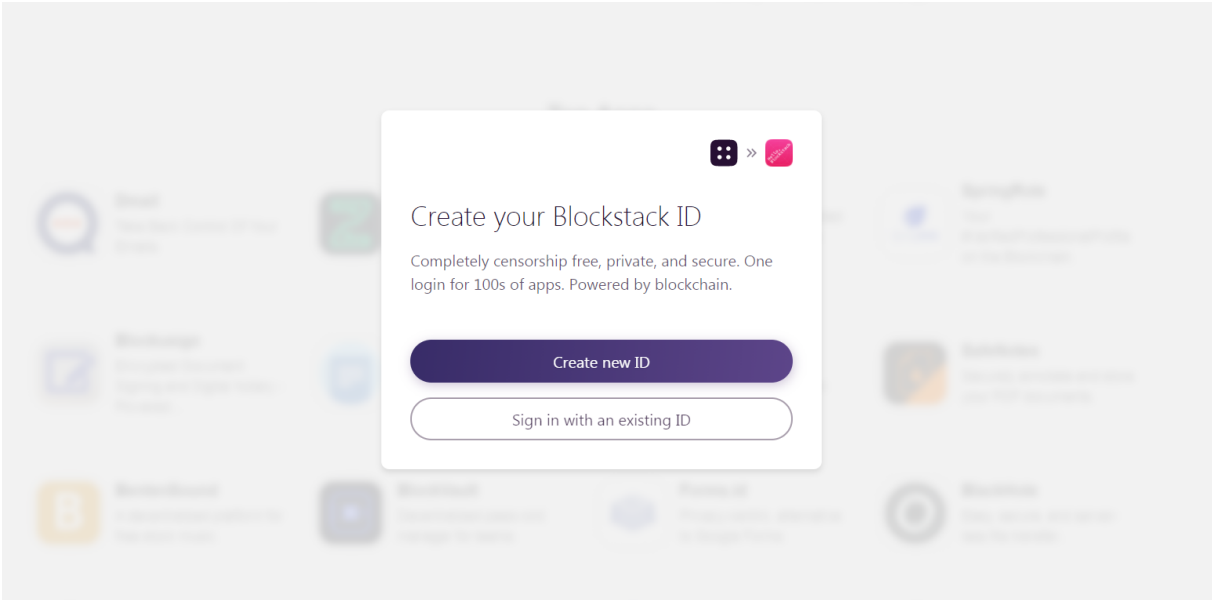

FIGURE 4.10 – Page d'authentification de Blockstack.

Si l'utilisateur choisi de créer un nouveau compte, il sera obligé de choisir un nom d'utilisateur unique qui sera sous la forme *username.id.blockstack* (Figure 4.11), un mot de passe (Figure 4.12) et de remplir son email (Figure 4.13).

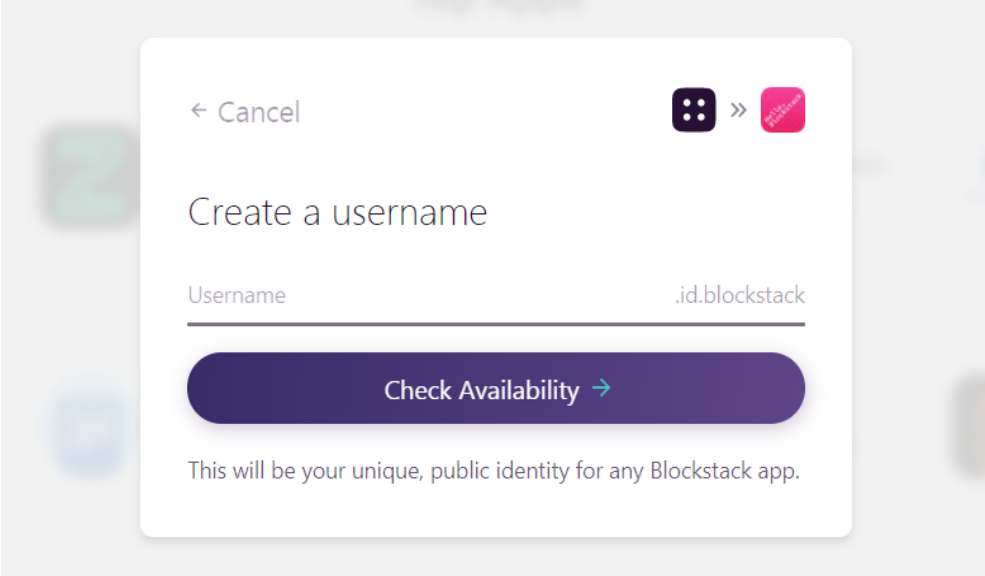

FIGURE 4.11 – Création d'un nom d'utilisateur.

