REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de fin d'études en vue de l'obtention d'un master académique

Filière : Informatique

Spécialité : Réseaux, Mobilités et systèmes embarqués

THEME :

Application décentralisée pour la sécurisation De transactions monétaires

Présenté par :

M. BENSAID Massyl

M. HAMMOUTENE Rabah

Encadré Par :

M. DAOUI Mehammed

Promotion : 2019-2020

Table des matières

Liste des Figures :

Remerciements

Nous rendons grâce à Dieu qui nous a donné l'aide, la patience et le courage pour accomplir ce travail et nous a maintenu en santé pour mener à bien cette année d'étude.

Nous tenons à remercier notre promoteur, DAOUI Mehammed, pour l'aide qu'il a fourni et les connaissances qu'il a su nous transmettre. Nous le remercions également pour sa disponibilité et la qualité de ses conseils.

Nos remerciements s'adressent aussi aux Mrs et Mmes les jurés pour l'intérêt qu'ils ont porté à ce travail en acceptant d'être examinateurs.

Nous tenons à saisir cette occasion et adresser nos profonds remerciements et nos profondes reconnaissances aux responsables et au personnel de l'UMMTO ainsi qu'aux professeurs de l'UMMTO, qui nous ont fourni les outils nécessaires à la réussite de nos études universitaires.

Un grand merci à nos parents pour leur amour, leurs conseils ainsi que leur soutien inconditionnel à la fois moral et économique, ainsi que nos sœurs et frères et toutes nos familles, qui nous ont permis de réaliser les études que nous voulions et par conséquent ce mémoire.

Nous voudrions exprimer nos reconnaissances envers les amis et collègues qui nous ont apporté leurs conseils et soutien moral et intellectuel tout au long de notre démarche.

Nous remercions toutes les personnes qui nous ont soutenues de près ou de loin dans la réalisation de ce travail.

Introduction générale

« La blockchain », un terme qui ne cesse de revenir dans la liste des recherches les plus populaires sur les différents moteurs de recherche.
Depuis que le mystérieux Satoshi Nakamoto a lancé le Bitcoin en 2009, une cryptomonnaie basée cette nouvelle technologie qu'est la blockchain.
Réputée ultra-sécurisée et inviolable, les internautes, les financiers, informaticiens ou encore Traders cherchent tous à en savoir encore plus sur cette technologie qui d'après certains experts pourrait devenir même l'internet de demain, tellement elle a bouleversée notre compréhension concernant le rapport aux informations et aux transferts de valeurs.

Une transaction monétaire électronique est une opération informatisée d'envoi et de réception d'argent entre deux comptes bancaires ou deux entités et c'est une opération qui exige une sécurité et confiance absolues. La transaction est gérée, validée et sécurisée par les serveurs centralisés des systèmes bancaires, ce qui peut donc engendrer un bug d'application et c'est ce qui rend la transaction altérable et hackable.
Grace à l'évolution des technologies et des services informatiques, tel que la blockchain, nous pouvons transmettre et de stocker des informations en ligne de manière transparente, fiable et sécurisée, et fonctionnant sans organe central de contrôle.

L'objectif de ce projet reposant sur les principes de la blockchain et de smart contract est de mettre en place une application pratique pour la gestion des transactions bancaires. Ce qui peut se faire en construisant une application décentralisée qui va regrouper tous les nœuds d'un réseau bancaire totalement distribué avec leurs détails ainsi que la possibilité d'échanger de l'argent entre eux. Cela va garantir la confiance de chaque nœud.
C'est dans ce cadre que s'inscrit notre projet intitulé « Application décentralisée pour sécurisation de transaction monétaire ». Notre application permettra de réaliser des transferts monétaires sans tiers de confiance grâce à la technologie blockchain.

Pour ce faire, nous avons structuré notre mémoire en quatre chapitres :

- Chapitre 1 : « L'architecture distribuée et le Cloud Computing », a pour objectif la présentation de quelques concepts informatiques de base sur lesquelles se reposent la blockchain et notre projet.

- Chapitre 2 : « Sécurité informatique, cryptographie et Blockchain », dans ce chapitre nous allons parcourir en détails les différents composants de la technologie blockchain ainsi que de la sécurité informatique et la cryptographie.

- Chapitre 3 : « Conception et études du projet », ce chapitre consistera à développer notre solution en présentant les différents objectifs et l'explication de l'architecture à l'aide des schémas.

- Chapitre 4 : « Réalisation et déploiement de la solution », consacré à la réalisation de notre système, ce chapitre va présenter les différents outils, services et technologies qui vont nous servir à réaliser notre projet ainsi que le résultat final de notre système une fois mis en place.

Chapitre 1
L’architecture distribuée
Et le Cloud Computing

Chapitre 1 : L’architecture distribuée et le Cloud Computing

1. Introduction

Dans ce chapitre, nous allons présenter les concepts informatiques de base sur lesquelles la technologie blockchain et notre projet s’appuient. Nous aborderons principalement l’architecture distribuée sur laquelle la blockchain se base ainsi que le cloud computing qui va nous servir à implémenter notre propre blockchain et l’application mobile qui gère les transferts monétaire entre les utilisateurs.

2. L’architecture distribuée [1]

L’architecture distribuée ou l’architecture répartie désigne un système d'information ou un réseau pour lequel l'ensemble des ressources disponibles ne se trouvent pas au même endroit ou sur la même machine. Ce concept, dont une version peut être une combinaison de transmissions du type client-serveur, s'oppose à celui d'informatique centralisée.

Internet est un exemple de réseau distribué puisqu'il ne possède aucun nœud central. Les architectures distribuées reposent sur la possibilité d'utiliser des objets qui s'exécutent sur des machines réparties sur le réseau et communiquent par messages au travers du réseau.

3. Le réseau pair-à-pair [2] [5]

3.1. Définition et Fonctionnement

Le réseau pair-à-pair ou système pair à pair (en anglais peer-to-peer, abrégé en « P2P ») est un modèle d'échange en réseau où chaque entité est à la fois client et serveur, contrairement au modèle client-serveur. Les termes « pair », « nœud » et « utilisateur » sont généralement utilisés pour désigner les entités composant un tel système. Un système pair à pair peut être partiellement centralisé (une partie de l'échange passe par un serveur central intermédiaire) ou totalement décentralisé (les connexions se font entre participants sans infrastructure particulière). Il peut servir entre autres au partage de fichier, au calcul distribué ou à la communication.

3.2. Principaux cas d'utilisation

3.2.1. Partage de fichiers

L'application la plus répandue du pair-à-pair est le partage de fichiers. L'avènement des connexions à Internet à haut débit (ADSL notamment) sans limites de temps a contribué à cet essor. Le principe réparti de ces systèmes permet de tirer parti de l'asymétrie des connexions, et donc de télécharger à débit important un fichier à partir de plusieurs sources à débit limité.

Chaque internaute est un pair du réseau et les ressources sont des fichiers. Chacun peut donc partager ses fichiers et télécharger les fichiers des autres. Ces systèmes s'avèrent très efficaces, y compris quand il s'agit d'échanger de gros volumes de données.

Parmi les applications les plus utilisées, on peut distinguer BitTorrent, µTorrent ...

3.2.2. Le calcul distribué

Une seconde application destinée au grand public ou à la recherche, mais toutefois moins répandue que le partage de fichier, est la possibilité pour les internautes de mettre à disposition une partie de leur puissance de calcul.

Les ordinateurs d'aujourd'hui sont tellement puissants que la majeure partie du temps, une grande partie de leur processeur est disponible pour effectuer des calculs. Le projet BOINC a saisi cette opportunité pour créer un parc informatique réparti dans le monde afin d'utiliser cette puissance de calcul totale pour effectuer des calculs trop complexes pour être réalisés dans un laboratoire.

3.3. Sécurité des réseaux P2P

La plupart des questions de sécurité dans les réseaux P2P sont dues au partage de fichier.

Les utilisateurs recherchent :

- L'anonymat (afin de protéger sa vie privée ou d'éviter d'éventuelles poursuites judiciaires).
- Le brouillage du protocole (afin d'éviter les filtrages du fournisseur d'accès internet).
- Le chiffrement (« on peut savoir qui je suis, mais pas ce que je télécharge »).

4. Le cloud computing [3]

4.1. Définition

Le cloud computing est un modèle permettant un accès réseau omniprésent, pratique et à la demande à un pool partagé de ressources informatiques configurables (par exemple, réseaux, serveurs, stockage, applications et services) pouvant être rapidement provisionnées et libérées avec un minimum d'effort de gestion. Ou interaction avec le fournisseur de services.

4.2. L'infrastructure cloud

Une infrastructure cloud est la collection de matériel et de logiciel qui permet les cinq caractéristiques essentielles du cloud computing. L'infrastructure cloud peut être considérée comme contenant à la fois une couche physique et une couche d'abstraction.
La couche physique comprend les ressources matérielles nécessaires pour prendre en charge les services de cloud fournis, et comprend généralement des composants de serveur, de stockage et de réseau.
La couche d'abstraction est constituée du logiciel déployé à travers la couche physique, qui manifeste les caractéristiques essentielles du cloud. Conceptuellement, la couche d'abstraction se trouve au-dessus de la couche physique.

Le cloud computing est composé de cinq (5) caractéristiques essentielles, trois (3) modèles de service ainsi que quatre (4) modèles de déploiement.

4.3. Caractéristiques essentielles [4]

4.3.1. Libre-service à la demande

Permet au consommateur de fournir de manière unilatérale des capacités de calcul, telles que le temps d'asservissement et le stockage réseau, selon les besoins, sans nécessiter d'interaction humaine avec chaque fournisseur de services

4.3.2. Large accès au réseau

Les capacités sont disponibles sur le réseau et sont accessibles les mécanismes qui favorisent l'utilisation par des plates-formes client hétérogènes minces ou épaisses (p. téléphones mobiles, tablettes, ordinateurs portables et stations de travail).

4.3.3. Mise en commun des ressources

Les ressources informatiques du fournisseur sont mises en commun pour servir plusieurs consommateurs en utilisant un modèle multi-locataire, avec différentes ressources physiques et virtuelles dynamiquement affecté et réaffecté en fonction de la demande des consommateurs. L'indépendance du stockage fait que le client n'a généralement aucun contrôle ou connaissance sur l'emplacement exact des ressources fournies, mais peut être en mesure de spécifier l'emplacement à un niveau supérieur d'abstraction (par exemple, pays, état ou centre de données). Traitement, mémoire et bande passante réseau.

4.3.4. Élasticité rapide

Les capacités peuvent être provisionnées et libérées de manière élastique, dans certains cas automatiquement. Les capacités disponibles pour l'approvisionnement semblent souvent illimitées et peuvent être approprié dans n'importe quelle quantité à tout moment.

4.3.5. Service mesuré

Les systèmes Cloud contrôlent et optimisent automatiquement l'utilisation des ressources en exploitant une capacité de mesure à un certain niveau d'abstraction approprié au type de service (p. stockage, traitement, bande passante et comptes d'utilisateurs actifs). L'utilisation des ressources peut être surveillé, contrôlé et signalé, fournissant la transparence pour le fournisseur et consommateur du service utilisé.

4.4. Modèles de déploiements du cloud

4.4.1. Le Cloud publique

Une infrastructure de cloud provisionnée pour une utilisation ouverte par le grand public. Elle peut être détenue, gérée et exploitée par un organisme commercial, universitaire ou gouvernemental, ou une combinaison d'entre eux. Elle est stockée dans les locaux du fournisseur de Cloud.

4.4.2. Le Cloud privé

Une infrastructure de cloud fournie pour une utilisation exclusive par une seule organisation comprenant plusieurs consommateurs (par exemple, des unités commerciales). Elle peut être détenue, gérée et exploitée par l'organisation, une tierce partie ou une combinaison d'entre eux, et elle peut être stockée dans ou hors des locaux.

4.4.3. Le Cloud hybride

Une infrastructure cloud composée de deux ou plusieurs infrastructures cloud distinctes (privées, publiques …) qui restent des entités uniques, mais sont liées par une technologie standardisée ou propriétaire qui permet la portabilité des données et des applications

4.4.4. Le Cloud communautaire

Une infrastructure cloud mise en service pour une utilisation exclusive par une communauté spécifique de consommateurs provenant d'organisations ayant des préoccupations communes (par exemple, la mission, les exigences de sécurité, les règles et les considérations de conformité). Il peut être détenu, géré et exploité par une ou plusieurs organisations de la communauté, une tierce partie ou une combinaison d'entre elles, et il peut exister dans ou hors des locaux.

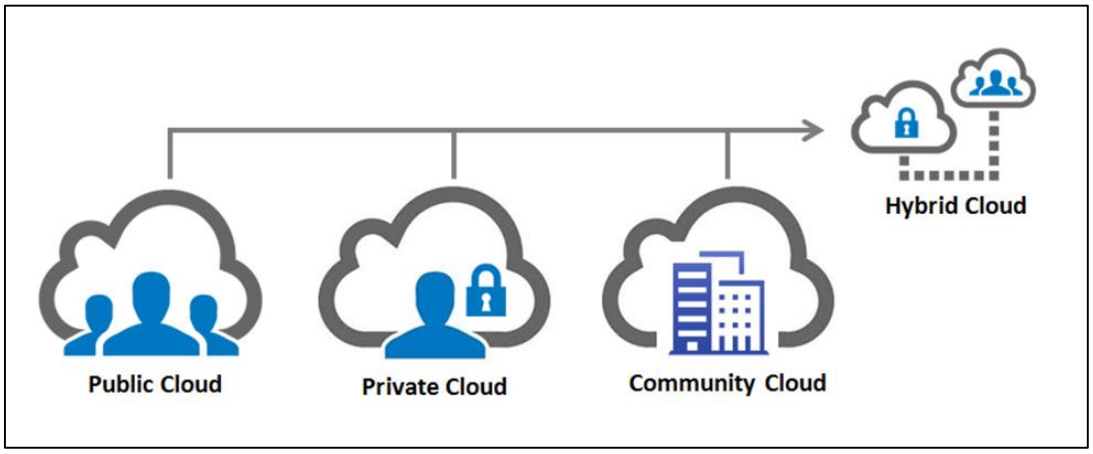


Figure 1.1 : Représentation des différents modèles de déploiements du cloud.

4.5. Modèles de services du cloud

4.5.1. Plate-forme en tant que service (PaaS)

La capacité offerte au consommateur consiste à déployer sur l'infrastructure cloud par le consommateur des applications créées à l'aide de langages de programmation, de bibliothèques, de services et d'outils pris en charge par le fournisseur. Le consommateur ne peut ni gérer ou contrôler l'infrastructure cloud sous-jacente, y compris le réseau, les serveurs, systèmes d'exploitation, ou de stockage, mais a le contrôle sur les applications déployées et éventuellement paramètres de configuration pour l'environnement d'hébergement d'applications. (Par exemple, les services d'hébergements web mutualisés).

4.5.2. L'infrastructure en tant que service (IaaS)

La capacité offerte au consommateur consiste à provisionner le traitement, le stockage, les réseaux et d'autres ressources informatiques fondamentales où le consommateur peut déployer et exécuter des logiciels arbitraires, qui peuvent inclure des systèmes d'exploitation et des applications. Le consommateur ne gère ni ne contrôle l'infrastructure cloud sous-jacente mais contrôle les systèmes d'exploitation, le stockage et les applications déployées ; et éventuellement un contrôle limité des composants réseau sélectionnés.

4.5.3. Le logiciel en tant que service (SaaS)

La capacité offerte au consommateur consiste à utiliser les applications du fournisseur fonctionnant sur une infrastructure cloud. Les applications sont accessibles à partir de divers dispositifs clients par le biais d'une interface de client légères, telle qu'un navigateur ou d'une interface de programme. Le consommateur ne gère ni ne contrôle l'infrastructure cloud sous-jacente, y compris le réseau, les serveurs, les systèmes d'exploitation, le stockage ou même les applications individuelles, à l'exception peut-être des paramètres de configuration d'application spécifiques aux utilisateurs. (Par exemple, l'encyclopédie en ligne Wikipédia).

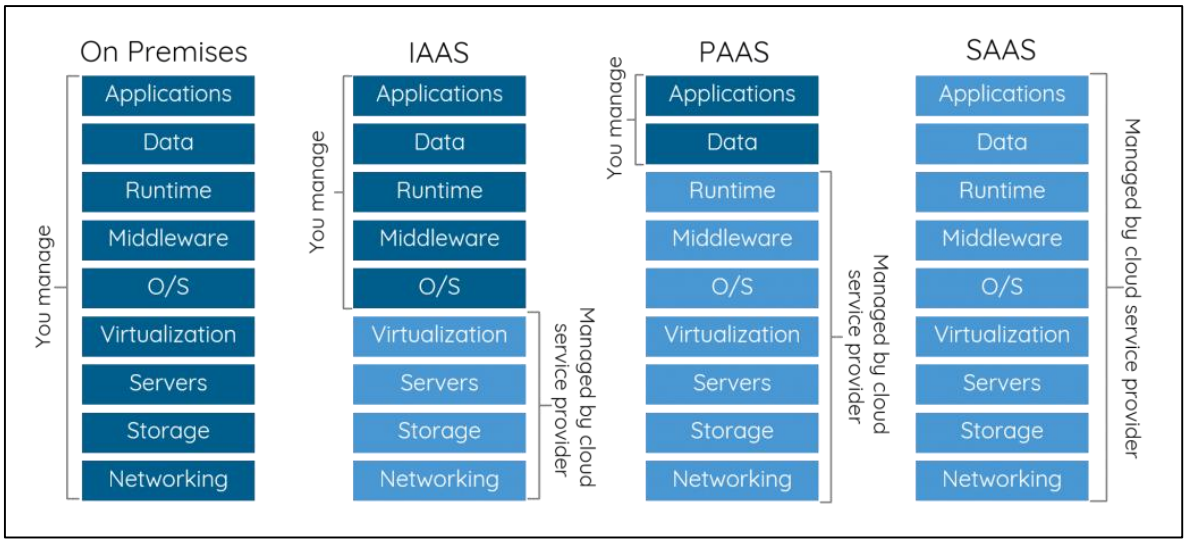


Figure 1.2 : comparaison des différents modèles de services du cloud.

4.5.4. La blockchain en tant que service (BaaS)

4.5.4.1. Définition

La blockchain en tant que service (en anglais Blockchain-As-A-Service ou BaaS) est une suite de services tiers offerte par fournisseurs de services cloud aux entreprises qui développent des applications blockchain. Ces services tiers sont un développement relativement nouveau dans le domaine de la technologie blockchain qui est en pleine croissance. L'activité de la technologie blockchain a bien dépassé son utilisation la plus connue dans les transactions de cryptomonnaie et s'est élargie pour traiter les transactions sécurisées de toutes sortes. En conséquence, il existe une demande de services d'hébergement.