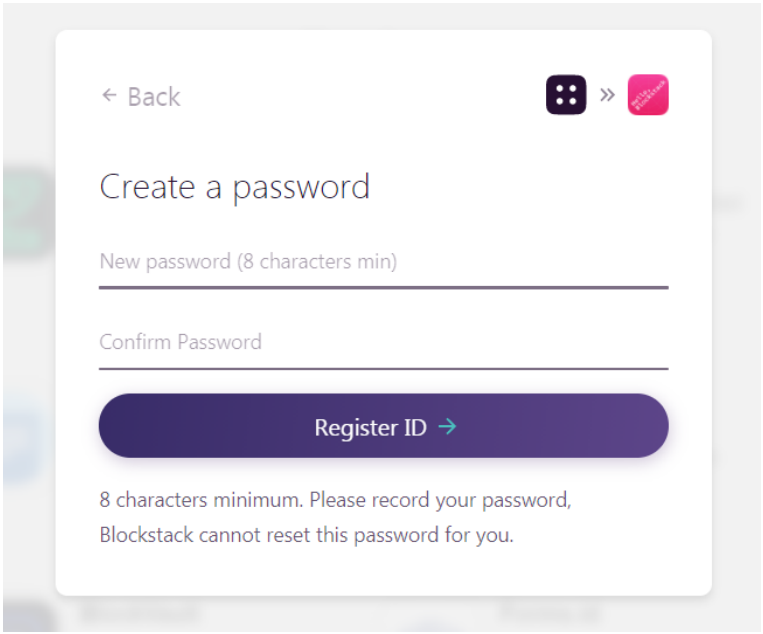

FIGURE 4.12 – Création d'un mot de passe.

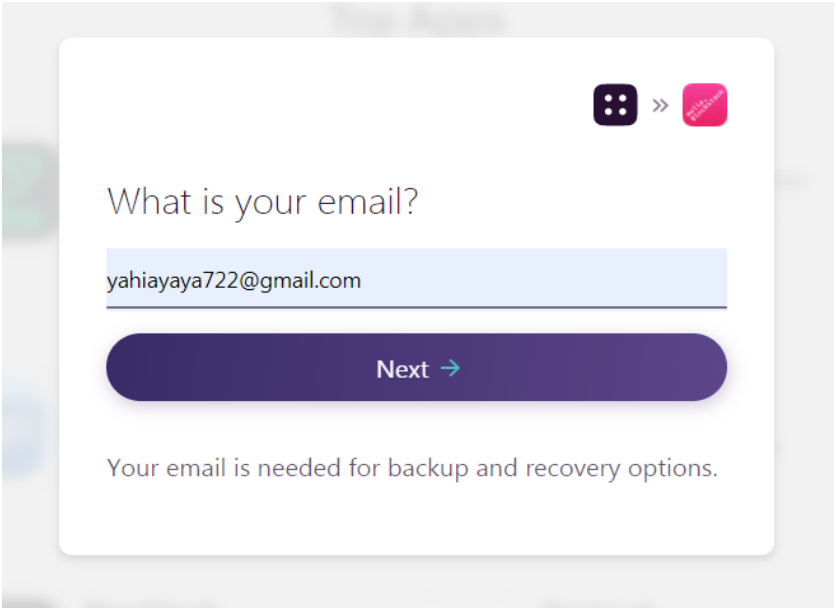

FIGURE 4.13 – Remplissage de l'email.

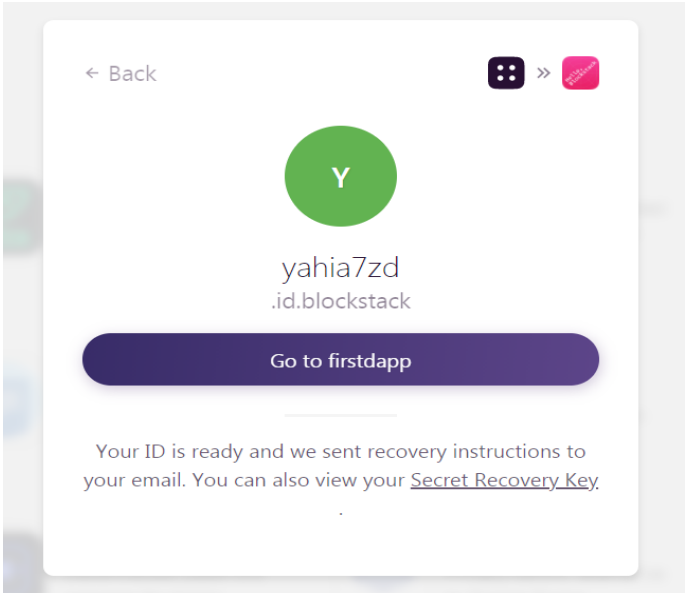

FIGURE 4.14 – Compte d'utilisateur créé.

Après la création de compte par l'utilisateur, il doit copier une clé secrète généré par Blockstack pour lui, et qui va l'utiliser pour accéder à son compte.

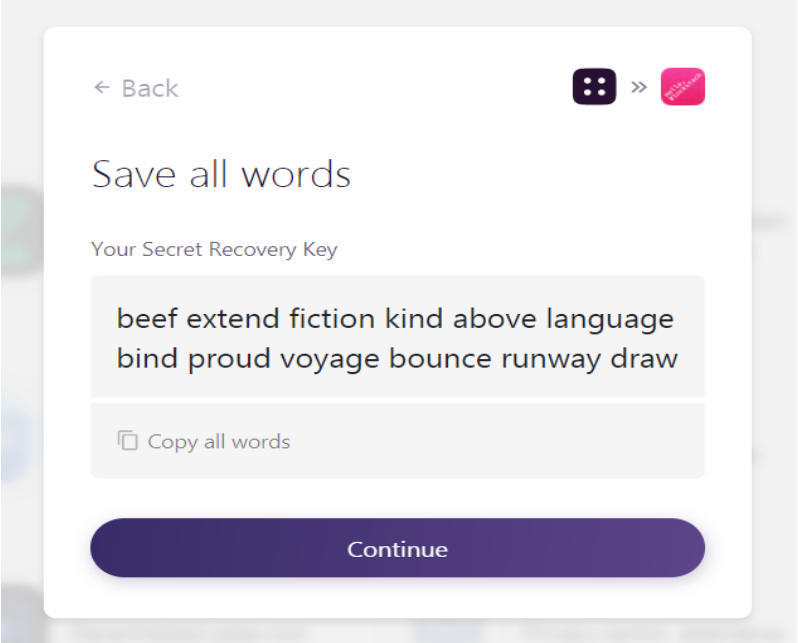

FIGURE 4.15 – Clé secrète générée.

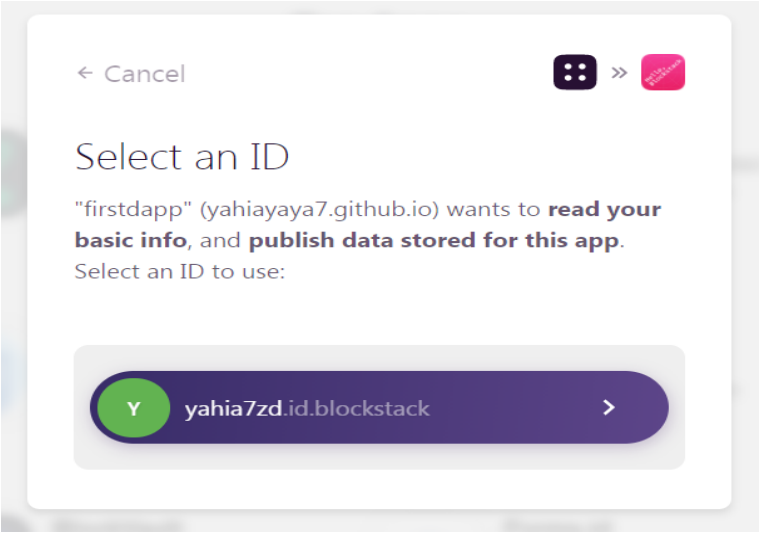

FIGURE 4.16 – Cas de connexion avec un compte existant.

Dès que l'utilisateur accède à son compte, il peut bénéficier des services de l'application (partager des publications publiques ou privées ou à un groupe, créer un groupe, parcourir un profil d'un autre utilisateur ou d'un groupe,...etc.). Il peut aussi de personnaliser son profil par le choix d'un nom d'utilisateur qui va apparaître dans son profil, remplir sa date de naissance,...etc.

La figure suivante représente un exemple d'un compte d'un utilisateur.

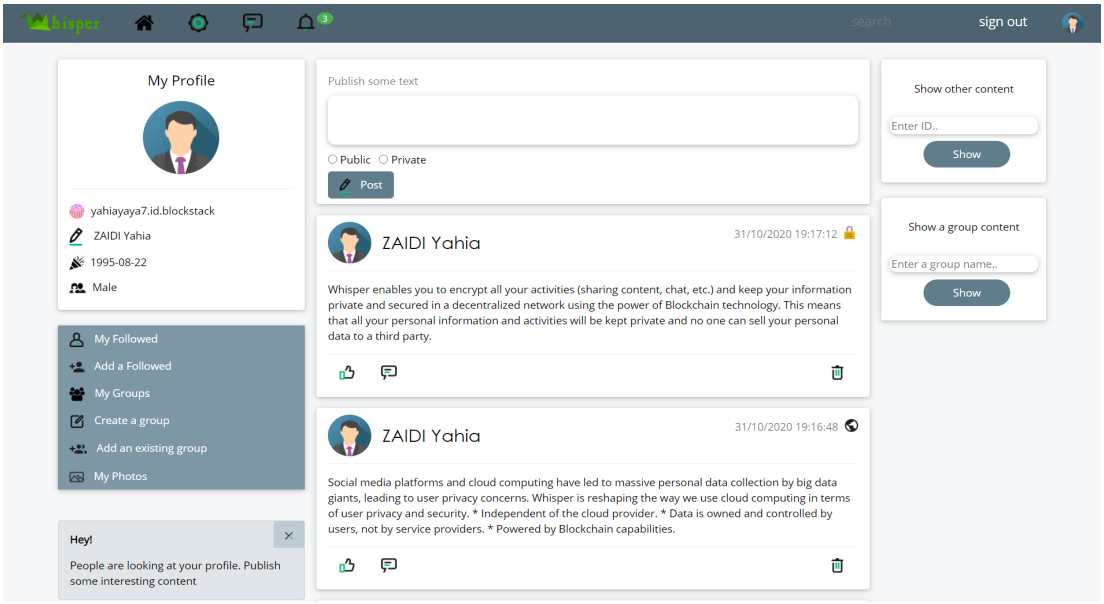

FIGURE 4.17 – Exemple d'un compte d'un utilisateur.