4.5.4.2. Principe et fonctionnement des BaaS [6]

Les consommateurs et les entreprises sont de plus en plus disposés à s'adapter à la technologie blockchain. Cependant, les complexités techniques et les frais généraux opérationnels impliqués dans la création, la configuration et l'exploitation d'une blockchain et la maintenance de son infrastructure agissent souvent comme une barrière.

Le fournisseur de services BaaS propose un moyen externe pour mettre en place toute la technologie et l'infrastructure de blockchain nécessaires moyennant des frais. Une fois créé, le fournisseur continue de gérer les opérations back-end complexes pour le client.

Le fournisseur de services BaaS propose généralement des activités de support telles que la gestion de la bande passante, l'allocation appropriée des ressources, les exigences d'hébergement et les fonctionnalités de sécurité des données. L'opérateur BaaS permet au client de se concentrer sur le travail principal, la fonctionnalité de la blockchain.

4.6. Fournisseurs de BaaS connus

On compte actuellement onze (11) fournisseurs actuellement en concurrence dans l'espace Cloud Computing et le modèle BaaS, parmis eux :

- Microsoft azure BaaS
- IBM Blockchain
- Oracle Blockchain Network
- AWS Amazon Managed Blockchain
- SAP Blockchain Applications and Services
- Alibaba Blockchain as a Service
- Huawei Cloud Blockchain services

Microsoft est impliqué dans des projets de blockchain depuis au moins 2015, et offre aujourd'hui une multitude de services sur sa plateforme cloud Azure, dont le plus notable est l'Azure Blockchain Workbench. Si Microsoft est globalement en tête, elle est suivie de très près par IBM, qui est légèrement en tête en termes d'innovation. Les deux sociétés ont une longueur d'avance sur les autres fournisseurs en termes d'offres BaaS.

4.7. Avantages du cloud computing

- Les fournisseurs d'hébergement dans le Cloud sont chargés d'acheter, d'héberger et d'entretenir, au sein de leur propre structure, les composants matériels et logiciels nécessaires. Les utilisateurs des services épargnent à la fois les investissements financiers et les tracas de maintenance liés aux investissements matériels qu'aurait engendrés le développement de ces services sur leur propre site.

- Libre-service via une interface Web Les utilisateurs des services peuvent, d'une part, amorcer des fonctions de services spécifiques et, d'autre part, augmenter ou diminuer leur niveau d'utilisation des services, via une interface Web, en réduisant au minimum les échanges avec le fournisseur de services.

- Les utilisateurs ne paient que les services qu'ils utilisent. Ce type de facturation peut entraîner des économies substantielles par rapport à l'approche traditionnelle, dans laquelle le développement sur site de structures informatiques est généralement prévu pour des scénarios d'utilisation maximale mais demeure sous-exploité la plupart du temps.

- En règle générale, les fournisseurs de services de Cloud Computing sont équipés des infrastructures nécessaires pour développer leur service à grande échelle. Pour les utilisateurs de services Cloud, cela signifie que le Cloud peut facilement s'adapter à la croissance de l'entreprise.

5. Conclusion

Dans ce chapitre, nous avons détaillé les concepts informatiques nécessaires à la réalisation de notre projet, ainsi que les concepts de base sur lesquels la technologie blockchain s'appuie elle-même. On peut maintenant poursuive et décrire en détail cette technologie.

Chapitre 2

Sécurité informatique, cryptographie et Blockchain

Chapitre 2 : Sécurité informatique, la cryptographie et Blockchain

1. Introduction générale

Dans ce chapitre, nous allons présenter en premier lieu les concepts informatiques de base sur lesquelles la technologie blockchain et notre projet s'appuient. Nous aborderons principalement l'architecture distribuée sur laquelle la blockchain s'appuie pour fonctionner ainsi que le cloud computing qui va nous servir à implémenter notre propre blockchain et l'application mobile qui gère les transferts monétaire entre les utilisateurs.

2. La sécurité informatique [7]

2.1. Définition

La sécurité informatique englobe toutes les pratiques liées à la protection des systèmes informatiques, réseaux, systèmes d'informations, services électroniques, ainsi que toutes les données qu'ils contiennent ou qui circulent à travers eux à fin de prévenir d'éventuels cyberattaques et dommages, d'empêcher des infiltrations, assurer la protection et restauration des données et des systèmes, et d'assurer leurs disponibilité, intégrité, authenticité et confidentialité.

2.1. Importance de la sécurité informatique

Au cours des dernières années et en raison de l'augmentation sans précédent des cyberattaques, les organisations se préoccupent de plus en plus de la sécurité informatique et de la cyber sécurité et accordent de plus en plus d'importance à celles-ci pour identifier et éliminer les vulnérabilités, se protéger contre les intrus, empêcher que des informations sensibles et privilégiées et des données personnelles ne tombent entre les mains de personnes extérieures et protéger les ordinateurs physiques.

2.2. Objectifs de la sécurité informatique

Les objectifs principaux de la sécurité informatique sont :

- **La confidentialité :** toutes les données importantes ne doivent être accessibles qu'aux personnes ou aux systèmes qui possèdent la permission.
- **L'intégrité :** les systèmes informatiques ainsi que les informations qu'ils contiennent doivent continuer à être disponibles, complets et intacts.

- **La disponibilité :** tous les systèmes, services et informations doivent être accessibles à la demande de l'entreprise ou de ses clients.
- **Authenticité :** Garantir la validité d'une transmission, donnée ou son expéditeur. Cela signifie vérifier que les sources et les données n'ont pas été altérée et sont authentiques.
- **La non-répudiation :** Empêche l'émetteur ou le receveur de nier avoir transmis ou reçu une donnée ou message.

3. La cryptologie et cryptographie [8]

Introduction

Les échanges faits à travers le réseau peuvent être interceptées et altérées, il faut donc garantir la sécurité de ces données. C'est à ce niveau que la cryptographie intervient. Le but de la cryptographie moderne est de fournir un certain nombre de services de sécurité et de traiter plus généralement des problèmes de sécurité dans les domaines de la communication, transmission et stockage des données.

Définition de la cryptologie

La cryptologie est l'étude des crypto-systèmes (la science des codes secrets). Elle se divise en deux disciplines fondamentales :

La cryptographie :

C'est la discipline qui s'attache à protéger des messages (assurant confidentialité, authenticité et intégrité) en s'aidant souvent de secrets ou clés.

La cryptanalyse :

C'est la discipline est la discipline sœur de la cryptographie. Elle se penche sur les processus de décryptage et de déchiffrement des messages. Elle est également utilisée lors de la conception des nouvelles techniques cryptographiques pour tester leurs forces de sécurité et leur robustesse.

Objectif de la cryptographie

L'objectif principal de la cryptographie est de sécuriser les données importantes sur un stockage ou lors de leur transmission sur un support qui peut ne pas être totalement sécurisé lui-même, tel qu'un réseau informatique.

Les Services principaux de la cryptographie

- **La confidentialité :** S'assurer que personne ne peut lire le message à l'exception du destinataire prévu. Les données sont gardées secrètes de ceux qui n'ont pas les informations d'identification appropriées.
- **L'intégrité :** Assurer au destinataire que le message reçu n'a en aucun cas été modifié par rapport à l'original.

- **L'authenticité :** S'assure de l'identité de l'expéditeur, confirme au destinataire que les données reçues proviennent d'un expéditeur identifié et vérifié. Elle peut également fournir une assurance sur d'autres paramètres liés aux données tel que la date et l'heure de création / transmission.

- **La non-répudiation :** C'est le mécanisme qui sert à prouver que l'expéditeur a vraiment envoyé le message.

Terminologie de la cryptographie :

La cryptographie utilise différents termes, Nous allons les définis ci-dessous :

- **Algorithme de cryptage :** Il s'agit d'un processus mathématique qui produit un texte chiffré pour tout texte clair reçu en entrée, en utilisant une clé de chiffrement donnée.

- **Texte chiffré :** Le résultat obtenu via l'algorithme de cryptage, une version cryptée (brouillée) du texte en clair produit par l'algorithme de cryptage en utilisant une clé de cryptage spécifique.

- **Algorithme de décryptage :** C'est un processus mathématique qui produit un texte en clair unique pour n'importe quel texte chiffré et clé de déchiffrement. Il s'agit d'un algorithme cryptographique qui prend un texte chiffré et une clé de déchiffrement en entrée, et produit un texte en clair.

- **Clé de cryptage :** C’est une valeur connue de l'expéditeur. L'expéditeur entre la clé de chiffrement dans l'algorithme de chiffrement avec le texte en clair afin de calculer le texte chiffré.
- **Clé de décryptage :** C'est une valeur connue du récepteur. La clé de déchiffrement est liée à la clé de chiffrement, mais ne lui est pas toujours identique. Le récepteur entre la clé de déchiffrement dans l'algorithme de déchiffrement avec le texte chiffré afin de calculer le texte en clair.

Algorithmes et protocoles cryptographiques

Algorithmes de chiffrement faibles

Les premiers algorithmes utilisés pour le chiffrement d'une information étaient assez rudimentaires dans leur ensemble. Ils consistaient notamment au remplacement de caractères par d'autres. La confidentialité de l'algorithme de chiffrement était donc la pierre angulaire de ce système pour éviter un décryptage rapide.

Exemples d'algorithmes de chiffrement faibles

- ROT13 (rotation de 13 caractères, sans clé)
- Chiffre de César (décalage de trois lettres sur la gauche)
- Chiffre de Vigenère (introduit la notion de clé)

Algorithmes de cryptographie symétrique

Les algorithmes de chiffrement symétrique se fondent sur une même clé pour chiffrer et déchiffrer un message. (Voir figure 2.1) L'un des problèmes de cette technique est que la clé, qui doit rester totalement confidentielle, doit être transmise au correspondant de façon sûre. La mise en œuvre peut s'avérer difficile, surtout avec un grand nombre de correspondants car il faut autant de clés que de correspondants.

Principe et fonctionnement :

La clé publique est mise à la disposition de quiconque désire chiffrer un message. Ce dernier ne pourra être déchiffré qu'avec la clé privée, qui doit rester confidentielle (Voir figure 2.1).

La cryptographie asymétrique est également utilisée pour assurer l'authenticité d'un message. L'empreinte du message est chiffrée à l'aide de la clé privée et est jointe au message. Les destinataires déchiffrent ensuite le cryptogramme à l'aide de la clé publique et retrouvent l'empreinte. Cela leur assure que l'émetteur est bien l'auteur du message. On parle alors de signature numérique.

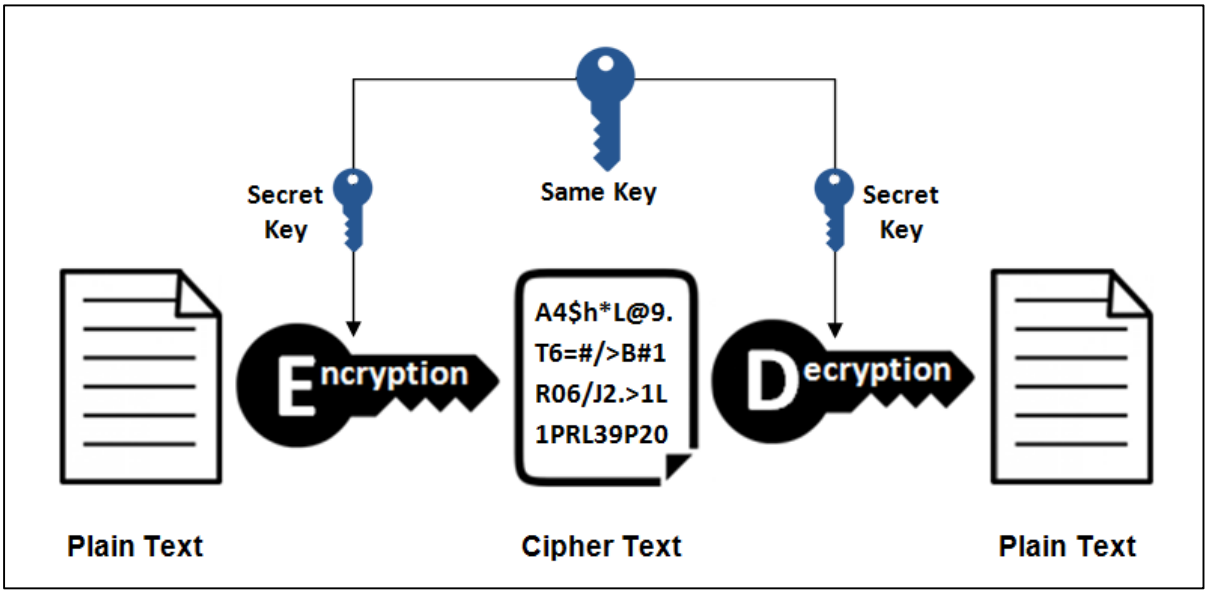


Figure 2.1 : Schéma illustrant le fonctionnement de l'algorithme de chiffrement **symétrique**

Exemples d'algorithmes de chiffrement symétrique

- Le Chiffrement de Vernam : le seul offrant une sécurité théorique absolue, à condition que la clé ait au moins la même longueur que le message à chiffrer, qu'elle ne soit utilisée qu'une seule fois et qu'elle soit totalement aléatoire.
- Le Chiffrement DES
- Le Chiffrement 3DES
- Le Chiffrement AES
- Le Chiffrement RC4
- Le Chiffrement RC5
- Le Chiffrement MISTY1

Algorithmes de cryptographie asymétrique

Pour résoudre le problème de l'échange de clés, la cryptographie asymétrique a été mise au point dans les années 1970. Elle se base sur le principe de deux clés :

- Une clé publique, permettant le chiffrement.
- Une clé privée, permettant le déchiffrement.

Principe et fonctionnement :

La clé publique est mise à la disposition de quiconque qui désire chiffrer un message. Ce dernier ne pourra être déchiffré qu'avec la clé privée, qui doit rester confidentielle (Voir figure 2.2).

La cryptographie asymétrique est également utilisée pour assurer l'authenticité d'un message. L'empreinte du message est chiffrée à l'aide de la clé privée et est jointe au message. Les destinataires déchiffrent ensuite le cryptogramme à l'aide de la clé publique et retrouvent l'empreinte. Cela leur assure que l'émetteur est bien l'auteur du message. On parle alors de signature numérique.

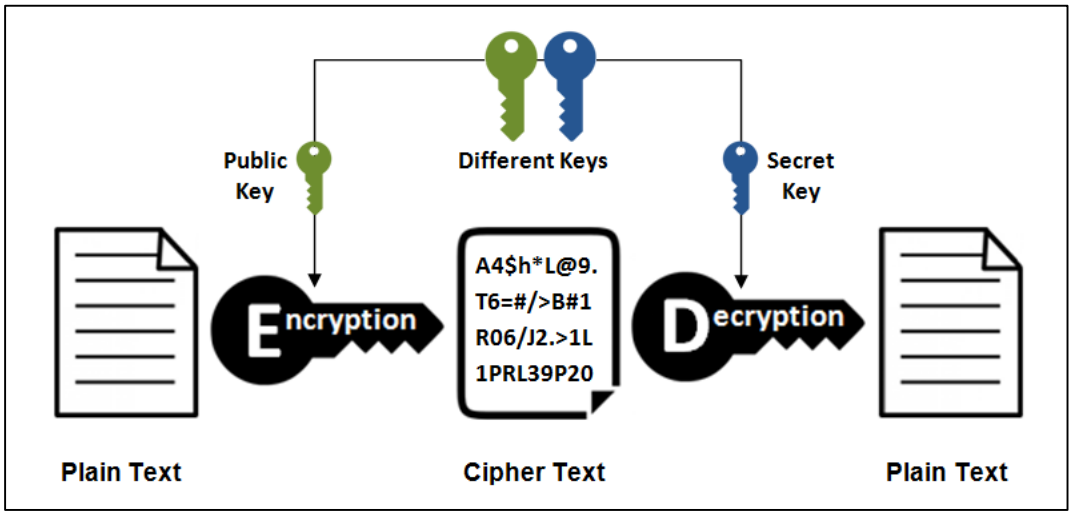


Figure 2.2 : Schéma illustrant le fonctionnement de l'algorithme de chiffrement asymétrique

Exemples algorithmes de cryptographie asymétrique très utilisés

- Le Chiffrement RSA (chiffrement et signature).
- Le Chiffrement DSA (signature).
- Protocole d'échange de clés Diffie-Hellman (échange de clé).

Inconvénients de la cryptographie asymétrique

Le principal inconvénient de RSA et des autres algorithmes à clés publiques est leur grande lenteur par rapport aux algorithmes à clés secrètes. RSA est par exemple 1000 fois plus lent que DES. En pratique, dans le cadre de la confidentialité, on s'en sert pour chiffrer un nombre aléatoire qui sert ensuite de clé secrète pour un algorithme de chiffrement symétrique. C'est le principe qu'utilisent des logiciels comme PGP par exemple.

La plupart des algorithmes de cryptographie asymétrique sont vulnérables à des attaques utilisant un calculateur quantique, à cause de l'algorithme de Shor. La branche de la cryptographie visant à garantir la sécurité en présence d'un tel adversaire est la cryptographie post-quantique.

3.1. Signature numérique

Définition

Une signature numérique est un analogue électronique d'une signature écrite, la signature numérique peut être utilisée pour garantir que le signataire revendiqué a bien signé les informations. De plus, une signature numérique peut être utilisée pour détecter si les informations ont été modifiées ou non après leur signature (c'est-à-dire pour détecter l'intégrité des données signées). Ces assurances peuvent être obtenues si les données ont été reçues dans une transmission ou extraites du stockage.

Principe et fonctionnement

Un algorithme de signature numérique comprend un processus de génération de signature et un processus de vérification de signature. Un signataire utilise le processus de génération pour générer une signature numérique sur des données ; un vérificateur utilise le processus de vérification pour vérifier l'authenticité de la signature.

Chaque signataire possède une clé publique et une clé. Comme le montre la figure 2.3, la clé privée est utilisé dans le processus de génération de signature. Le propriétaire de la paire de clés est la seule entité autorisée à utiliser la clé privée pour générer des signatures numériques. Afin d'empêcher d'autres entités de prétendre être le propriétaire de la paire de clés et d'utiliser la clé privée pour générer des signatures frauduleuses, la

clé privée doit rester secrète. La clé publique est utilisée dans le processus de vérification de signature (voir Figure 2.3). La clé publique n'a pas besoin d'être gardée secrète, mais son intégrité doit être préservée. Tout le monde peut vérifier un message correctement signé à l'aide de la clé publique.

Pour les processus de génération et de vérification de signature, le message (c'est-à-dire les données signées) est converti en une représentation de longueur fixe du message au moyen d'une fonction de hachage approuvée. Le message original et la signature numérique sont mis à la disposition d'un vérificateur.

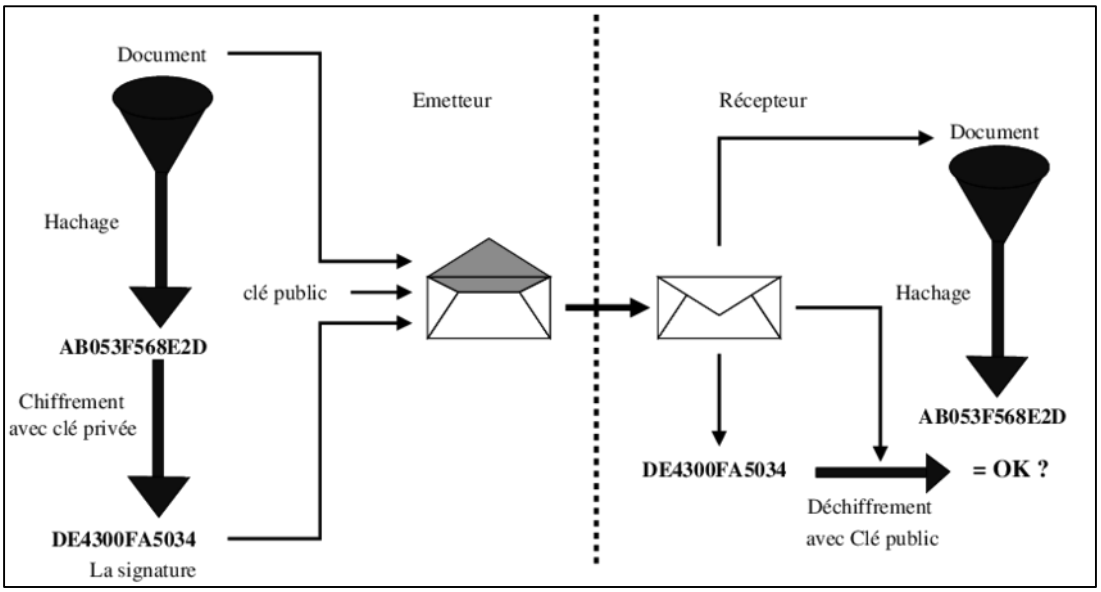


Figure 2.3 : Schéma illustrant le fonctionnement de l'algorithme de signature numérique

3.2. Fonctions de hachage

Définition :

Une fonction de hachage est une fonction qui convertit une chaîne de bits de longueur arbitraire (variable) à une chaîne de bits longueur fixe appelée l'empreinte ou le condensé. Il est impossible de la déchiffrer pour revenir à l'ensemble d'origine, ce n'est donc pas une technique de chiffrement.

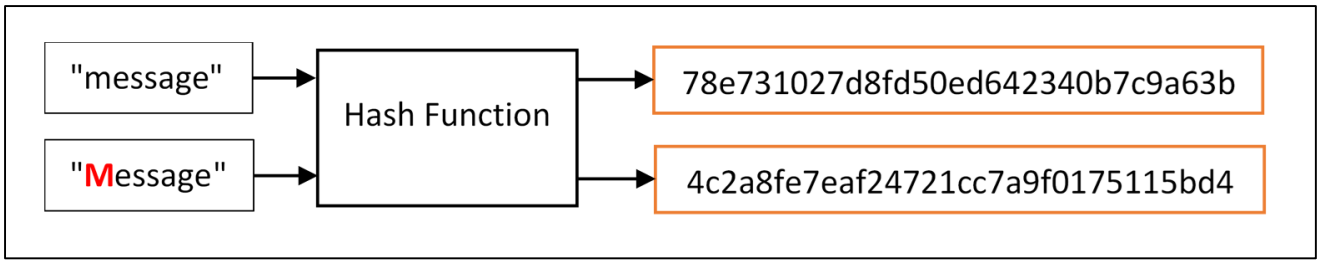


Figure 2.4 : Schéma illustrant le fonctionnement de la fonction de hachage.

Caractéristiques des fonctions de hachage

Les fonctions de hachage cryptographique ont ces propriétés de sécurité importantes :

Unidirectionnelles : Cela signifie qu'ils sont à sens unique, il n'y a aucun moyen de calculer la valeur d'entrée correcte étant donné une certaine valeur de sortie (par exemple, étant donné un résumé y, trouvez x tel que hash (x) = y).
Cela signifie aussi que l'on ne peut pas trouver une entrée hachée dans une sortie spécifique. Plus spécifiquement, les fonctions de hachage cryptographique sont conçues de telle sorte qu'étant donné une entrée spécifique, il est impossible de trouver une deuxième entrée qui produit la même sortie (par exemple, étant donné x, trouvez y tel que hash (x) = hash (y)).

Résistantes aux collisions : Cela signifie que l'on ne peut pas trouver deux entrées qui hachent la même sortie. Plus précisément, il est impossible de trouver deux entrées qui produisent le même condensé (par exemple, trouvez un x et un y tel que hash (x) = hash (y)).

Une fonction de hachage cryptographique spécifique est utilisée dans de nombreuses implémentations de la blockchain, c'est le Secure Hash Algorithm (SHA) avec une taille de sortie de 256 bits (SHA-256). De nombreux ordinateurs prennent en charge cet algorithme, ce qui rend le calcul rapide. SHA-256 a une sortie de 32 octets (1 octet = 8bits, 32 octets = 256 bits) et généralement affichés sous forme de chaîne hexadécimale de 64 caractères

Exemples fonctions de hachage très utilisées : MD5, SHA-1, SHA-256 et d'autres.

4. La Blockchain [9] [10] [12]

4.1. Définition

Une blockchain est un registre collaboratif et inviolable qui conserve des enregistrements transactionnels. Les enregistrements transactionnels (données) sont regroupés en blocs. Un bloc est connecté au bloc précédent en incluant un identifiant unique basé sur les données du bloc précédent. En conséquence, si les données sont modifiées dans un bloc, ce sont des changements d'identifiant unique, qui peuvent être vus dans chaque bloc suivant (fournissant une preuve de falsification). Cet effet domino permet à tous les utilisateurs de la blockchain de savoir si les données d'un bloc précédent ont été falsifiées. Puisqu'un réseau blockchain est difficile à modifier ou à détruire, il fournit une méthode résiliente de sauvegarde de données collaborative.

L'ensemble est sécurisé par cryptographie. Par extension, une chaîne de blocs est une base de données distribuée qui gère une liste d'enregistrements protégés contre la falsification ou la modification par les nœuds de stockage ; c'est donc un registre distribué et sécurisé de toutes les transactions effectuées depuis le démarrage du système réparti.

Il existe une analogie avec le réseau Internet, car dans les deux cas les technologies emploient des protocoles informatiques liés à une infrastructure décentralisée. Internet permet de transférer des paquets de données d'un serveur « sûr » à des clients distants (charge aux destinataires de vérifier l'intégrité des données transmises), alors qu'une blockchain permet à la « confiance » de s'établir entre des agents distincts du système. Avec la technologie blockchain, le « tiers de confiance » devient le système lui-même : chaque élément réparti de la blockchain contient les éléments nécessaires à garantir l'intégrité des données échangées (par un algorithme cryptographique).

4.2. Caractéristiques principales [10]

- **Le registre :** la technologie utilise un registre d'ajout pour fournir un historique complet des transactions. Contrairement aux bases de données traditionnelles, les transactions et les valeurs d'une blockchain ne sont pas remplacées.

- **La sécurité :** les blockchains sont sécurisées par cryptographie, ce qui garantit que les données contenues dans le registre n'ont pas été falsifiées et que les données contenues dans le registre peuvent être attestées.

- **Le partage :** le registre est partagé entre plusieurs participants. Cela offre une transparence entre les participants aux nœuds du réseau blockchain.

- **Distribuée :** la blockchain est distribuée. Cela permet de mettre en place un grand nombre de nœuds sur le réseau pour le rendre plus résistant aux attaques de mauvais acteurs. En augmentant le nombre de nœuds, la capacité d'un mauvais acteur à impacter le protocole de consensus utilisé par la blockchain est réduite.

4.3. Contexte et historique [9] [10] [11]

Les idées fondamentales de la technologie blockchain sont apparues à la fin des années 80 et au début des années 90. En 1989, Leslie Lamport a développé le protocole Paxos et, en 1990, a soumis le document « The PartTime Parliament » à ACM Transactions on Computer Systems; l'article a finalement été publié dans un numéro de 1998. Le document décrit un modèle de consensus pour parvenir à un accord sur un résultat dans un réseau d'ordinateurs où les ordinateurs ou le réseau lui-même peuvent ne pas être fiables. En 1991, une chaîne d'informations signée a été utilisée comme registre électronique pour signer numériquement des documents d'une manière qui pouvait facilement montrer qu'aucun des documents signés de la collection n'avait été modifié. Ces concepts ont été combinés et appliqués à la monnaie électronique en 2008 et décrits dans l'article « Bitcoin : A Peer to Peer Electronic Cash System », qui a été publié sous un pseudonyme par Satoshi Nakamoto, puis plus tard en 2009 avec la mise en place du réseau de blockchain de crypto-monnaie Bitcoin. L'article de Nakamoto contenait le modèle que la plupart des systèmes de crypto-monnaie modernes suivent (bien qu'avec des variations et des modifications). Bitcoin n'était que la première de nombreuses applications blockchain.

De nombreux systèmes de paiement électronique existaient avant Bitcoin (par exemple, ecash et NetCash), mais aucun d'entre eux n'a été largement utilisé. L'utilisation d'une blockchain a permis à Bitcoin d'être mis en œuvre de manière distribuée de sorte qu'aucun utilisateur ne contrôle la monnaie électronique et qu'aucun point de défaillance unique n'existait ; cela a favorisé son utilisation. Son principal avantage était de permettre des transactions directes entre les utilisateurs sans avoir besoin d'un tiers de confiance. Il a également permis l'émission de nouveaux cryptomonnaie d'une manière définie aux utilisateurs qui parviennent à publier de nouveaux blocs et à conserver des copies du registre, ces utilisateurs sont appelés mineurs dans Bitcoin. Le paiement automatisé des mineurs ont

permis l'administration distribuée du système sans avoir besoin de s'organiser. En utilisant une blockchain et une maintenance basée sur le consensus, un mécanisme d'auto-régulation a été créé, cela garantissait que seules les transactions et les blocs valides étaient ajoutés à la blockchain.

Dans Bitcoin, la blockchain permettait aux utilisateurs d'être pseudonymes. Cela signifie que les utilisateurs sont anonymes, mais que leurs identifiants de compte ne le sont pas, en outre, toutes les transactions sont visibles publiquement. Cela a effectivement permis à Bitcoin d'offrir un pseudo-anonymat car les comptes peuvent être créés sans aucun processus d'identification ou d'autorisation (de tels processus sont généralement requis par les lois Know-Your-Customer (KYC)).

Étant donné que Bitcoin était pseudonyme, il était essentiel de disposer de mécanismes pour créer la confiance dans un environnement où les utilisateurs ne pouvaient pas être facilement identifiés. Avant l'utilisation de la technologie blockchain, cette confiance était généralement fournie par des intermédiaires auxquels les deux parties faisaient confiance.

4.4. Les différents types de blockchain et leurs gouvernances [10]

La Blockchain historique (Publique)

La Blockchain « historique » est la Blockchain publique. C'est-à-dire une Blockchain que n'importe qui dans le monde peut lire, chacun peut lui envoyer des transactions et s'attendre à ce qu'elles soient incluses dans le registre, du moins tant que ces transactions respectent les règles de cette Blockchain.

C'est le cas du Bitcoin par exemple, pour lequel chacun a libre accès au registre. Chacun participe également librement au processus d'approbation, celui qui permet de décider quel bloc sera ajouté à la chaîne, et qui définit l'état actuel du système.

En tant que substitut aux réseaux centralisés, les Blockchains publiques sont sécurisées par la cryptoéconomie, c'est-à-dire la combinaison d'incitations économiques et de mécanismes de vérification cryptographiques. Autrement dit, chacun est incité à participer à la vérification des transactions par des avantages économiques, comme pour le Bitcoin.

La Blockchain sous contrôle (Semi-Privée) :

On rencontre également des Blockchains « de consortium », dans laquelle le processus d'approbation est contrôlé par un nombre restreint et choisi de nœuds. Par exemple, une quinzaine d'institutions financières pourraient se mettre d'accord et organiser une blockchain dans laquelle un bloc devrait être approuvé par au moins 10 d'entre elles pour être valide. Double modification donc au système originel, puisque non seulement les participants au processus d'approbation sont limités et sélectionnés, mais qu'en plus ce n'est plus la règle de la majorité qui s'impose.

Le droit de lire la Blockchain, c'est-à-dire l'accès au registre, peut alors être public, réservé aux participants, ou bien hybride.

Les Blockchains privées :

Dans les blockchains privées, le processus d'approbation est limité à un unique acteur, bien que les autorisations de lecture par exemple puissent être publiques.

Les Blockchains privées ou de consortium ont plusieurs avantages, attractifs pour les institutions financières, qui peuvent expliquer l'intérêt qu'elles leur portent depuis peu : gouvernance simplifiée, acteurs connus, coûts réduits, rapidité, confidentialité. Le tout sans la perte de contrôle qu'impliquait la version publique issue du Bitcoin.

4.5. La cryptomonnaie [9]

Une crypto-monnaie est une monnaie numérique peer-to-peer, décentralisée, dont la mise en œuvre repose sur les principes de la cryptographie pour valider les transactions et la génération de la monnaie elle-même. Les implémentations de crypto-monnaie utilisent souvent un schéma de preuve de travail pour se prémunir contre la contrefaçon numérique. Bien que plus de 30 spécifications et protocoles de crypto-monnaie différents aient été définis, la plupart sont similaires et dérivés de la première crypto-monnaie entièrement implémentée, le Bitcoin.

4.6. Le mining [9]

Le mining désigne le processus sécurisation de blocs via des processus cryptographiques. On appelle mineur toute personne (particuliers ou sociétés) qui alimente le réseau en puissance de calcul, pour permettre la mise à jour de la base de données décentralisée et participer au processus de sécurisation.

Chaque mineur est rémunéré au prorata de la puissance de calcul qu'il apporte au réseau.

4.7. Principe et fonctionnement [13]

La chaîne de blocs est une forme de mise en œuvre de la solution du « problème de consensus ». Ce problème mathématique consiste à s'assurer qu'un ensemble de composants informatiques fonctionnant de concert, sait gérer des défaillances (ou malveillances) et arrive à produire un consensus.

Le système doit pouvoir maintenir sa fiabilité dans le cas où une part minoritaire des composants enverrait des informations erronées (bug) ou malveillantes (hack), comme dans le cas d'une cryptomonnaie. Pour contourner la vérification de la double dépense par les mineurs du réseau (la double dépense consiste à réaliser deux paiements simultanément : un vers soi-même et un autre vers une victime, l'objectif est de voir le paiement vers la victime inscrit dans la chaîne de blocs suffisamment longtemps pour tromper la victime mais inscrit de sorte qu'il finisse par être entièrement remplacé par le paiement vers soi-même).

La méthode historique pour aboutir à ce type de consensus est « la preuve de travail » (proof of work).

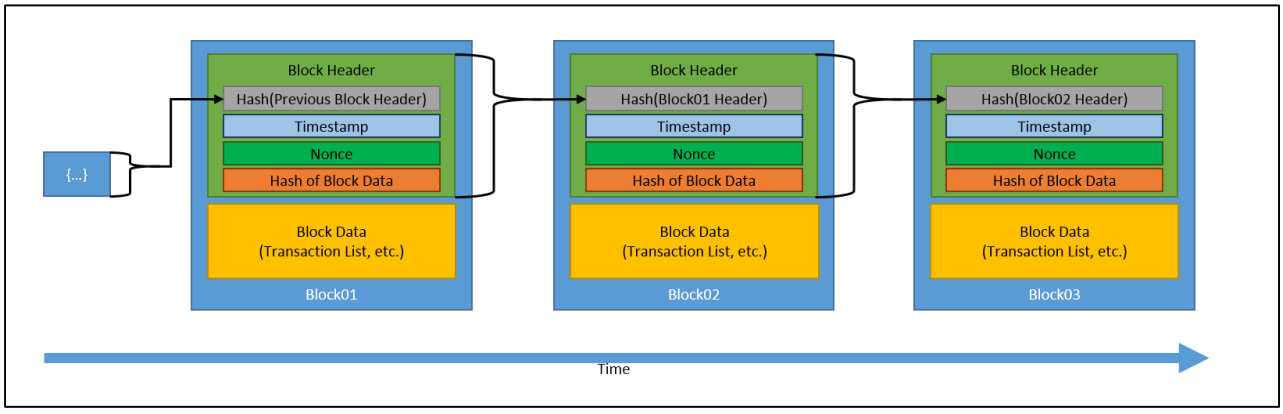


Figure 2.4 : Schéma illustrant la forme d'une blockchain.

4.8. Modèles de consensus de la blockchain [10] [17]

Modèle de consensus de preuve de travail :

Dans le modèle de preuve de travail (Proof-of-Work ou PoW en anglais), un utilisateur (ou mineur) publie le bloc suivant en étant le premier à résoudre un puzzle intensif en calcul (Voir figure 2.5). La solution à ce puzzle est la «preuve» qu'il a bien réalisé travail. Le puzzle est conçu de telle sorte que résoudre le puzzle soit difficile mais vérifier qu'une solution soit valide est facile. Cela permet à tous les autres nœuds de valider facilement les blocs suivants proposés, et tout bloc proposé qui ne répondrait pas au casse-tête serait rejeté.

Une méthode de puzzle courante consiste à exiger que le condensé de hachage d'un en-tête de bloc soit inférieur à une valeur cible. Les nœuds de publication apportent de nombreux petits changements à leur en-tête de bloc (par exemple, en changeant le nonce) en essayant de trouver un condensé de hachage qui répond à l'exigence. Pour chaque tentative, le nœud de publication doit calculer le hachage pour l'en-tête du bloc entier. Hachage de l'en-tête de bloc plusieurs fois devient un processus intensif en calculs.

La valeur cible peut être modifiée au fil du temps pour ajuster la difficulté (vers le haut ou vers le bas) pour influencer la fréquence à laquelle les blocs sont publiés. Par exemple, Bitcoin, qui utilise le modèle de preuve de travail, ajuste la difficulté du puzzle à chaque Blocs pour influencer le taux de publication des blocs à environ une fois toutes les dix minutes.

La puissance de calcul augmente avec le temps, tout comme le nombre de nœuds de publication, de sorte que la difficulté du puzzle augmente généralement. En conséquence, les calculs de résolution d'énigmes nécessitent une consommation de ressources importante.

Un aspect important de ce modèle est que le travail mis dans un puzzle n'influence pas la probabilité de résoudre les énigmes actuelles ou futures car les énigmes sont indépendantes. Cela signifie que lorsqu'un utilisateur reçoit un bloc terminé et valide d'un autre utilisateur, il abandonne son travail actuel et commence à construire sur le bloc nouvellement reçu.