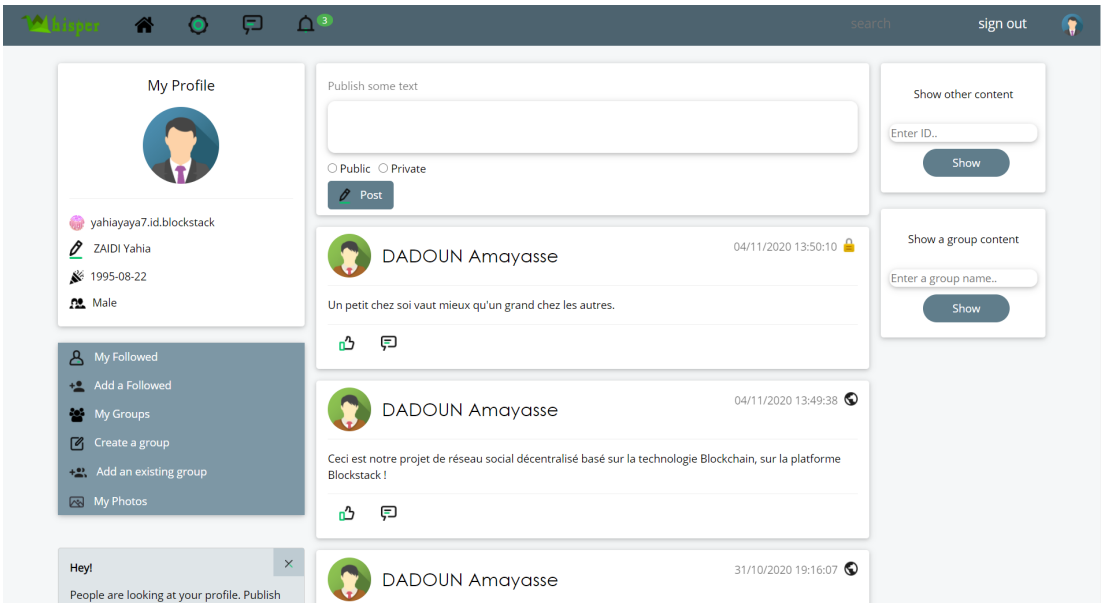

FIGURE 4.18 – Exemple d'un contenu d'un autre utilisateur.

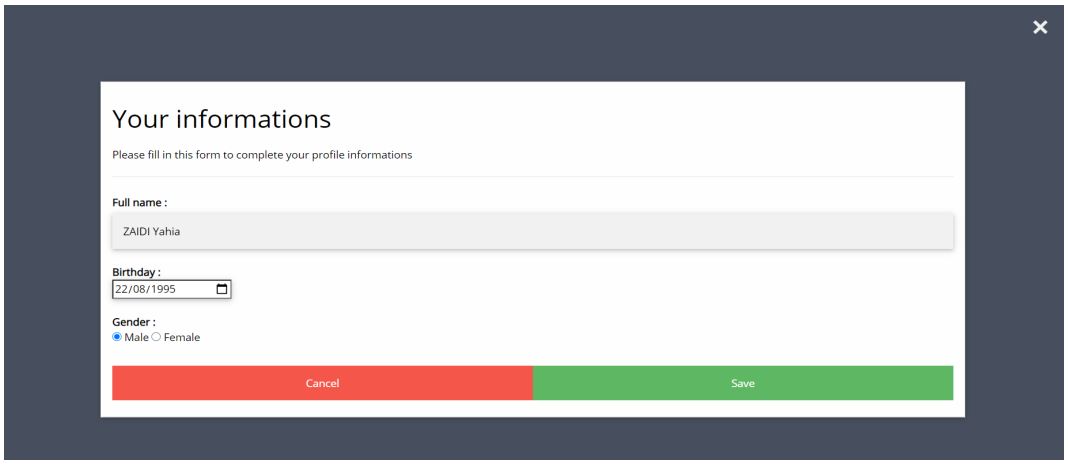

FIGURE 4.19 – Remplissage des informations personnelles par un utilisateur.

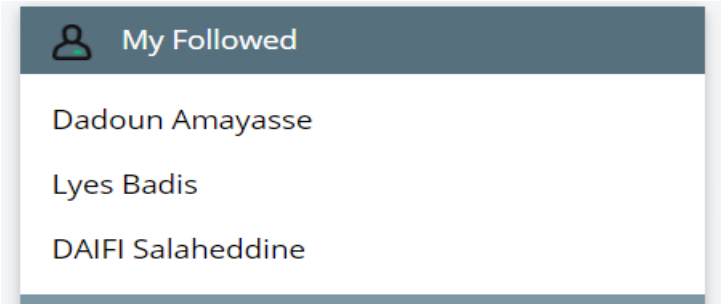

FIGURE 4.20 – Exemple d'une liste d'abonnement d'un utilisateur.

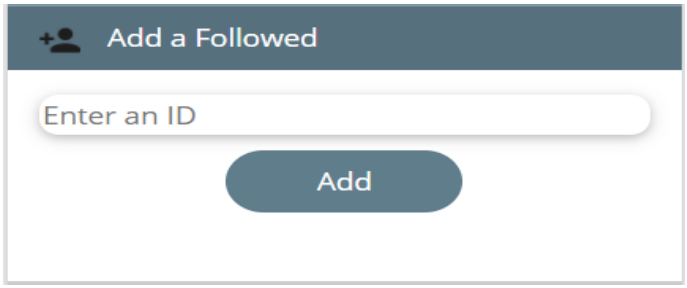

FIGURE 4.21 – Abonner un autre utilisateur.