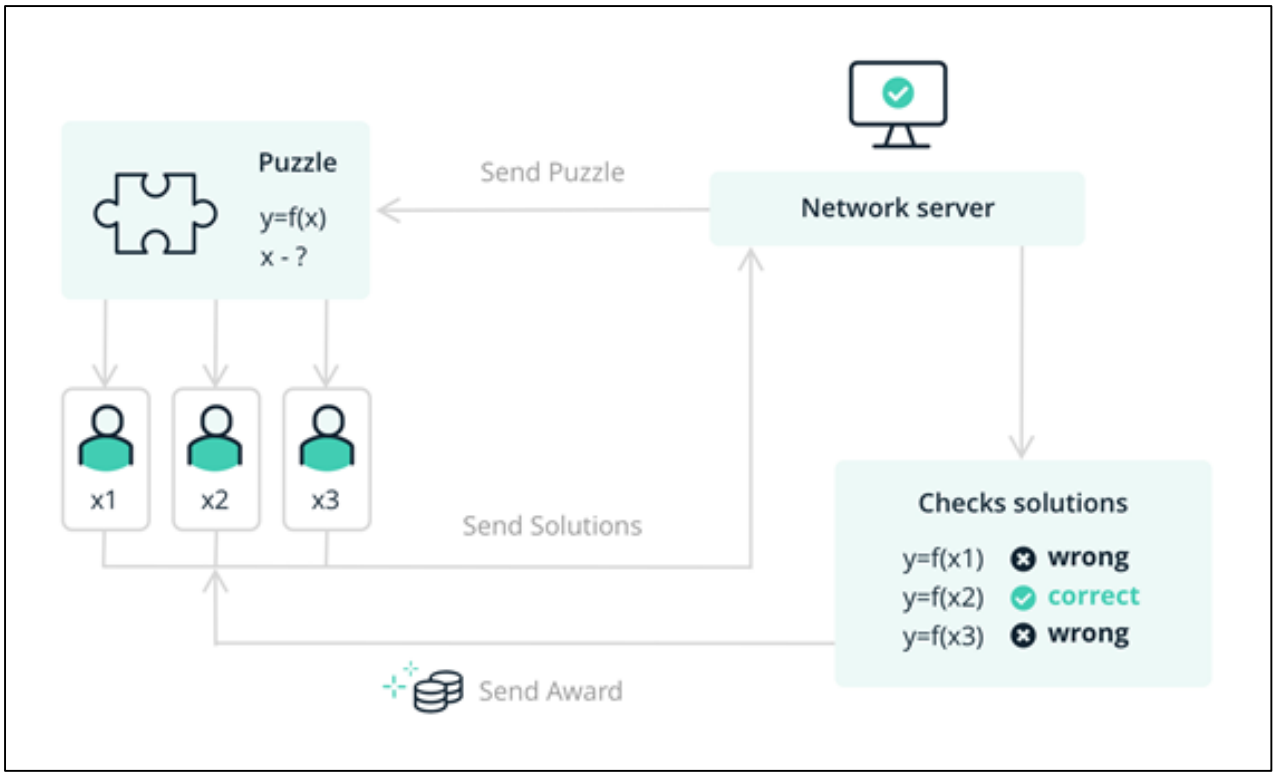


Figure 2.5 : Schéma illustrant le fonctionnement du proof-of-work

Modèle de preuve de consensus d'enjeu :

Le modèle de preuve de participation (PoS) est basé sur l'idée que plus les utilisateurs ont investi dans le système, plus ils voudront probablement que le système réussisse. L'enjeu est souvent une quantité de cryptomonnaie que l'utilisateur du réseau blockchain a investi dans le système (par divers moyens, par exemple en la verrouillant via une transaction, ou en l'envoyant à une adresse spécifique, ou en la gardant dans un portefeuille). Les réseaux blockchain à preuve d'enjeu utilisent le montant de la mise d'un utilisateur comme facteur déterminant pour la publication de nouveaux blocs. Ainsi, la probabilité qu'un utilisateur du réseau blockchain publie un nouveau bloc est liée au rapport de sa participation au montant global de cryptomonnaies du réseau blockchain et du montant de cryptomonnaie investi sur le réseau.

Avec ce modèle de consensus, il n'est pas nécessaire d'effectuer des calculs gourmands en ressources (impliquant le temps, l'électricité et la puissance de traitement) comme indiqué dans la preuve de travail, donc utilise moins de ressources en général, certains réseaux blockchain ont décidé de renoncer à récompenser la création de blocs, ces systèmes sont conçus pour que toute la cryptomonnaie soit déjà distribuée parmi les utilisateurs plutôt que de nouvelles cryptomonnaies générées à un rythme constant. Dans

de tels systèmes, la récompense pour la publication en bloc est généralement le montant que l'utilisateur a fourni pour les frais de transaction.

Modèle de consensus à la ronde

Le modèle de consensus à la ronde (ou Round Robin en anglais) est un modèle de consensus utilisé par certains réseaux de blockchain. Dans ce modèle de consensus, les nœuds créent à tour de rôle des blocs. Le consensus Round Robin a une longue histoire ancrée dans l'architecture des systèmes distribués. Pour gérer les situations où un nœud de publication n'est pas disponible pour publier un bloc à son tour, ces systèmes peuvent inclure une heure limite pour permettre aux nœuds disponibles de publier des blocs afin que les nœuds indisponibles ne provoquent pas d'arrêt et donc bloquer la publication. Ce modèle garantit qu'aucun nœud ne crée la majorité des blocs. Il profite d'une approche simple, un faible besoin en énergie et peu d'énigmes cryptographiques sont utilisées. Puisqu'il y a un besoin que de la confiance entre les nœuds, le round robin ne fonctionne pas bien dans les réseaux blockchain publiques utilisés par la plupart des cryptomonnaies. C'est parce que les nœuds malveillant pourraient ajouter en permanence des nœuds supplémentaires pour augmenter leurs chances de publier de nouveaux blocs. Dans le pire des cas, ils pourraient l'utiliser pour perturber le bon fonctionnement du réseau blockchain.

Modèle de consensus sur la preuve d'autorité / preuve d'identité

Le modèle consensuel de la preuve d'autorité (également appelé preuve d'identité) repose sur la confiance des nœuds de publication grâce à des liens connus avec des identités du monde réel (Par exemple, le réseau d’une entreprise connue). Les nœuds de publication doivent avoir leurs identités prouvée et vérifiable au sein du réseau blockchain (par exemple, identifier les documents qui ont été vérifiés et notariés et inclus dans la blockchain). L'idée est que le nœud de publication jalonne son identité / réputation pour publier de nouveaux blocs. Plus la réputation est faible, moins il y a de chances de pouvoir publier un bloc. Par conséquent, il est dans l'intérêt d'un nœud de publication de maintenir une bonne réputation. Ce modèle ne s'applique qu'aux réseaux blockchain avec des niveaux de confiance élevés.

4.9. Nonce cryptographique [9]

Un nonce cryptographique est un nombre arbitraire qui n'est utilisé qu'une seule fois. Un nonce cryptographique peut être combiné avec des données pour produire différents condensés de hachage par nonce :

Hash (data + nonce) = condensé

Seule la modification de la valeur nonce fournit un mécanisme pour obtenir différentes valeurs de résumé tout en gardant les mêmes données. Cette technique est utilisée dans le modèle de consensus de preuve de travail.

4.10. Adresses et dérivation d'adresses [10]

Certains réseaux blockchain utilisent une adresse, qui est une courte chaîne alphanumérique de caractères dérivés de la clé publique de l'utilisateur du réseau blockchain à l'aide d'une fonction hachage cryptographique, avec quelques données supplémentaires (par exemple, numéro de version, sommes de contrôle). La plupart des implémentations de blockchain utilisent des adresses comme points de terminaison «à» et «de» dans une transaction. Les adresses sont plus courtes que les clés publiques et ne sont pas secrètes.
Une méthode pour générer une adresse consiste à créer une clé publique, à lui appliquer une fonction de hachage cryptographique et à convertir le hachage en texte :

Clé publique → fonction de hachage cryptographique → adresse

Chaque implémentation de blockchain peut implémenter une méthode différente pour dériver une adresse. Pour les réseaux blockchain sans autorisation, qui permettent la création de compte anonyme, l'utilisateur du réseau peut générer autant de paires de clés asymétriques, et donc d'adresses qu'il le souhaite, permettant un degré variable de pseudo-anonymat.

Les adresses peuvent jouer le rôle d’identifiant dans un réseau blockchain pour un utilisateur, et souvent une adresse sera convertie en un code QR (Quick Response Code, un code à barres bidimensionnel pouvant contenir des données arbitraires) pour une utilisation plus facile avec les appareils mobiles.

Les utilisateurs du réseau blockchain peuvent ne pas être la seule source d'adresses au sein des réseaux blockchain. Il est nécessaire de fournir une méthode d'accès à un contrat intelligent une fois qu'il a été déployé au sein d'un réseau blockchain. Pour Ethereum, les contrats intelligents sont accessibles via une adresse spéciale appelé un compte de contrat. Cette adresse de compte est créée lors du déploiement d'un contrat intelligent (l'adresse d'un compte de contrat est calculée de manière déterministe à partir de l'adresse du créateur du contrat intelligent). Ce compte de contrat permet d'exécuter le contrat chaque fois qu'il reçoit une transaction, ainsi que de créer des contrats intelligents supplémentaires à leurs tours.

4.11. Les registres [22]

Un registre (Ledger En anglais) est un ensemble de transactions. Tout au long de l'histoire, les livres de papier et de stylo ont été utilisés pour suivre les échanges de biens et de services. Dans les temps modernes, les registres ont été stockés numériquement, souvent dans de grandes bases de données détenues et gérées par un tiers de confiance centralisé (c'est-à-dire le propriétaire du registre) au nom d'une communauté d'utilisateurs. La propriété des registres centralisés peut être mise en œuvre de manière centralisée ou distribuée (c.-à-d. un seul serveur ou coordination d'un cluster de serveurs).
Il y a un intérêt croissant à explorer la répartition de la propriété du registre. La technologie Blockchain permet une telle approche en utilisant à la fois cette propriété distribuée et une architecture distribuée physique.

4.12. Les blocs

Les différentes transactions enregistrées sont regroupées dans des blocs. Après avoir enregistré les transactions récentes, un nouveau bloc est généré et toutes les transactions vont être validées par les mineurs, qui vont analyser l'historique complet de la chaîne de blocs. Si le bloc est valide, il est horodaté et ajouté à la chaîne de blocs. Les transactions qu'il contient sont alors visibles dans l'ensemble du réseau. Une fois ajouté à la chaîne, un bloc ne peut plus être ni modifié ni supprimé, ce qui garantit l'authenticité et la sécurité du réseau.

Chaque bloc de la chaîne est constitué des éléments suivants :

- Des transactions effectuées sur le réseau.
- Une somme de contrôle « hash », utilisée comme identifiant.
- La somme de contrôle du bloc précédent (à l'exception du premier bloc de la chaîne, appelé bloc de genèse).
- Une mesure de la quantité de travail qui a été nécessaire pour produire le bloc. Celle-ci est définie par la méthode de consensus utilisée au sein de la chaîne, telle que le nonce « preuve de travail ».

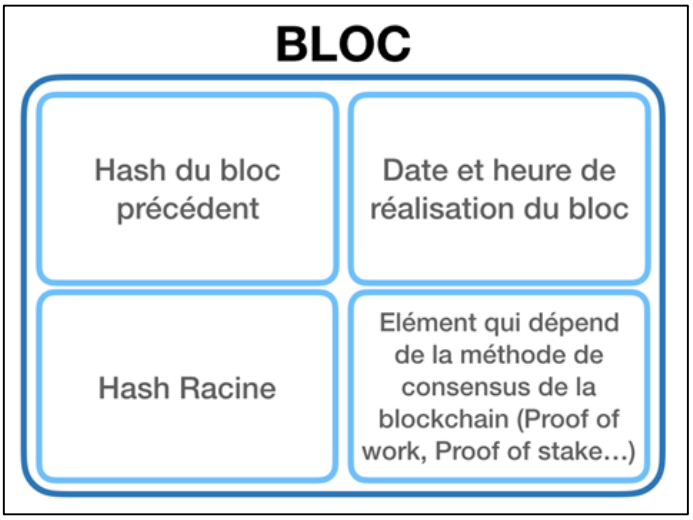


Figure 2.6 : Schéma illustrant un bloc enregistré dans la blockchain.

4.13. Le smart-contract [18]

Les contrats intelligents (en anglais Smart-Contract) est un programme autonome qui, une fois démarré, exécute automatiquement des conditions inscrites en amont dans la blockchain, sans nécessiter d'intervention humaine. Il fonctionne comme toute instruction conditionnelle de type « if – then » (si telle condition est vérifiée, alors telle conséquence s'exécute). Pour déclencher son exécution, un smart-contract se connecte à une base de données jugée fiable, via l'intermédiaire d'un oracle (un service qui fait le lien entre le smart-contract et le monde réel).

Les contrats intelligents visent à assurer une sécurité supérieure à la mise en application de la loi sur les contrats et de réduire les coûts de transaction associés à la passation des contrats.

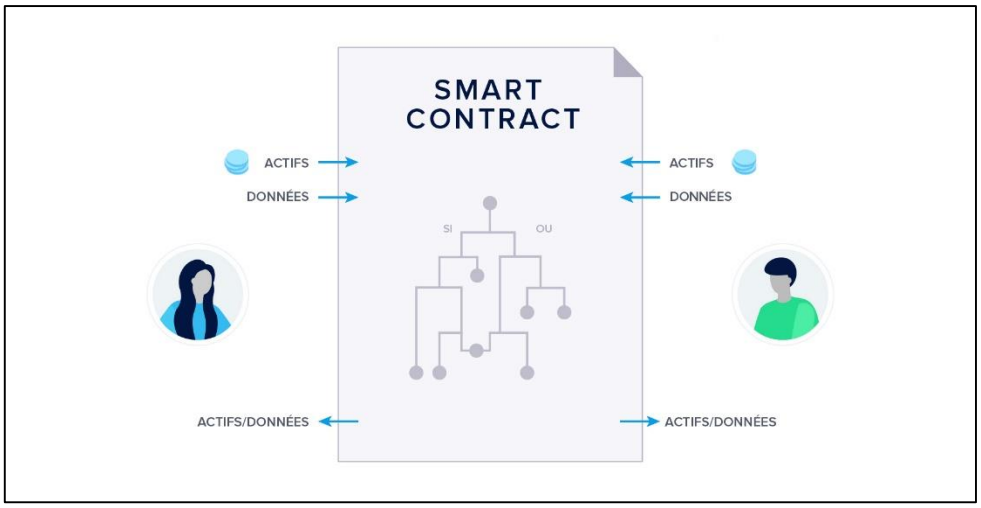


Figure 2.6 : Schéma illustrant le fonctionnement d'un smart-contract.

4.14. Le portefeuille de cryptomonnaie (Wallet) [19]

Un portefeuille de cryptomonnaie (En anglais cryptocurrency wallet ou wallet) est un appareil, un support physique, un programme ou un service qui stocke les clés publiques et/ou privées. En plus de cette fonction de base de stockage des clés, ils offrent le plus souvent également la fonctionnalité de chiffrement et / ou de signature des informations. La signature peut par exemple entraîner l'exécution d'un contrat intelligent, une transaction de crypto-monnaie, l'identification ou la signature légale d'un «document».

4.15. Considérations supplémentaires sur la blockchain [10] [11]

Pour décider d'utiliser ou non une blockchain, il faut prendre en compte des facteurs et déterminer si ces facteurs limitent la capacité d'utilisation en général de la blockchain ou bien d'un type particulier de celle-ci.

Visibilité des données

Les réseaux de blockchain peuvent ou non révéler des données de blockchain publiquement. Les données ne peuvent être disponibles que pour les utilisateurs du réseau blockchain. Il faut Envisager des scénarios dans lesquels les données peuvent être régies par des politiques ou des réglementations (comme les Informations personnelles identifiables (PII) ou le règlement général sur la protection des données (RGPD)). Des données comme celles-ci peuvent ne pas être appropriées pour un stockage au sein d'un réseau blockchain.

Les réseaux blockchain sans autorisation peuvent permettre à quiconque d'inspecter et de contribuer à la blockchain. Les données sont généralement publiques. Cela conduit à plusieurs questions qui doivent être prises en compte. Les données de l'application doivent-elles être disponibles pour toutes les personnes ? Y a-t-il un préjudice à avoir des données publiques ?

Historique transactionnel complet

Certains réseaux blockchain fournissent un historique public complet d'un actif numérique, de la création à chaque transaction dans laquelle il est inclus. Cette fonction peut être bénéfique pour certaines solutions et non bénéfique pour d'autres.

Les entrées erronées (Fake Data Input)

Étant donné que plusieurs utilisateurs contribuent à une blockchain, certains pourraient soumettre de fausses données, imitant les données de sources valides (telles que les données de capteur). Il est difficile d'automatiser la vérification des données qui entrent dans un réseau blockchain.

Les implémentations de contrat intelligent peuvent fournir des vérifications supplémentaires pour aider à valider les données lorsque cela est possible.

Données inviolables et inaltérables

De nombreuses applications suivent le modèle «CRUD» (créer, lire, mettre à jour, supprimer) des fonctions pour la manipulation des données. Avec une blockchain, il n'y a que du «CR» (créer, lire). Il existe des méthodes qui peuvent être utilisées pour mettre à jour des données anciennes si une version plus récente est trouvée, mais il n'y a pas de processus de suppression pour les données d'origine. En utilisant de nouvelles transactions pour modifier et mettre à jour les transactions précédentes, les données peuvent être mises à jour tout en fournissant une histoire complète.

Cependant, même si une nouvelle transaction a marqué une transaction plus ancienne comme «supprimé» - les données seraient toujours présentes dans les données de la blockchain, même si elles ne sont pas affichées dans une application traitant les données.

Transactions par seconde

La vitesse de traitement des transactions dépend fortement du modèle de consensus utilisé. Les transactions actuellement sur de nombreux réseaux de blockchain sans

autorisation ne sont pas exécutés au même rythme que les autres solutions informatiques en raison du temps de publication lent pour les blocs (généralement en secondes, mais parfois en minutes). Ainsi, un ralentissement des applications dépendantes de la blockchain peut se produire pendant une attente de publication des données.

Conformité

L'utilisation de la technologie blockchain n'exclut pas la conformité aux lois et réglementations applicables. Par exemple, il existe de nombreuses considérations relatives à la législation et aux politiques liées aux PII ou au RGPD qui déterminent quelles informations ne doivent pas être placées sur la blockchain. De plus, certains pays peuvent limiter le type de données pouvant être transférées au-delà des frontières. Dans d'autres cas, certaines lois peuvent imposer que la «première écriture» de la transaction financière doit être écrite sur un nœud présent à l'intérieur de leurs frontières. Dans l'un de ces cas, une blockchain publique sans autorisation peut être moins appropriée, avec un approche autorisée ou hybride requise pour satisfaire les besoins réglementaires. Un exemple supplémentaire de lois et de réglementations concerne tout réseau blockchain qui gère les archives fédérales. Les archives fédérales sont soumises à de nombreuses lois et réglementations. Les agences fédérales elles-mêmes doivent suivre des directives fédérales spécifiques lors de l'utilisation de la technologie blockchain.

Autorisations

Pour les réseaux blockchain autorisés, il y a des considérations sur les permissions elles-même :

Granularité

Les autorisations dans le système permettent-elles une granularité suffisante pour des rôles spécifiques dont les utilisateurs peuvent avoir besoin ? (d'une manière telle que l'accès soit basé sur les méthodes de contrôle des rôles pour effectuer des actions au sein du système).

Les réseaux blockchain autorisés permettent des rôles plus traditionnels tels qu'administrateur, utilisateur, validateur, auditeur, etc.

Administration

Qui peut administrer les autorisations ? Une fois que les autorisations sont administrées à un utilisateur, peuvent-ils être facilement révoqués ?

Diversité des nœuds

Un réseau blockchain n'est fort que lorsque tous les nœuds existants participant au réseau. Si tous les nœuds partagent un matériel similaire, logiciel, emplacement géographique et schéma de messagerie alors il existe un certain nombre de risque associé à la possibilité de vulnérabilités de sécurité non découvertes. Ce risque est atténué par la décentralisation du réseau d'appareils hétérogènes, qui peuvent être définies comme «les caractéristiques non partagées entre les nœuds et l'ensemble généralisé »

4.16. Les applications décentralisées [19] [20] [21]

Définition

Une application décentralisée (en anglais Decentralized Application), est une application qui fonctionne sur un réseau décentralisé, par opposition aux applications classiques qui reposent sur des serveurs centralisés.

Elle a comme support un ou plusieurs smart contracts **[6]** déployés sur une blockchain. La partie front-end (interface utilisateur), elle, peut être développée comme sur les applications classiques. La grande majorité des applications décentralisées sont aujourd'hui construites dans l'écosystème de la blockchain Ethereum, celle-ci sont appelées **dApps**.

Le plus souvent, elle présente plusieurs autres caractéristiques tel que :

- Elle n'a pas besoin d'autorité centrale pour fonctionner : elle rend ainsi possible des interactions directes, pair-à-pair, entre utilisateurs, via des smart contracts.
- Les données sont stockées de façon chiffrée et transparente sur une blockchain.
- Elle utilise une cryptomonnaie ou un token.
- Un code informatique entièrement open source.

Les différents types de dApps

Les dAapps qui gèrent de l'argent

Ces applications décentralisées permettent de réaliser des transferts monétaires sans tiers de confiance grâce à la technologie blockchain. Bitcoin est un parfait exemple de ce type d'applications décentralisées.

Les dApps qui gèrent des transferts financiers en lien avec des évènements intervenant dans le monde réel

Ces Dapps, en utilisant une technologie de contrats intelligents permettent d'automatiser des transferts financiers en fonction de la réalisation d'événements dans le monde réel. Par exemple, un fournisseur pourrait recevoir de manière automatique son paiement lorsque les produits sont arrivés à destination chez le client. Un traceur envoie alors un signal sur la blockchain, celui-ci déclenchant le déblocage des fonds pour payer le fournisseur.

Organisation autonome et décentralisée (DAO)

Expliquées simplement, les DAO sont des organisations qui fonctionnent de manière décentralisée sans organe de gestion centralisée. Ces organisations reposent sur des règles techniques qui expliquent comment chaque participant peut voter afin de prendre les décisions inhérentes à l'organisation.

Enjeux des dApps [20]

Un des enjeux est de faire émerger des applications qui soient à la fois autres que le transfert de valeur et la spéculation et qui tirent vraiment partie des propriétés inédites de la blockchain (résistance à la censure, immuabilité des données, etc).

Un des freins spécifiques à l'adoption des DApps, cela étant, réside dans le fait que leurs utilisateurs doivent payer, sur Ethereum, des frais de transaction minimum (le gas) pour que leurs transactions soient inscrites dans la blockchain. Or l'utilisateur veut parfois simplement effectuer une action qui n'a rien à voir avec le transfert d'argent – par exemple, effectuer un vote. Pour que ce vote soit inscrit dans la blockchain il n'y pas d'autre option que de payer le gas pour la transaction. Des méthodes sont en train d'être pensées et construites pour renverser le problème : selon certains, demain les utilisateurs pourraient déléguer à d'autres acteurs le paiement du gas, la gestion des transactions, etc. Les utilisateurs auraient uniquement à effectuer l'action désirée (voter, par exemple) et laisseront un autre acteur enregistrer la transaction sur la blockchain et payer pour celle-ci.

5. Conclusion

Grâce à ce chapitre, nous connaissons maintenant les différents composants de la technologie Blockchain ainsi que des mécanismes de sécurité informatique et de cryptographie qui ont permis son bon fonctionnement. Nous pouvons dès à présent poursuivre sur l'implémentation de notre propre solution.

Chapitre 3
Études et Conception du projet

Chapitre 3 : Conception et études du projet

1. Introduction

Dans ce présent chapitre, nous allons présenter les étapes qui vont nous permettre de construire et de développer notre application.

Nous commencerons d'abord par rappeler les objectifs de notre travail, puis détailler la conception globale de notre application, nous verrons en détail par la suite les différentes parties fondamentales de notre système du coté client et du coté blockchain (réseau) ainsi que des organigrammes des algorithmes clés.

2. Objectifs du projet

Le but notre travail est de reconstruire un système capable de gérer les transactions monétaires en ligne à l'aide de la technologie blockchain. Celui-ci reprendra les mêmes fonctionnalités que les systèmes utilisés actuellement par nos banquse de manière centralisée mais nous rajouterons l'aspect décentralisé apporté par la blockchain ainsi qu'une nouvelle couche de sécurité pour les transactions et les données contenues dans le système de façon à ce qu'il soit inviolable.

3. Etat actuel du système de gestion de transaction

Actuellement les transactions sont traitées de manière totalement centralisée, le schéma ci-dessous (Figure 3.1) résume le chemin qu'une transaction effectue pour être traitée. On voit que la transaction passe obligatoirement par le réseau de la banque pour qu'elle soit traitée. Puis le système central de la banque s'occupe de mettre à jour la base de données.

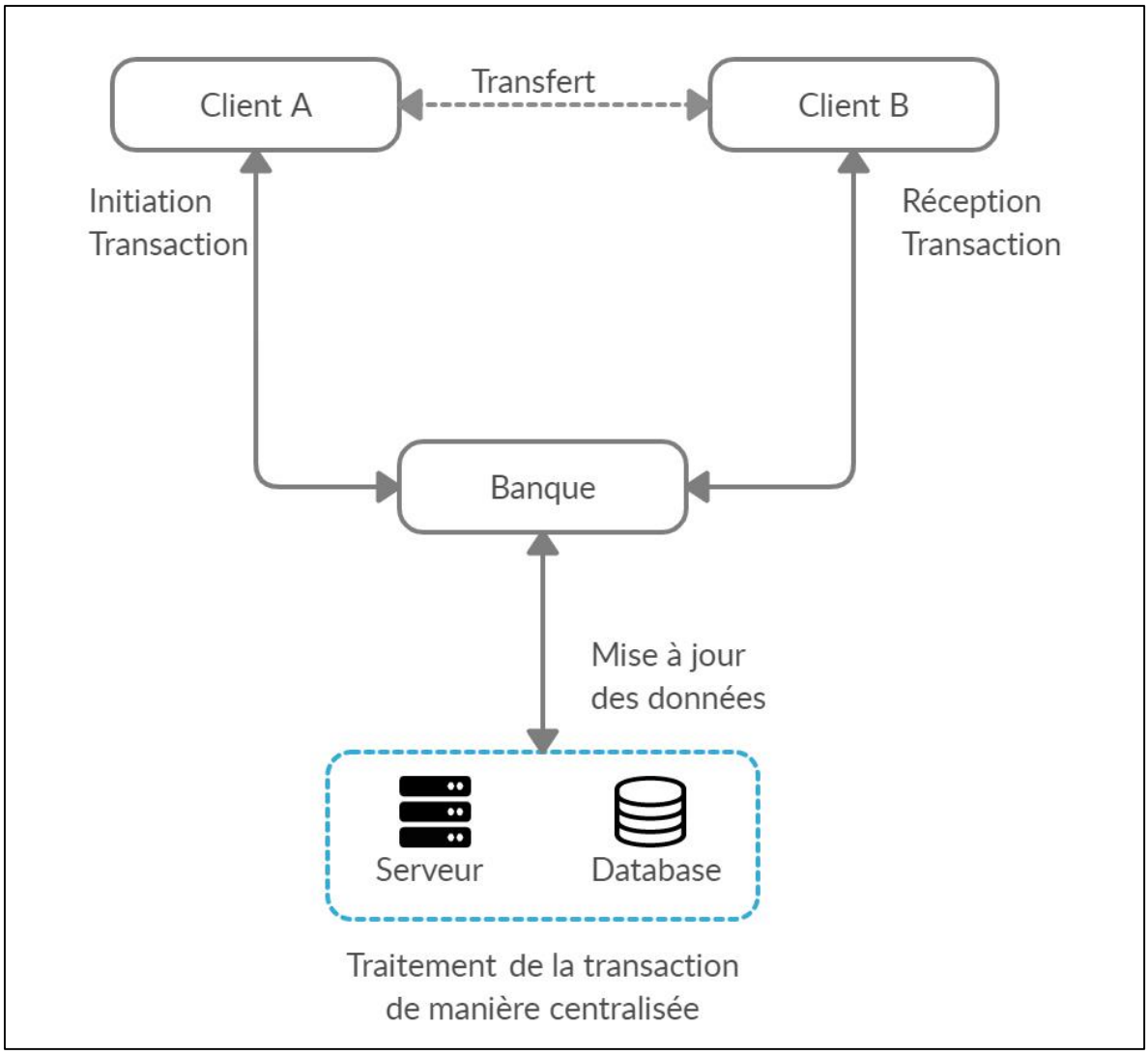


Figure 3.1 : Schéma illustrant le processus de transaction de manière centralisée

4. Description de la solution proposée

Nous souhaitons à travers ce projet retirer l'aspect « centralisation ». Pour ce faire nous allons ajouter au système actuel une partie Blockchain composée d'un réseau et d'un registre totalement distribués (Voir Figure 3.2) qui s'occuperont de traiter les transactions entre clients. Le système central de la banque sera connecté au réseau Blockchain et sera mis à jour en temps réél dès qu'une transaction sera validée et enregistrée au niveau de la blockchain.

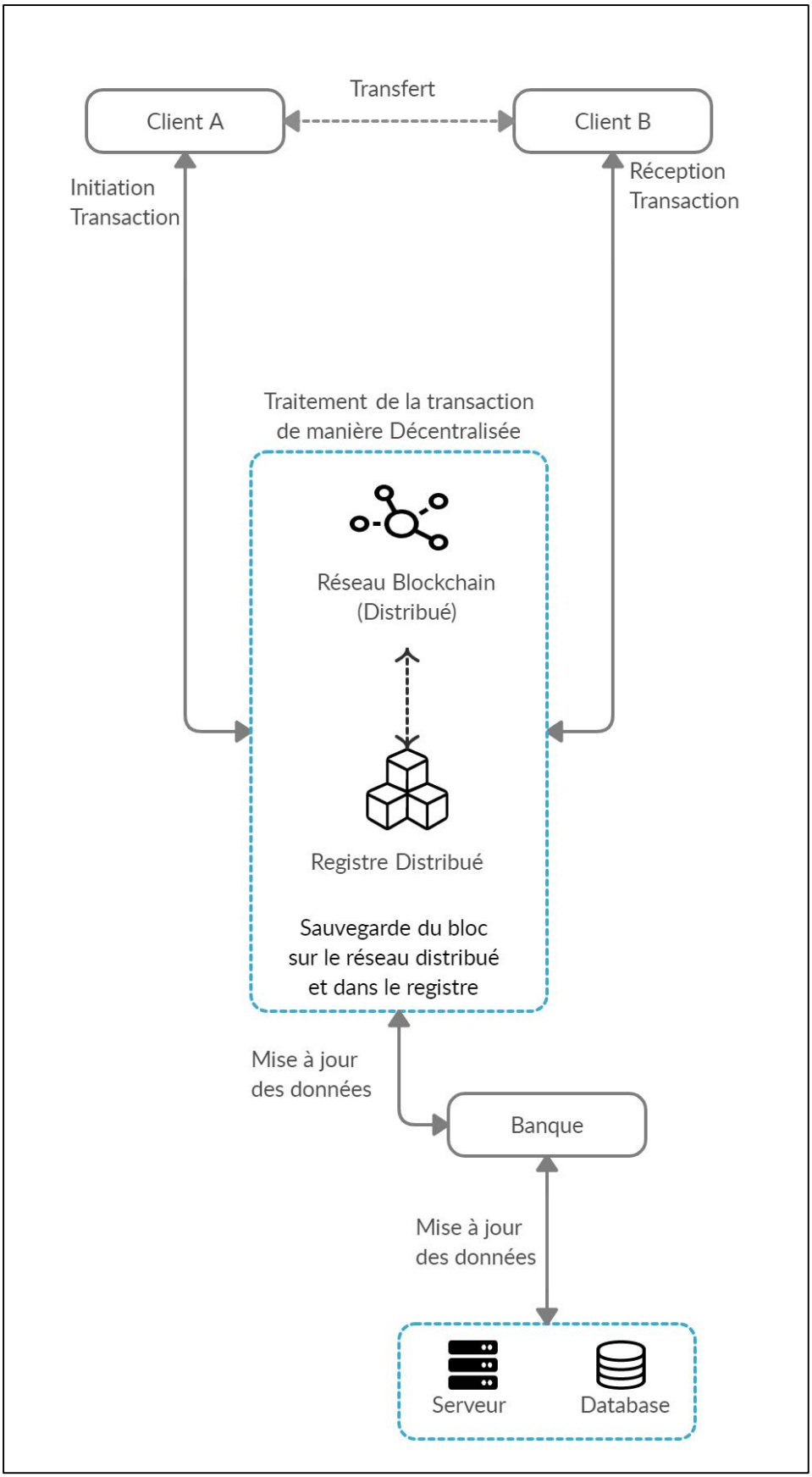


Figure 3.2 : Schéma illustrant la solution décentralisée pour le traitement de transaction

5. Application blockchain

Le schéma ci-dessous (Figure 3.3) illustre le fonctionnement global de notre système. Le service Firebase est utilisé lors de l'authentification des utilisateurs, via leurs numéros de téléphones. Quand un utilisateur souhaite envoyer une somme depuis son portefeuille (wallet), celui-ci initie la transaction au niveau de l'application mobile. La transaction est signée avec sa clé privée de se trouvant au niveau du stockage sécurisé de son téléphone, ensuite elle sera transmise au nœud de réception de notre blockchain privée qui se trouve au niveau du service blockchain de Microsoft Azure qui va s'occuper de vérifier la signature et d'appeler le smart contract (compilé et déployé à travers Remix IDE). Une fois le bloc créé et validé, le nœud le communique au réseau distribué publique d'Ethereum ou il est ajouté à la blockchain (partagé vers tous les autres nœuds et stocké dans le registre de la blockchain) et devient ainsi immutable et inviolable.

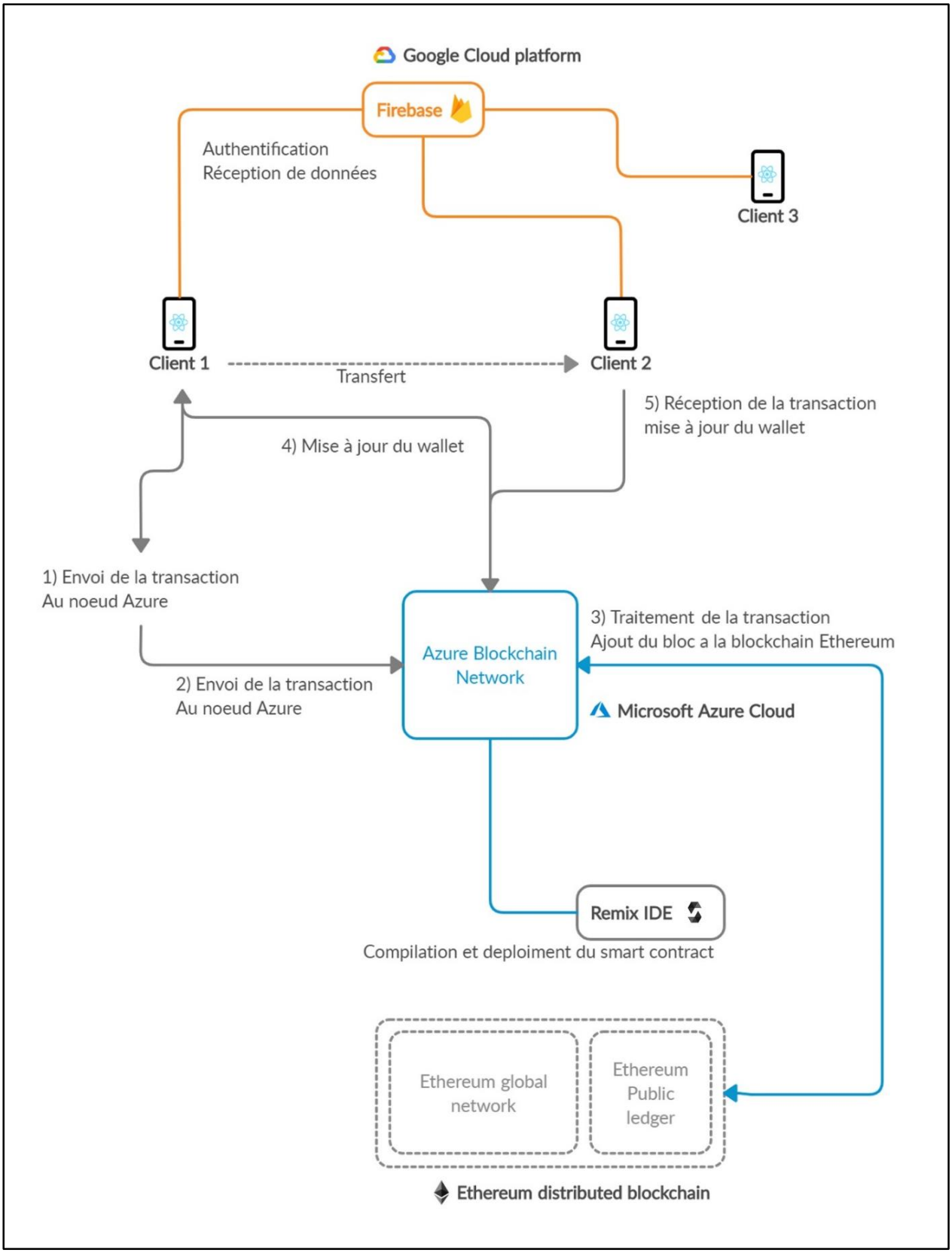


Figure 3.3 : Schéma illustrant le fonctionnement global de notre application décentralisée

6. Application client

Le schéma ci-dessous (Figure 3.4) illustre en détail les différentes parties de l'application client, ainsi que les différentes bibliothèques utilisées.