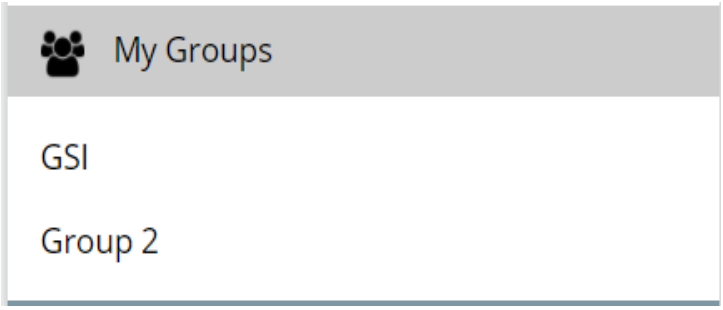

FIGURE 4.22 – Exemple d'une liste des groupes d'un utilisateur.

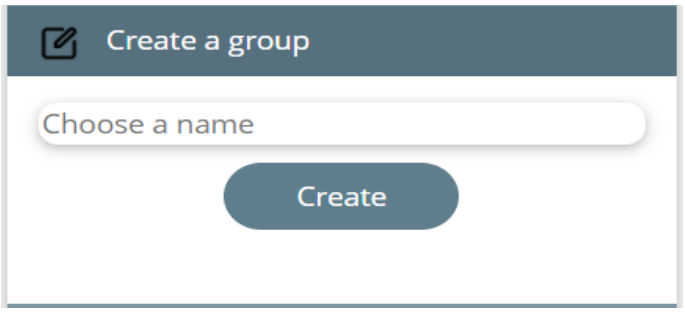

FIGURE 4.23 – Création d'un groupe par un utilisateur.

La fonction *creategroup()* qu'on a implémenté est responsable de la responsable de la création des groupes pour un utilisateur, l'utilisateur choisi un nom de group et clique sur le bouton *Create*, ensuite la fonction lui génère une clé de groupe et stocke le nom et la clé dans le fichier des groupes surnommé LocalGroupindex. L'ajout d'un membre a ce groupe de partage se fait par l'envoi du nom de groupe et de la clé de cryptage (de la meme façon pour une invitation à une réunion sur Zoom ou Google Meet par exemple).

```
async function creategroup() {
   var length = 10;
   var groupkey = '';
   var characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
   var charactersLength = characters.length;
     for ( var i = 0; i < length; i++ ) {
        groupkey += characters.charAt(Math.floor(Math.random() * charactersLength));
     }

    var groupname=document.getElementById("groupname").value;
    var groupindex= 'LocalGroupindex';
    try{
      var groupindexcont= await blockstack.getFile(groupindex, {decrypt: true });
      var str = groupname +'XX'+ groupkey + '</br>';
      if(groupindexcont == null){
        var strgroup = str;
      }
      else{
        var strgroup = groupindexcont.toString() + str;
      }
      blockstack.putFile(groupindex, strgroup, {encrypt: true});
      var showkey = 'Your group key: ' + groupkey;
      document.getElementById("groupkey").innerHTML = showkey;
   }
    catch(err){
     alert(err.message);}
}
```

FIGURE 4.24 – Fonction de création d'un groupe.

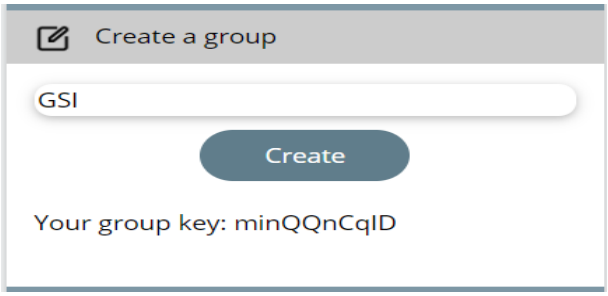

FIGURE 4.25 – Clé générée après la création d'un groupe.

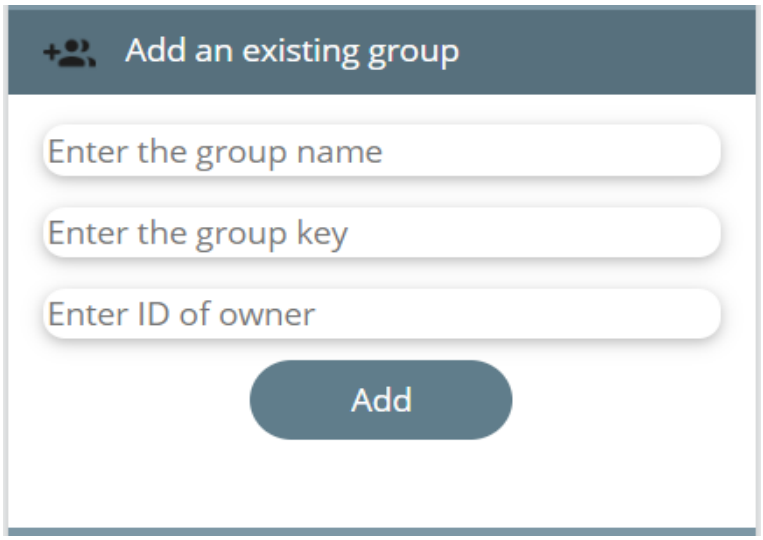

FIGURE 4.26 – Ajout d'un groupe existant.