App.js représente le point d'entrée de notre application, c'est le premier fichier qui exécuté au lancement de l'application et qui s'occupe de lier toutes les différentes parties de l'application.

La partie React Navigation, représente la navigation au niveau de l'application. Elle se divise en deux élément, « Auth navigation » qui s'occupe de la navigation des utilisateurs non-connectés, et gére les vue d'authentification à l'application. Le second élément « Main navigation » gére la navigation dans la partie principale de l'application (une fois l'utilisateur connecté). Quant à « root Navigation » il s'agit de la navigation globale, celle-ci s'occupe de vérifier l'état de l'utilisateur (connecté ou non) pour le renvoyer vers la bonne partie de l'application.

La partie Redux détaille les différentes parties que la bibliothèque Redux gère. Cette partie montre comment est découpée la structure Redux en premier lieu, les actions sont divisées en deux groupes : les actions « authentification » et les actions « blockchain ».

Les actions « authentification » déclenchent les fonctions liées à l'authentification (Connexion, Déconnexion Récupération de données …) et sont connectée à la partie Firebase.

Les actions « blockchain » quant à eux s'occupe de déclencher les actions liées à la blockchain (récupération de wallets, création de transaction).

Le store Redux s'occupe de stocker tous les états de l'application et de distribuer les données attendues aux différentes parties de l'application.

Les bibliothèques « EthereumTx » et « Web3 » sont utilisée pour assurer la création, signature et envoi de transactions ainsi que la connexion au réseau blockchain.

La partie « React-Native » représente quant à elle, les différentes vues qui composent l'application, celles-ci comportent plusieurs de composants. Cette partie est liée principalement au visuel de l'application, la gestion des interactions utilisateur et l'affichage des données.

La partie « Ethereum distributed blockchain » représente l'intégralité du réseau Ethereum global, ainsi que le Ledger « registre » qui nous servira de bases de données pour les différentes transactions. Celle-ci communiquera de manière constante avec notre réseau Azure Blockchain

qui sera utilisé pour communication entre l'Ethereum distributed blockchain et notre Application décentralisée.

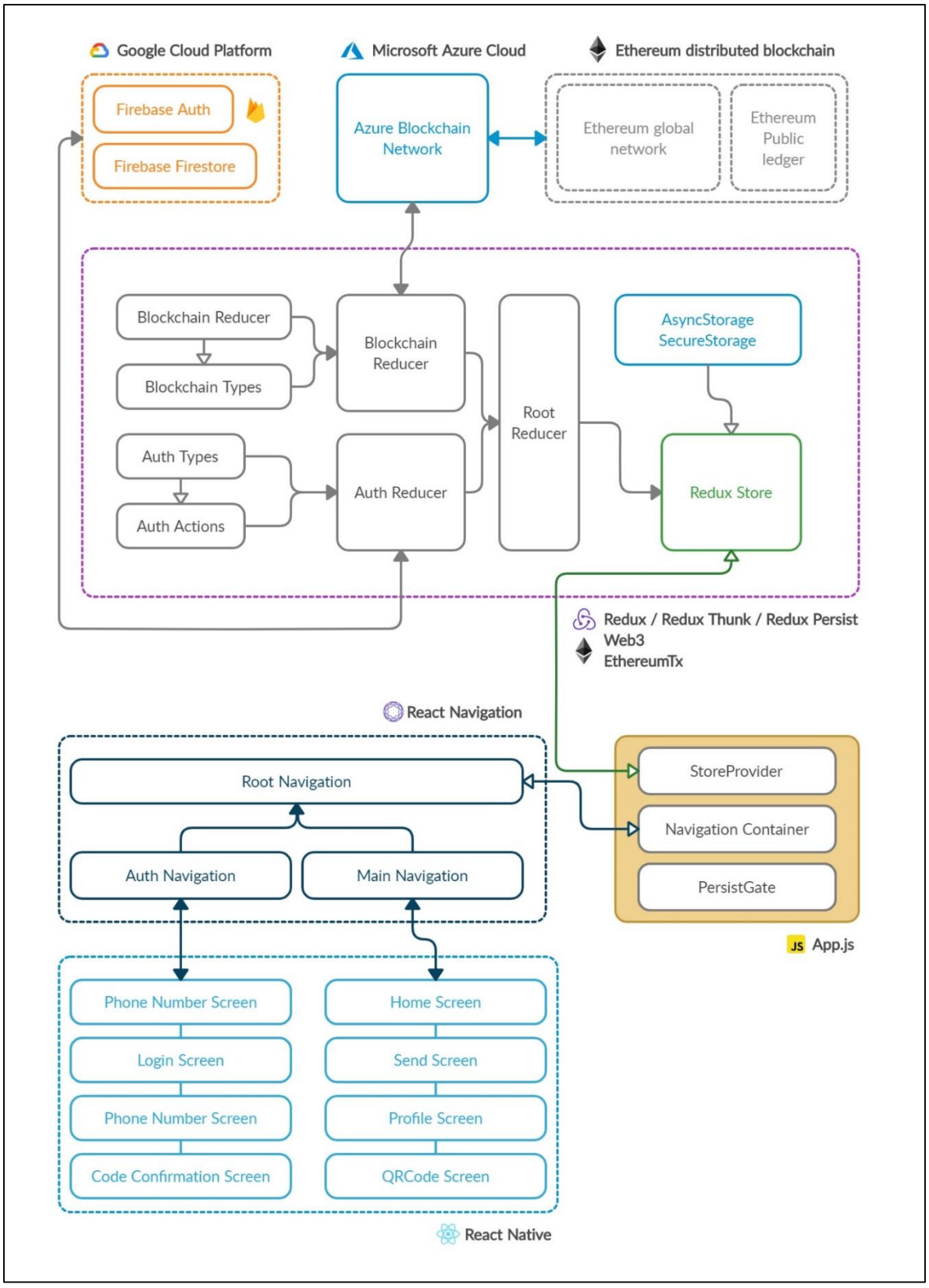


Figure 3.4 : Schéma détaillant les différentes parties de notre application client

7. Partie authentification

Le schéma fonctionnel ci-dessous (Figure 3.5.1), illustre comment les différentes parties de notre système interagissent entre elles lors de l'authentification des utilisateurs.

L'utilisateur est invité à rentrer son numéro de téléphone pour commencer, il reçoit un SMS contenant un code de validation qu'il devra saisir dans l'application pour confirmer son authentification (utilisation de l'authentification à deux facteurs). En cas d'inscription, une nouvelle clé privée et adresse de wallet sont générées et stockées au niveau du stockage sécurisé du téléphone.

La figure 3.5.2 représente l'organigramme de notre algorithme d'authentification.

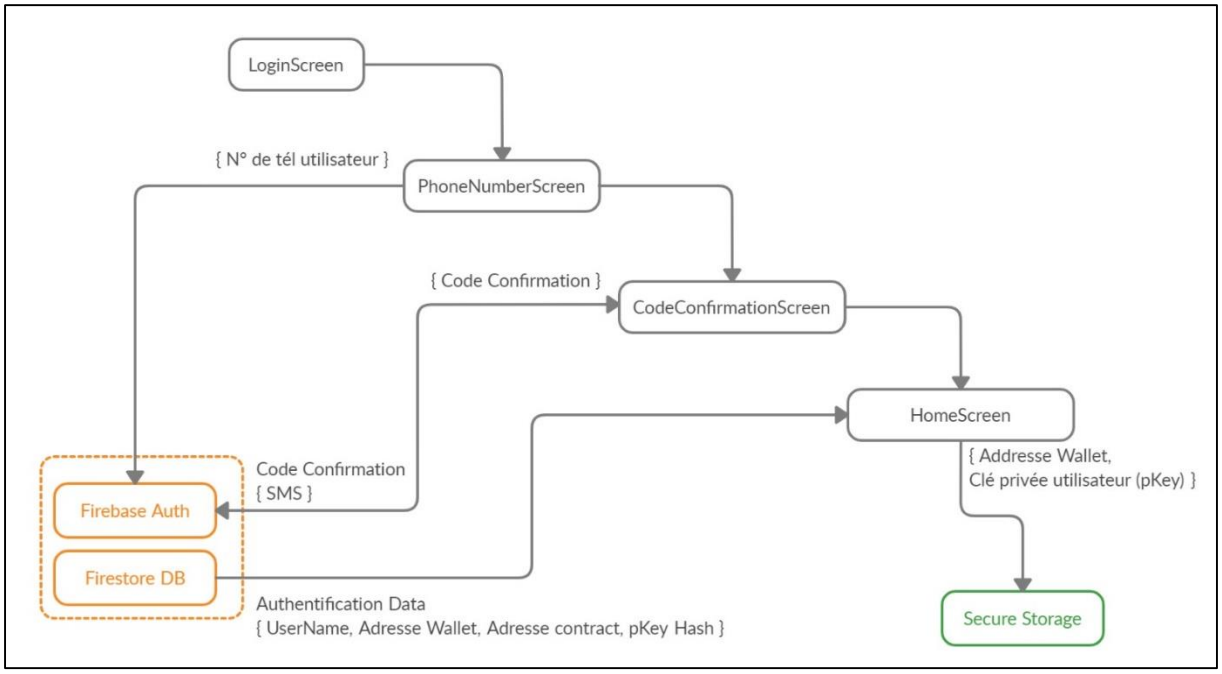


Figure 3.5.1 : Schéma illustrant l'interaction des différentes parties de l'application lors de l'authentification

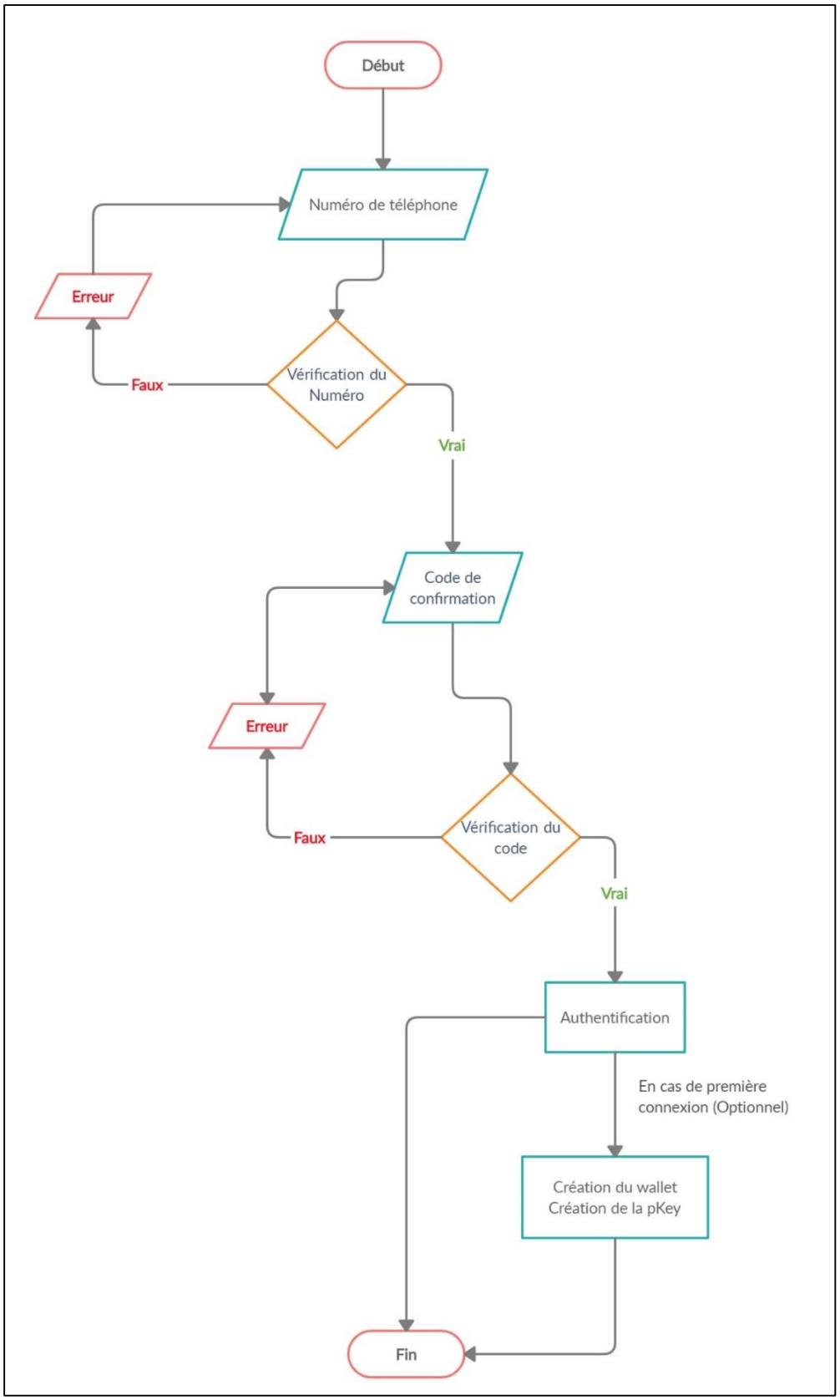


Figure 3.5.2 : Organigramme illustrant la partie authentification

8. Partie principale de l'application

Le schéma ci-dessous (Figure 3.6.1) nous montre comment, une fois l'utilisateur connecté, l'application procède à la récupération des données du wallet utilisateur.

Quand l'utilisateur est connecté, l'application s'occupe de comparer le hash de la clé privée stocké dans la base de données, ainsi que le hash de la clé privée contenu dans le stockage du téléphone pour s'assurer de l'authenticité de l'appareil. Ensuite l'application se connecte à la blockchain et retourne à l'utilisateur ses données de portefeuille.

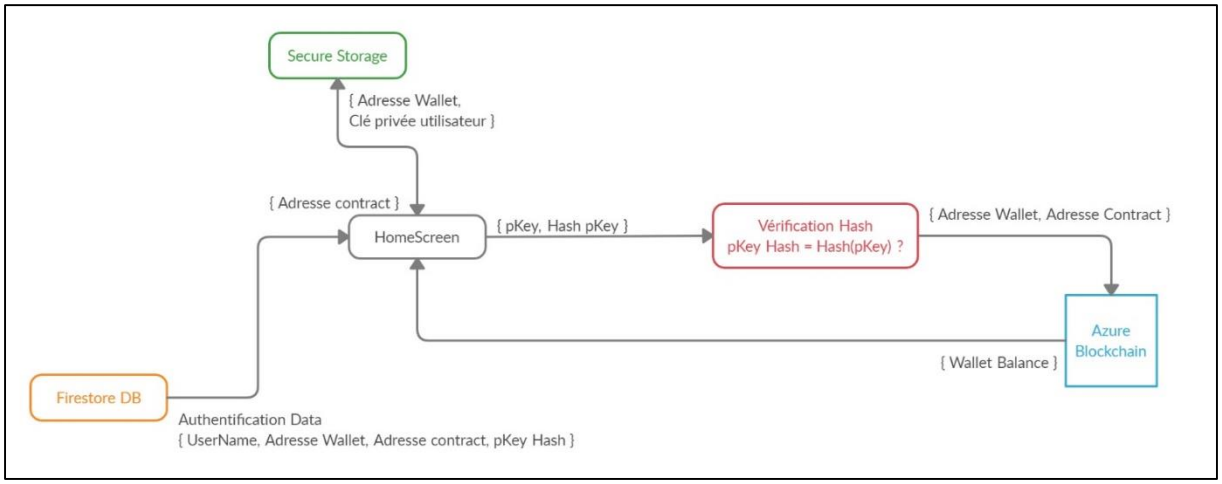


Figure 3.6.1 : Schéma illustrant la récupération des données du Wallet de l'utilisateur connecté