4.4 Conclusion

L'objectif de ce chapitre a été l'implémentation de toutes les méthodes définis et étudiées dans chapitres précédents. Dans cette phase "Implémentation" nous avons montré les différentes fonctionnalités de notre système ainsi que le résultat final par des captures d'écrans de notre application finale.

Conclusion générale et perspectives

Au terme de notre travail, nous espérons avoir atteint l'objectif que nous nous sommes fixé au début, à savoir la réalisation d'une application de réseau social décentralisé basé sur la technologie Blockchain permettant principalement au utilisateurs de partager tout type de fichier tout en gardant le contrôle total de leurs données et identités. Nous espérons également que cette application sera bénéfique et rentable dans le domaine des réseaux sociaux et ce dans la mesure où elle facilitera les taches de ses utilisateurs en ce qui concerne la bonne expérience d'un RS fiable qui prend en compte la sécurité de la vie privée de ses utilisateurs.

Nous avons présenté dans ce mémoire les fonctionnalités de l'application (partage d'un contenu, abonnement, découverte,... etc.) modélisées avec le langage UML suivant la plateforme Blockstack que nous avons choisi pour le développement de notre application profitant de ses fonctionnalités offertes en terme de sécurisation des interactions et des identités des utilisateurs, et un contrôle total de leurs propres données.

Nous avons présenté aussi la technologie Blockchain, qui est une technique révolutionnaire sécurisée, transparente et qui fonctionne sans entité centrale de contrôle, pour sécuriser l'accès aux données dans l'espace de stockage et les processus d'authentification et d'inscription, et assurer une distribution des noms décentralisée et une parfaite gestion des identités.

Ce travail nous a amené à confirmer qu'avec de la détermination il est possible d'explorer des domaines totalement nouveaux. Un travail qui nous a permis également d'apprendre et d'acquérir des connaissances dans le domaine des réseaux sociaux et des réseaux sociaux décentralisés en particulier qui utilisent la technologie Blockchain, et de conforter nos connaissances en conception logicielle, et qui sera amélioré par la suite.

Nous envisageons comme perspective du travail réalisé dans ce mémoire :

— Ajout de possibilité de partager d'autres types de fichiers (image, audio, vidéo,... etc.).

— Ajout des fonctionnalités de liker (aimer) et commenter une publication.

— Ajout de la fonctionnalité de Tagging.

— Ajout de la possibilité d'échanger des crypto-monnaies entre les utilisateurs.

— Récompenser les utilisateurs avec la crypto-monnaie pour leur contenu de valeur.

— Développer une bonne recommandation des profils et des contenus.

Bibliographie

a. Bibliographie :

[2] C. DWYER , "Privacy in the Age of Google and Facebook," Technology and Society Magazine, IEEE, vol 30, no 3, pages 58-63, 2011.

[3] G. Pallis, D. Zeinalipour-Yazti, and M. D. Dikaiakos,"Online social networks : Status and trends", New Directions in Web Data Management 1, pages 213–234, 2011

[4] D. M. Boyd and N. B. Ellison, "Social network sites : Definition, history, and scholarship", Journal of Computer-Mediated Communication, vol 13, no 1, 2007.

[6] Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of topological characteristics of huge online social networking services", The 16th International Conference on World Wide Web, pages 835–844, 2007.

[8] F. Michel, "Définir et analyser les réseaux sociaux", Informations sociales, vol 3, no 147, page 138, 2008.

[9]. p. Torloting , "Enjeux et perspectives des réseaux sociaux", Institut Supérieur du Commerce, Paris, mémoire de fin d'étude, 2006.

[10]. S. Buchegger, D. Schiöberg, L.-H. Vu, and A. Datta, "Peerson : P2p social networking : Early experiences and insights", The 2nd ACM EuroSys Workshop on Social Network Systems, pages 46–52, 2009.

[11]. A. Shakimov, A. Varshavsky, L. P. Cox, and R. Cáceres, "Privacy, cost, and availability tradeoffs in decentralized osns", The 2nd ACM Workshop on Online Social Networks, pages 13–18, 2009.

[12]. L. Badis , Dj. Aissani and M. Amad , "A Log Based Update of Replicated Profiles in Decentralized Social Networks", Journal of Digital Information Management, vol 16, Octobre 2018.

[13]. S. Buchegger and A. Datta, "A case for p2p infrastructure for social networks - opportunities & challenges", The 6th International Conference on Wireless On-Demand Network Systems and Services, IEEE, Mars 2009.

[14]. S. Dahmani and S. Hammadi , "Sécurisation des communications dans les réseaux sociaux décentralisée", Mémoire de fin d'étude, Université de Bouira, 2018.

[15]. M. R. Bouadjenek, H. Hacid, and M. Bouzeghoub, "Social networks and information retrieval, how are theyconverging ? A survey, a taxonomy and an analysis of socialinformation retrieval approaches and platforms", Information Systems, Elsevier, vol 56, pages 1-18, Mars 2016.

[16]. G. O. Oparaocha, "Towards building internal social network architecture that drives innovation : a social exchange theory perspective", Journal of Knowledge Management, 2016.