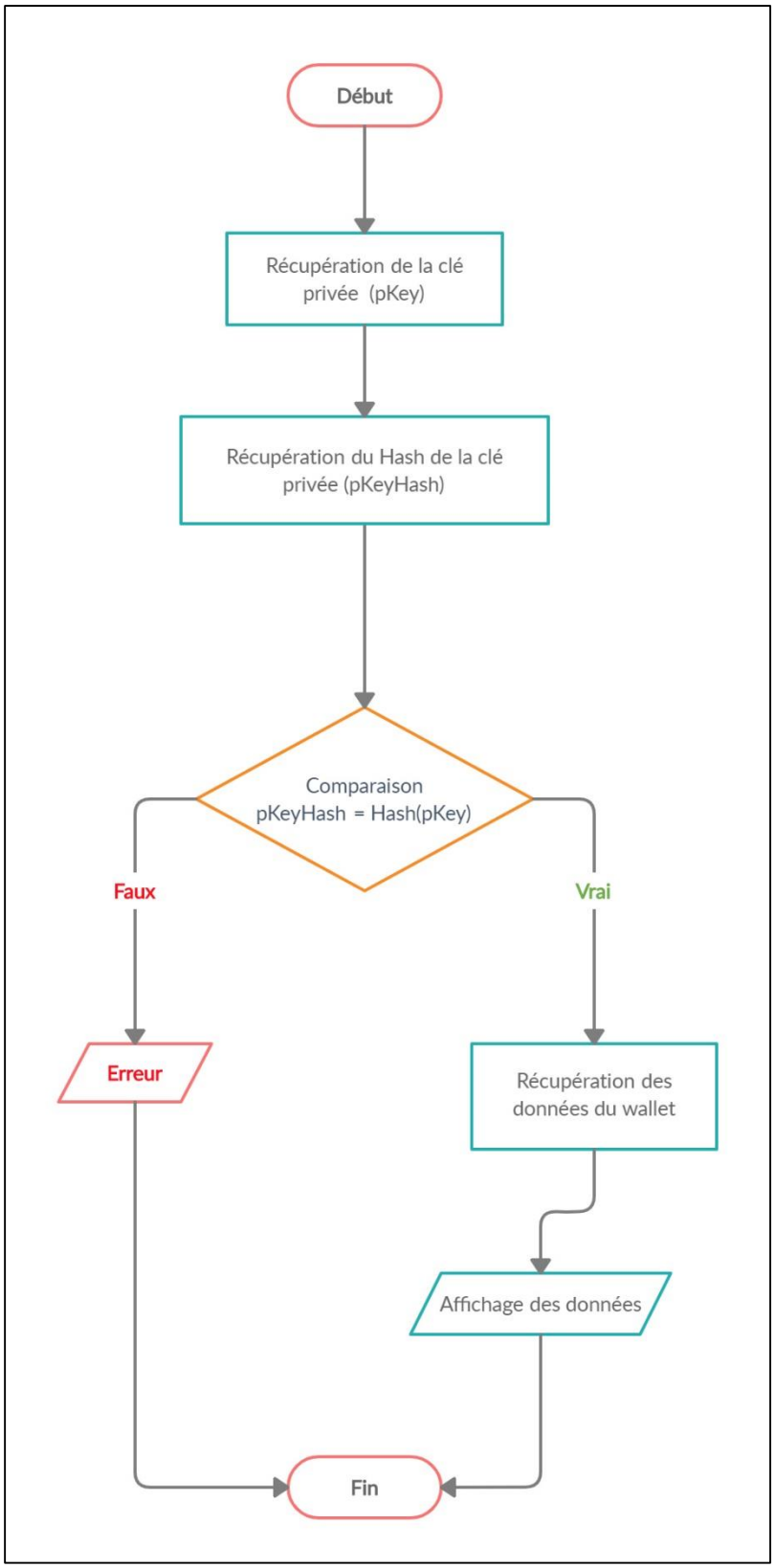
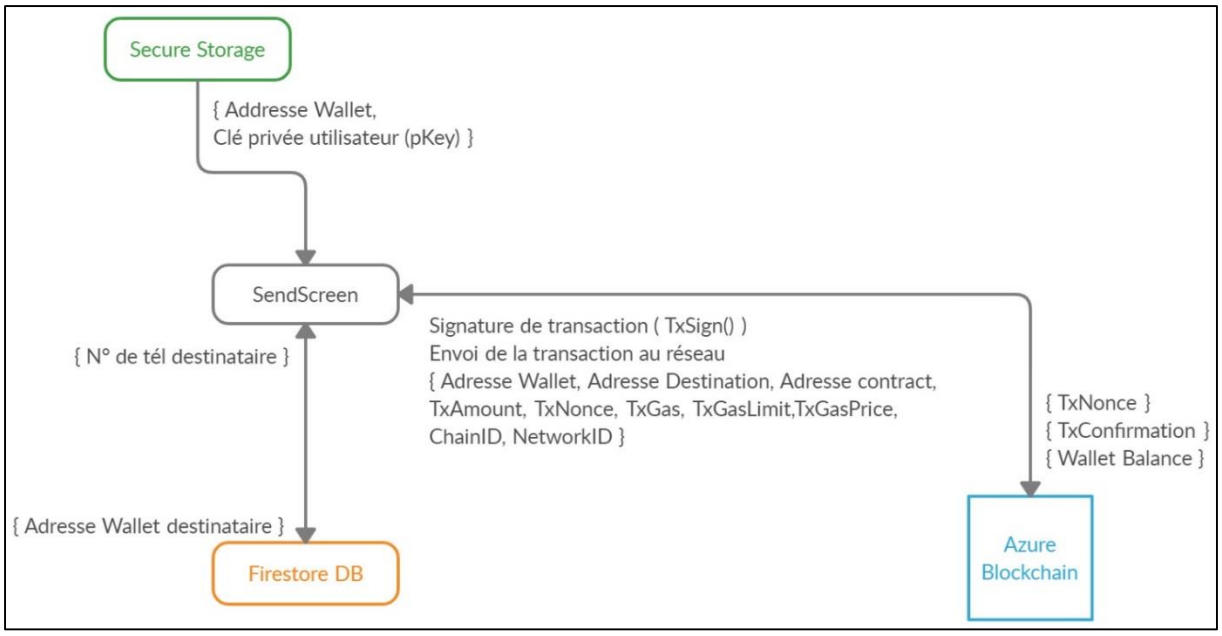


Figure 3.6.2 : Organigramme de la partie principale de l'application

9. Fonction « envoi de transactions »

Le schéma ci-dessous (Figure 3.7.1) nous détaille, la procédure d'envoi de transactions au réseau. L'utilisateur authentifié, commence par saisir le numéro de téléphone du destinataire, la base de données s'occupe de retourner l'adresse wallet correspondante. L'application crée une nouvelle transaction et la signe avec la clé privée de l'utilisateur se trouvant dans le stockage sécurisé de l'application, rajoute des données tel que : l'adresse wallet de l'utilisateur, l'adresse du wallet du destinataire, le nonce, le gas, l'identifiant du réseau, l'identifiant de la blockchain, puis l'envoie au réseau blockchain. Le réseau blockchain s'assure de la validité de la transaction, rajoute la transaction dans un bloc et une fois le tout validé, renvoie une confirmation ainsi que la balance actuelle du wallet utilisateur. (Le nonce est envoyé en premier lieu par la blockchain, car il est nécessaire lors de la création de la transaction).



Tx* : Transaction.

Figure 3.7.1 : Schéma illustrant l'envoi de transaction au réseau

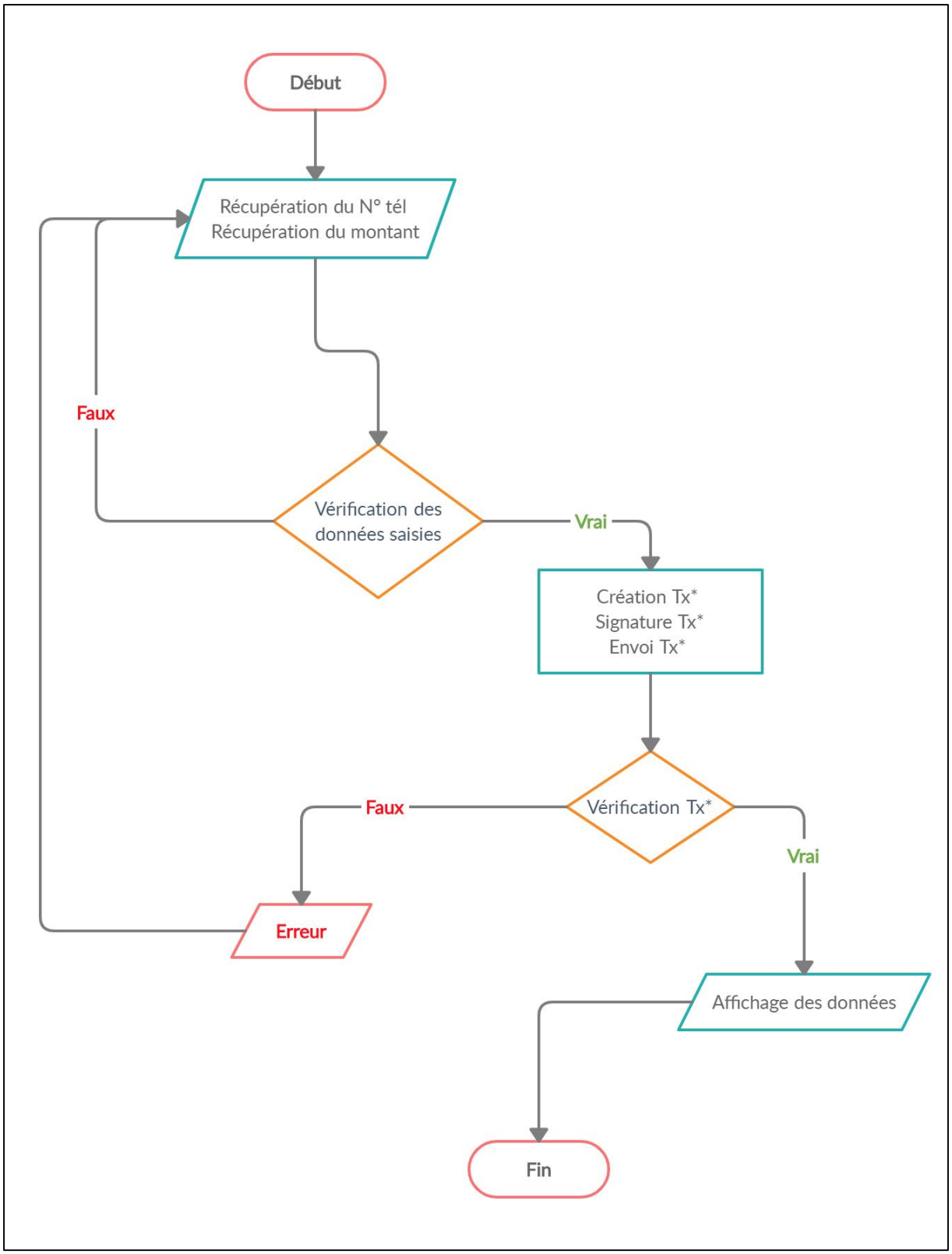


Figure 3.7.2 : Organigramme de la fonction d'envoi de transactions

10. Partie blockchain [22]

Le schéma ci-dessous (Figure 3.8) montre les étapes de la création du bloc et de la validation de la transaction au niveau de la blockchain.

Comme on peut le constater, au cours de l'étape 1 le nœud A, initie la transaction Tx en précisant l'adresse de réception et le montant du transfert puis signe cette dernière avec sa clé privée pour certifier et confirmer son identité dans le réseau.

Dans l'étape 2 le bloc est créé et confirmé grâce à la signature sTx puis il est envoyé vers le réseau Blockchain pour la vérification, validation et l'ajout à la chaîne de blocs.

Dans l'étape 3, le Smart Contract qui doit prendre en charge la transaction s'active et lance ses fonctions de Vérification des transactions Tx en employant plusieurs paramètres tels que le montant, l'adresse de réception, la balance de l'émetteur ainsi que la signature de ce dernier ; et donc après vérifier et valider la transaction Tx, le bloc est ajouté à la blockchain et est diffusé sur tout le réseau pour mettre à jour cette dernière.

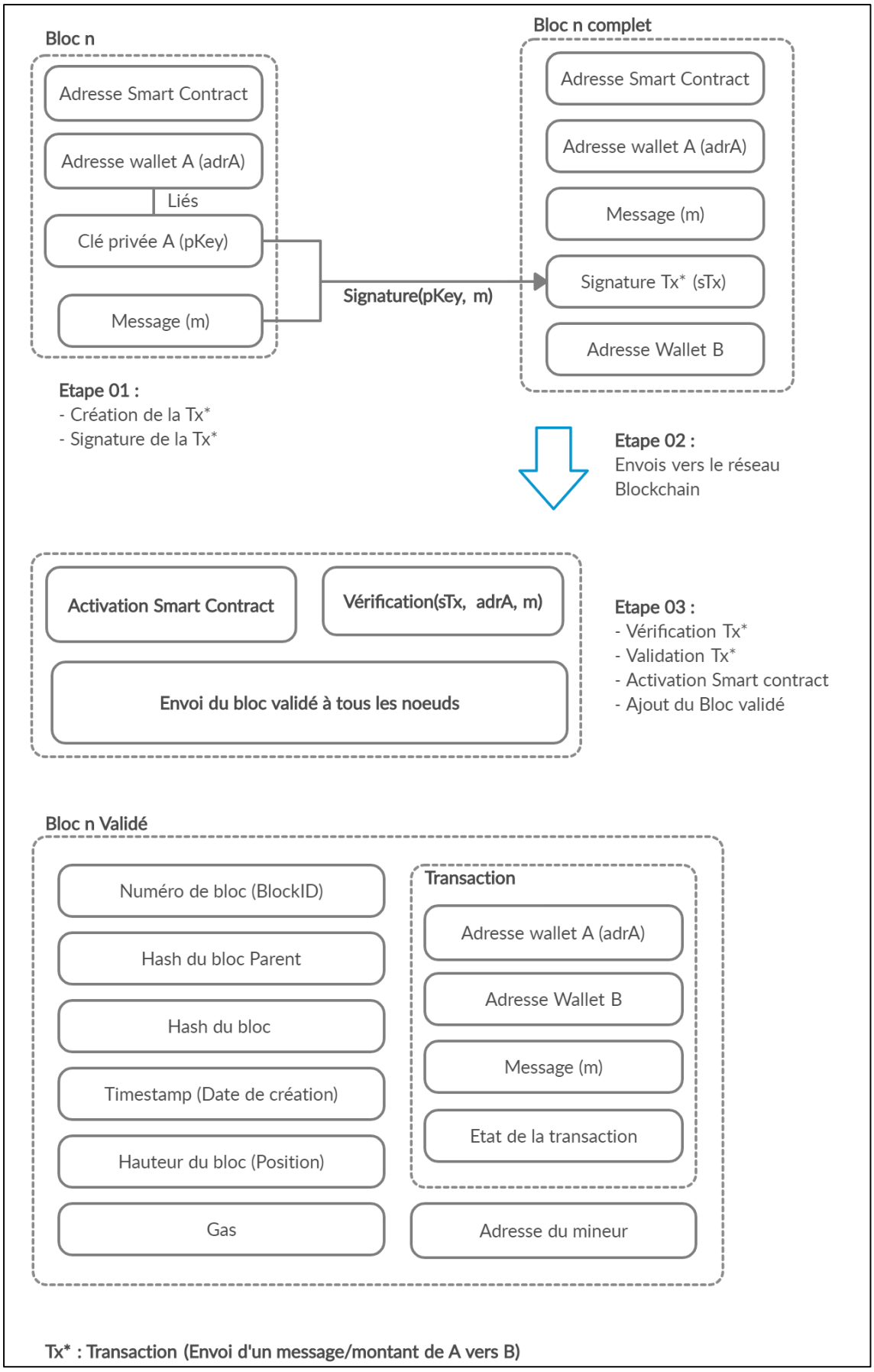


Figure 3.8 : Processus de création et validation d'un Bloc dans le réseau

Schéma descriptif de la validation de blocs [20]

Comme monté dans le schéma ci-dessus (Figure 3.9.1), le bloc est validé via le processus du proof-of-work expliqué déjà dans le chapitre 1. Le hash du bloc est soumis sous forme de puzzle que les nœuds du réseau doivent élucider, puis le nœud qui trouve le bon hash et donc valide tous les aspects de la transaction gagne une récompense (sous forme de monnaie virtuelle), ce qui encourage les nœuds du réseau à participer massivement à la vérification des blocs et leurs validations

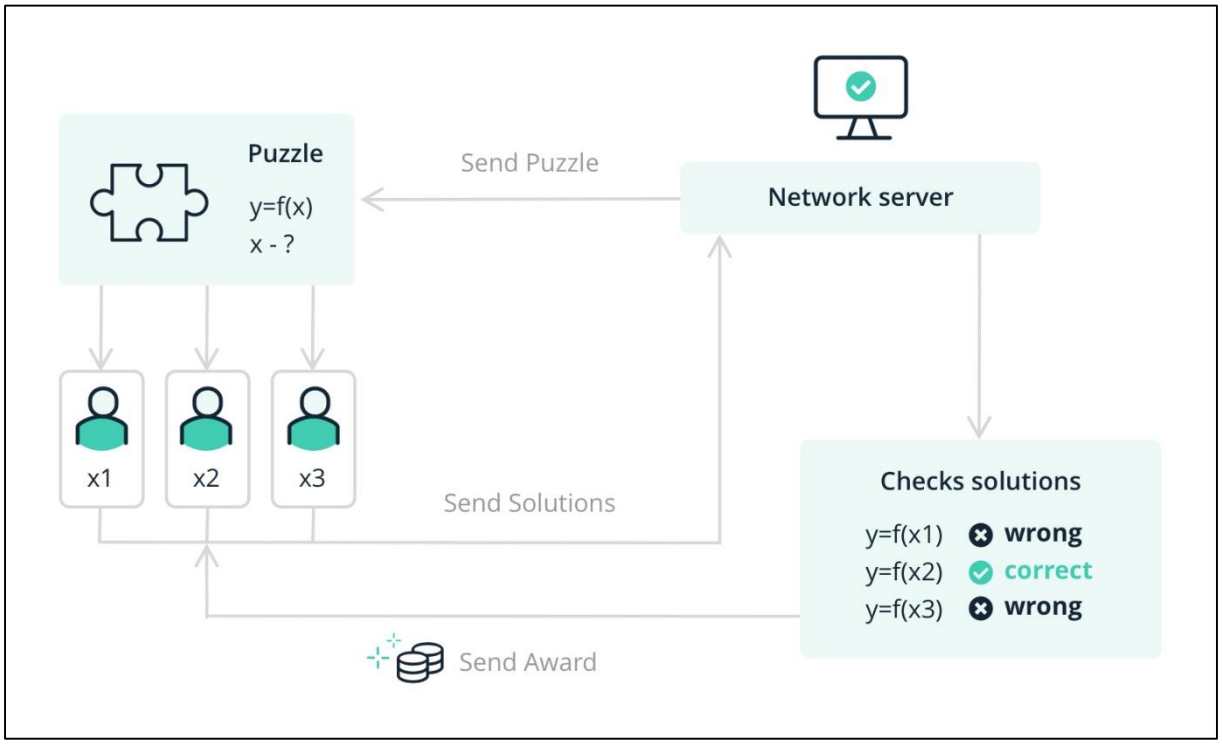


Figure 3.9.1 : Schéma illustrant le processus de proof-of-work

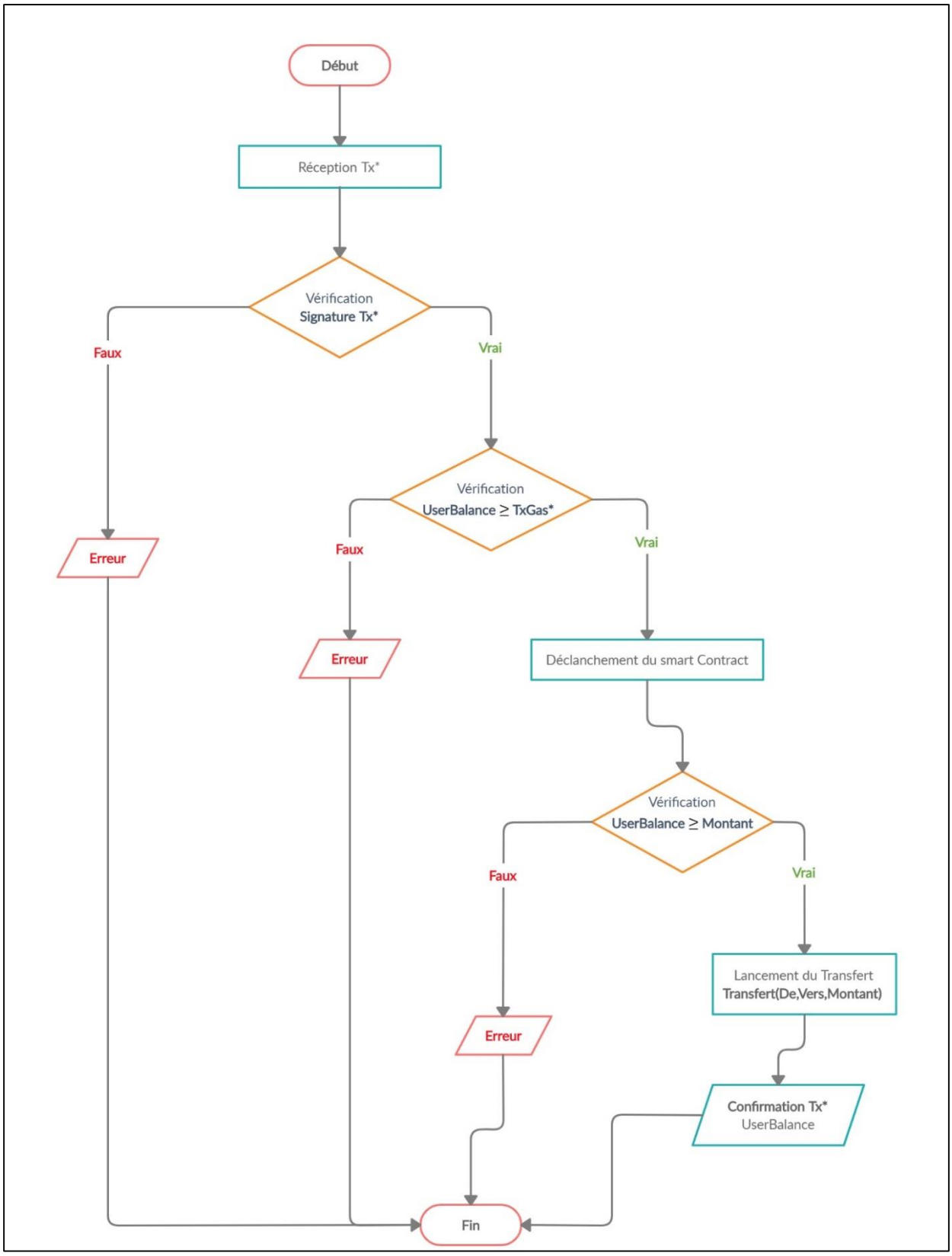


Figure 3.9.2 : Organigramme de l'algorithme du smart contract

11. Conclusion

Nous avons vu dans ce chapitre l'architecture et les différents composants de notre système ainsi que l'organisation de ces derniers et leurs interactions et fonctionnements ; après cela nous allons à présent passer la phase de réalisation et déploiement de cette application décentralisée.