[17]. N. ZAMMAR , "Reseaux Sociaux numeriques : essai de categorisation et cartographie des controverses", Université Rennes 2, Thèse de doctorat, 2012.

[18]. CANALBLOG, http ://reseauxlapie.canalblog.com, consulté le 10/09/2020.

[19]. P. W. L. Fong, M. Anwar, and Z. Zhao, "A Privacy Preservation Model for Facebook-Style Social Network Systems", The 14th European Conference on Research in Computer Security, ESORICS, pages 303–320, 2009.

[20]. B. Carminati, E. Ferrari, and A. Perego, "Enforcing access control in web-based social networks", ACM Transactions on Information and System Security (TISSEC), vol 13, no 1, pages 1–6, 2009.

[21]. A. De Salve, P. Mori, and L. Ricci, "A survey on privacy in decentralized online social networks", Computer Science Review, Vol 27, Pages 154-176, Février 2018.

[22]. B.Greschbach, G. Kre itz, and S. Buchegger, "The devil is in the metadata - new privacychallenges in decentralised online social networks", IEEE International Conference on Pervasive Computing and Communications Workshops, IEEE, pages 333-339, Mars 2012.

[23]. S. Rathore, P.K. Sharma, V. Loia, Y-S. Jeong and J.H. Park, "Social network security : Issues, challenges, threats, and solutions", Information Sciences, vol 421, pages 43-69, Décembre 2017.

[24]. T. Paul, A. Famulari, and T. Strufe, "A survey on decentralized online social networks. Computer Networks", vol 75, pages 437-452, 2014.

[25]. S. R. Chowdhury, A. R. Roy, M. Shaikh, and K. Daudjee, "A taxonomy of decentralized online social networks", Peer-to-Peer Networking and Applications, pages 1–17, 2014.

[26]. D. Konforty, Y. Adam, D. Estrada and L. G. Meredith, "Synereo : The Decentralized and Distributed Social Network", 15 Mars 2015.

[27]. S. W. Seong, J. Seo, M. Nasielski, D. Sengupta, S. Hangal, S. K. Teh, R. Chu, B. Dodson, and M. S. Lam, "Prpl : a decentralized social networking infrastructure", Computer Science and Electrical Engineering Departments, Stanford University, Janvier 2010.

[28]. L. A. Cutillo, R. Molva, and T. Strufe, "Safebook : A privacy-preserving online social network leveraging on real-life trust", IEEE Communications Magazine, vol 47, no 12, pages 94–101, Janvier 2009.

[29]. S. Nilizadeh, S. Jahid, P. Mittal, N. Borisov, and A. Kapadia, "Cachet : a decentralized architecture for privacy preserving social networking with caching", The 8th international conference on Emerging networking experiments and technologies, Décembre 2012.

[30]. A. Shakimov, H. Lim, R. Caceres, L. P. Cox, K. Li, D. Liu, and A. Varshavsky, "Vis-a-vis : Privacy-preserving online social networking via virtual individual servers", 2011 Third International Conference in Communication Systems and Networks (COMSNETS), IEEE, pages 1–10, 2011.

[31]. R. Sharma and A. Datta, "Supernova : Super-peers based architecture for decentralized online social networks", 2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012), pages 1–10, IEEE, Janvier 2012.

[32]. D. Sandler and D. S. Wallach, "Birds of a fethr : open, decentralized micropublishing", IPTPS, Janvier 2009.

[33]. L. Schwittmann, C. Boelmann, M. Wander and T. Weis, "Sonet-privacy and replication in federated online social networks", ICDCSW, Springer, pages 51–57, 2013.

[34]. S. Schulz, and T. Strufe, "d2 deleting diaspora : practical attacks for profile discovery and deletion", ICC, pages 2042– 2046, 2013.

[35]. M. Durr, M. Maier and F. Dorfmeister, "Vegas – a secure and privacypreserving peer-to-peer online social network", Privacy, Security, Risk and Trust (PASSAT), pages 868–874, 2012.

[36]. D. Liu, A. Shakimov, R. Càceres, A. Varshavsky, and L.P. Cox, "Confidant : protecting osn data without locking it up", Middleware, 2011.

[37]. Y. MOUALKIA, "Performance et Optimisation des Architectures P2P pour les Applications des Réseaux Sociaux", Mémoire de Magister, Université de Bejaia, 2016.

[38]. A. Datta, S. Buchegger, L. Hung Vu, T. Strufe, and K. Rzadca, "Decentralized Online Social Networks", Handbook of Social Network Technologies and Applications, pages 349-378, Octobre 2010.

[39]. H. T. M. Gamage, H. D. Weerasinghe and N. G. J. Dias, "A Survey on Blockchain Technology Concepts, Applications, and Issues", SN Computer Science, Avril 2020.

[40]. M. D Pierro, "What is the Blockchain ?", Computing in Science & Engineering, 2017.

[41]. B. M. Hoy, "An introduction to the blockchain and its implications for libraries and medicine", Medical reference services quarterly, vol 36, no 3, pages 273-279, 2017

[42]. S. Nakamoto, "Bitcoin : A peer-to-peer electronic cash system", https ://bitcoin.org/bitcoin.pdf, 2008.

[43]. J. L. de la Rosa, V. Torres-Padrosa, A. el Fakdi and D. Gibovic, "A survey of blockchain technologies for open innovation", World Open Innovation Conference, San Francisco, 2017.

[44]. I. Eyal, "Blockchain technology : Transforming libertarian cryptocurrency dreams to finance and banking realities", Computer 50(9), pages 38-49, Janvier 2017.

[45].M. E. Peck, "Blockchain world - Do you need a blockchain ? This chart will tell you if the technology can solve your problem", IEEE Spectrum 54(10), pages 38-60, Octobre 2017.

[46]. Y. Yuan and F.Y. Wang, "Blockchain and cryptocurrencies : Model, techniques, and applications", IEEE Transactions on Systems, Man, and Cybernetics : Systems 48(9), pages 1421-1428, 2018.

[47]. K. Kulkarni, "Learn Bitcoin and Blockchain : Understanding blockchain and Bitcoin architecture to build decentralized applications", Packt Publishing, Août 2018.

[48]. O. A. Ayadi, "CHAPITRE III : État de l'art de la Blockchain", Université de Constantine 2, Juillet 2019.

[49]. S. Tessier, "Fonctionnement de la blockchain et son intérêt pour le monde pharmaceutique", Thèse de doctorat, Université de Bordeau, 2019.

[50]. T. U. Tulsidas, "Smart contracts from a legal perspective", Projet de fin d'étude, Université d'Alicante, 2018.

[51]. D. Gerard, "Attack of the 50 Foot Blockchain : Bitcoin, Blockchain, Ethereum & Smart Contracts", CreateSpace Independent Publishing Platform, United States, Juillet 2017.

[52]. S. Jani, "An Overview of Ethereum & Its Comparison with Bitcoin", International Journal of Scientific & Engineering Research, Vol 10, no 8, Décembre 2017.

[53]. M. Javor, B. Skvorc, T. Jankov, and A. Bouchefra, "Ethereum : Tools & Skills", SitePoint Pty. Ltd., Août 2018.

[54]. M. Westerkamp, S. Göndör, and A. Küpper, "Tawki : Towards Self-Sovereign Social Communication", Janvier 2019.

[55]. M. U. Rahman, F. Baiardi, B. Guidi et L. Ricci, "Protecting Personal Data using Smart Contracts", Octobre 2019.

[56]. M. U. Rahman, B. Guidi, F. Baiardi, "Blockchain-based access control management for Decentralized Online Social Networks", Journal of Parallel and Distributed Computing, Juillet 2020.

[57]. B. Guidi, "When Blockchain meets Online Social Networks", Pervasive and Mobile Computing, Février 2020.

[59]. Steem whitepaper, "Steem : An incentivized, blockchain-based, public content platform", url https ://steem.com/steem-whitepaper.pdf, 2018.

[60]. L.Chao , B. Palanisamy, "Incentivized Blockchain-based Social Media Platforms : A Case Study of Steemit", The 10th ACM Conference, Juin 2019.

[61]. M. Ali, J. Nelson, R. Shea et M. J. Freedman, "Blockstack : A global naming and storage system secured by blockchains", 2016 USENIX annual technical conference (USENIX ATC 16), pages 181-194, 2016.

[62]. Z. Corbyn, "Decentralisation : the next big step for the world wide web", The Guardian, vol 8, 2018.

[63]. A. Mühle, A. Grüner, T. Gayvoronskaya, C. Meinel, "A Survey on Essential Components of a Self-Sovereign Identity". Computer Science Review, vol 30, pages 80–86,November 2018.

[65].A. Muneeb , "Trust-to-Trust Design of a New Internet", une dissertation présenté à la faculté de l'université Princeton, page 30, Juin 2017.

[66]. H. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau, and A. Narayanan. An empirical study of Namecoin and lessons for decentralized namespace design. WEIS '15 : Proceedings of the 14th Workshop on the Economics of Information Security, juin 2015.

[67]. D. Kaminsky, "Spelunking the triangle : Exploring aaron swartz's take on zooko's triangle", 2011.

[68]. O. Glassey and J. L. Chappelet. Comparaison de trois techniques de modélisation de processus : ADONIS, OSSAD et UML. IDHEAP, Institut de hautes études en administration publique, 2002.

[71]. B. FRAIN, Responsive Web Design avec HTML 5 et CSS 3, 1re édition-Pearson, fevrier 2013.

[72]. C. VIGOUROUX, Apprendre à développer avec JavaScript, Editions-ENI, Avril 2014.

b. Webographie :

[1]. https ://blog.keepersecurity.com/2012/06/07/linkedinpasswords-leaked-6-5-million-accounts-compromised/, KEEPER SECURITY, consulté 22/06/2020.

[5]. https ://www.arturin.com/infographie-evolution-reseaux-sociaux-1997-2019/, consulté le 22/06/2020.

[7]. https ://fr.statista.com/statistiques/570930/reseaux-sociaux-mondiaux-classes-par-nombre-d-utilisateurs/, consulté le 17/07/2020.

[58]. https ://steemblockexplorer.com/, consulté le 15/10/2020.

[64]. https ://docs.blockstack.org/ , consulté le 17/07/2020

[69]. https ://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1443812-github-definition-api-desktop/

[70]. http ://www.techterms.com/definition/html

[73]. https ://www.json.org/json-en.html