Chapitre 4
Réalisation et déploiement du système

Chapitre 4 : Réalisation et déploiement du système

1. Introduction

Dans ce chapitre nous allons en premier lieu définir tous les langages de programmation, bibliothèques, frameworks et outils qu'on a utilisés pour créer un système de paiement mobile complet basé sur la blockchain.

Dans un premier temps nous allons voir le coté client (client-side) qui sera composé d'une application mobile suivi par le coté réseau qui sera basé sur la technologie blochchain.

Ensuite nous allons parcourir en détail les différentes parties de notre implémentation du réseau

Ainsi que son architecture globale

2. Langages, outils et technologies utilisées

2.1. Coté utilisateur (Client-Side)

Javascript [23]

JavaScript (qui est souvent abrégé en « JS ») est un langage de script léger, principalement utilisé au niveau des pages web, il est aussi utilisé dans de nombreux environnements extérieurs aux navigateurs web tels que Node.JS, Apache CouchDB, Electron JS, React JS et pleins d'autres. Le code JavaScript est compilé à la volée. Ce langage orienté objet utilise le concept de prototype et dispose d'un typage faible et dynamique qui permet de programmer suivant plusieurs paradigmes de programmation : fonctionnelle, impérative et orientée objet.

Node JS [24]

Node.js est une plateforme de développement open-source écrite en JavaScript. Elle est orientée vers les applications réseau événementielles hautement concurrentes ainsi que les applications demandant une importante extensibilité.

Elle utilise la machine virtuelle V8 (un moteur JavaScript open-source développé par le projet Chromium pour les navigateurs Web Google Chrome et Chromium).V8 compile directement le code JavaScript en code machine natif avant de l'exécuter, au lieu d'utiliser des techniques plus traditionnelles telles que l'interprétation du bytecode ou la compilation du programme complet en code machine et l'exécution à partir d'un système

de fichiers. Le code compilé est en outre optimisé (et ré-optimisé) dynamiquement au moment de l'exécution, en fonction du profil d'exécution du code), la librairie libuv pour sa boucle d'évènements, et implémente sous licence MIT les spécifications CommonJS.

Parmi les modules natifs de Node.js, on retrouve HTTP qui permet le développement de serveur HTTP. Il est donc possible de se passer de serveurs web tels que Nginx ou Apache lors du déploiement de sites et d'applications web développés avec Node.js.

Concrètement, Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur.

Node package manager (NPM) [25]

Node package manager (souvent abrévié en « NPM ») est le gestionnaire de paquets officiel de Node.JS. Il fait partie de l'environnement et est donc automatiquement installé par défaut. NPM fonctionne avec un terminal et gère les dépendances pour une application. Il permet également d'installer des applications Node.JS disponibles sur le dépôt NPM.

React JS [26]

React (aussi appelé React.js ou ReactJS) est une bibliothèque JavaScript open-source développée et maintenue par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.

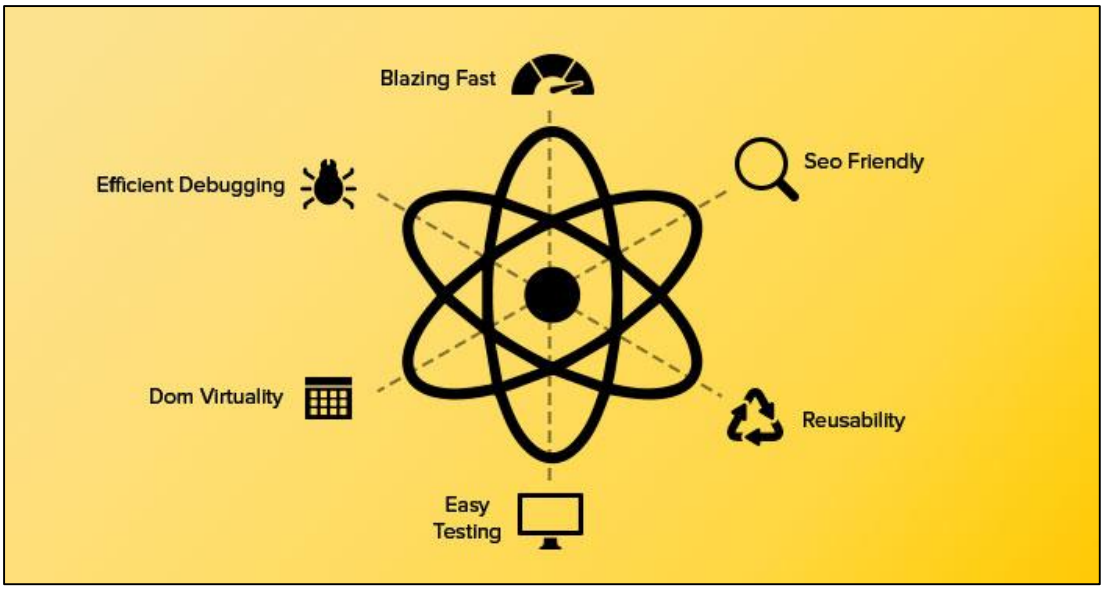


Figure 4.1 : Caractéristiques principales de React JS.

React est une bibliothèque qui ne gère que l'interface de l'application, considéré comme la vue dans le modèle MVC. Elle peut ainsi être utilisée avec une autre bibliothèque ou un framework MVC comme AngularJS. La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité.

React Native [27]

React Native est un framework d'applications mobiles open source créé par Facebook. Il est utilisé pour développer des applications pour Android, iOS et UWP (Universal Windows Platform) en permettant aux développeurs d'utiliser React avec les fonctionnalités natives de ces plateformes.

Les principes de fonctionnement de React Native sont pratiquement identiques à ceux de React, à la différence que React Native ne manipule pas le DOM via le DOM virtuel. Il s'exécute dans un processus en arrière - plan (qui interprète le code JavaScript écrit par les développeurs) directement sur le terminal et communique avec la plate-forme native via une passerelle de sérialisation, asynchrone et par lots.

React Native n'utilise pas HTML. Au lieu de cela, les messages du thread JavaScript sont utilisés pour manipuler des vues natives.

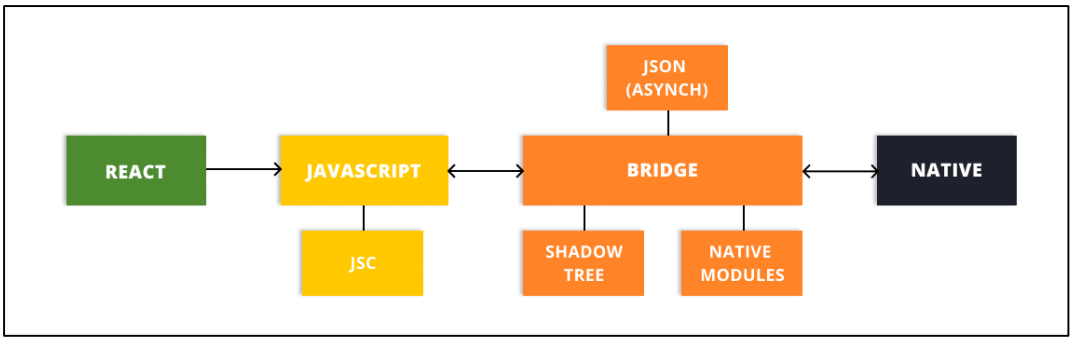


Figure 4.2 : Principe de fonctionnement de React Native.

Redux [28]

Redux est une bibliothèque open-source JavaScript de gestion d'état pour applications web. Elle est plus couramment utilisée avec des bibliothèques comme React ou Angular pour la construction d'interfaces utilisateur. Semblable à (et inspirée de) l'architecture Flux, elle a été créée par Dan Abramov et Andrew Clark.

Fonctionnement de Redux

Redux se compose de trois composants fondamentaux, l'action, le reducer et le store.

- **L'action :** Comme son nom l'indique, définit l'action que l'utilisateur fait, celle-ci déclenche une fonction qui est gérée elle aussi par Redux et renvois le résultat souhaité à l'utilisateur.
- **Le store :** Représente une base de données locale (stockage en cache) de tous les états de l'application, ainsi que des données contenues dans les états
- **Le reducer :** Regroupe toutes les fonctions qui peuvent être déclenchées par l'utilisateur et qui sont déclenchées par les actions.

Exemple : l'utilisateur appuie sur le bouton « rafraichir la page », ce qui déclenche l'action rafraichir au niveau de Redux et celui-ci appelle la fonction « actualiser » qui va s'occuper de récupérer les nouvelles données depuis la source. Enfin, Redux stocke les nouvelles données dans le store en tant qu'état et seront partagé à tous les composants de l'application qui les manipulent et qui s'occuperont de les afficher a l'utilisateur.

Redux Thunk [29]

Redux Thunk est une bibliothèque open-source JavaScript complémentaire pour Redux. Elle permet de surcharger les fonctions standards que Redux peut gérer. (Un thunk est une autre appellation pour la fonction).

Redux Persist

Redux Persist est une bibliothèque open-source JavaScript complémentaire pour Redux et Redux Thunk. Elle permet de stocker les états de l'application dans le cache de l'appareil.

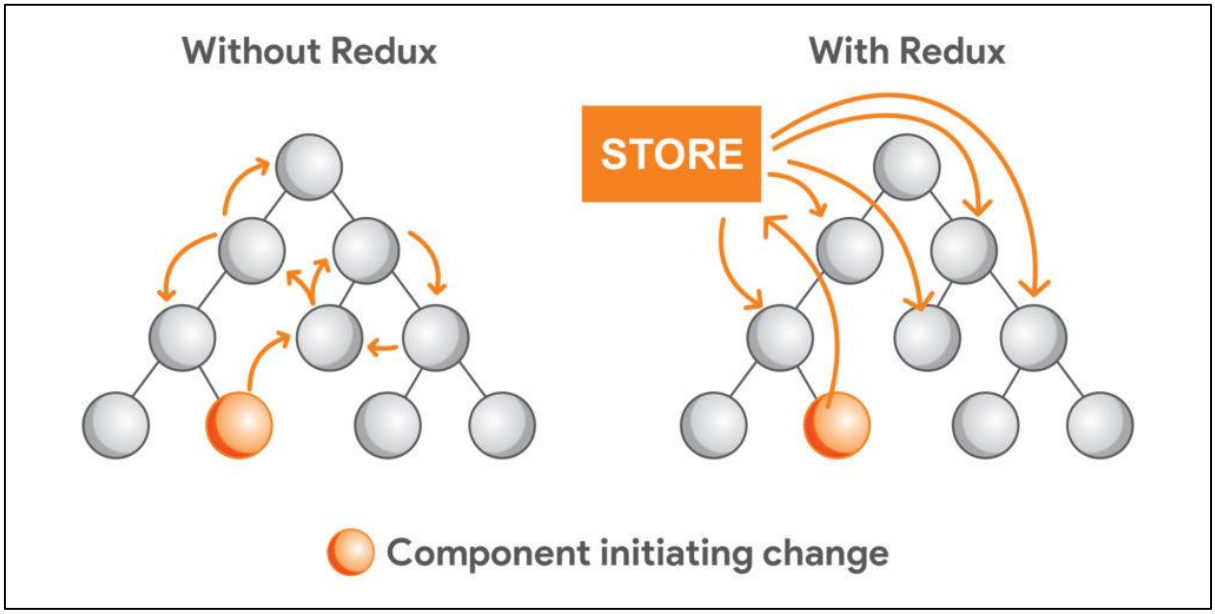


Figure 4.3 : Description du fonctionnement de redux au sein d'une application ReactJS

web3.JS [34]

web3.JS ou API JavaScript Ethereum est une collection de bibliothèques qui nous permettent d'interagir avec les nœuds d'une blockchain Ethereum locale ou distante à l'aide de HTTP, IPC ou WebSocket.

Web3 Utilise le protocole de communication JSON-RPC qui est un protocole d'appel de procédure à distance codé en JSON. Il est similaire au protocole XML-RPC, ne définissant que quelques types de données et commandes. JSON-RPC permet les notifications (données envoyées au serveur qui ne nécessitent pas de réponse) et l'envoi de plusieurs appels au serveur auxquels il est possible de répondre de manière asynchrone.

3. Coté Réseau et Blockchain (Network-Side)

3.1. Solidity [30]

Solidity est un langage de programmation orienté objet dédié à l'écriture de contrats intelligents. Il est utilisé pour implémenter des smart-contrat sur diverses blockchains, notamment Ethereum. Il a été développé par Christian Reitwiessner, Alex Beregszaszi, Yoichi Hirai et plusieurs anciens contributeurs principaux d'Ethereum pour permettre l'écriture de contrats intelligents sur des plateformes de blockchain telles qu'Ethereum.

Grâce à Solidity, les développeurs sont en mesure d'écrire des applications implémentant une logique commerciale s'exécutant de manière autonome au travers des contrats intelligents, laissant une trace de transactions non répudiables et faisant autorité. Écrire des contrats intelligents dans des langages spécifiques aux contrats intelligents tels que Solidity est considéré comme facile (en apparence pour ceux qui ont déjà des compétences en programmation).

3.2. Remix IDE [31]

Remix est un puissant outil open source qui aide à la rédaction des contrats Solidity directement à partir du navigateur. Écrit en JavaScript, Remix prend en charge à la fois l'utilisation dans le navigateur et localement.

Remix prend également en charge les tests, le débogage et le déploiement de contrats intelligents directement vers un réseau blockchain.

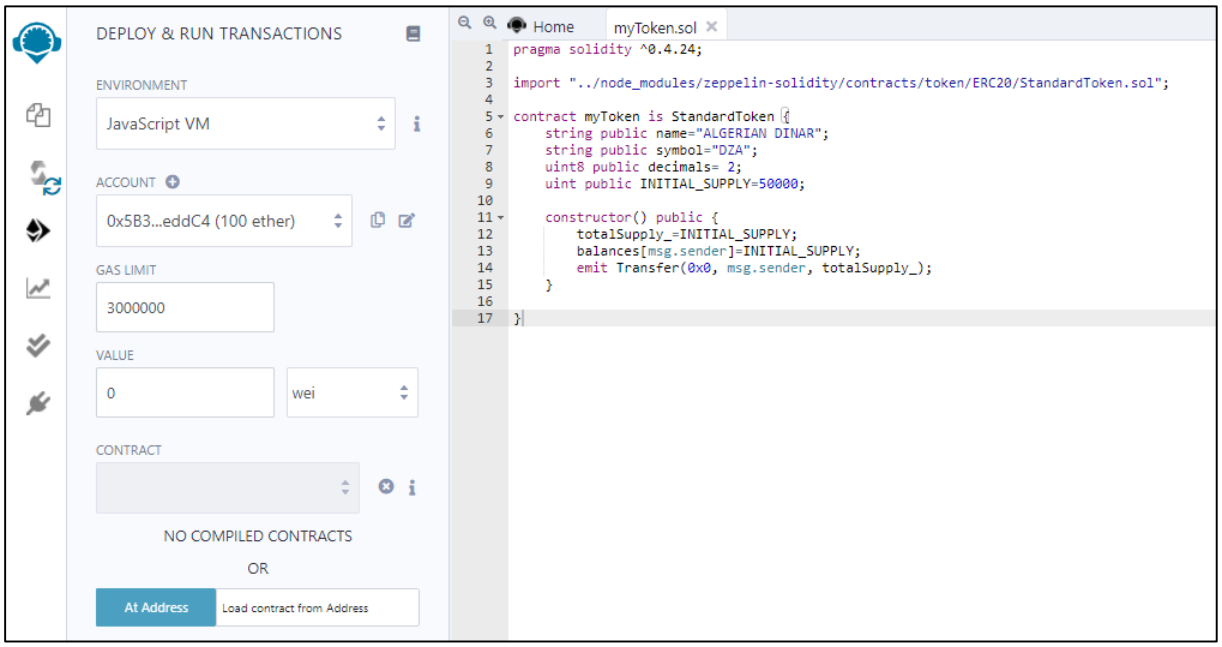


Figure 4.4 : Capture de l'interface de déploiement de contrats de Remix IDE

3.3. Ethereum Blockchain [32]

Ethereum est une blockchain open source décentralisée dotée d'une fonctionnalité de contrat intelligent. C'est la deuxième plus grande crypto-monnaie en termes de capitalisation boursière, derrière Bitcoin.

Ethereum fournit une machine virtuelle répliquée décentralisée, appelée Ethereum Virtual Machine (EVM), qui peut exécuter des scripts à l'aide d'un réseau international de nœuds publics. Le jeu d'instructions de la machine virtuelle est Turing-complet, contrairement à Bitcoin.

Le gas fait référence aux frais nécessaires pour mener à bien une transaction sur Ethereum et est payé dans la devise native d'Ethereum, Ether (ETH). Les prix du gas sont indiqués en Gwei, qui est une dénomination de l'ETH. Chaque Gwei est égal à 0,00000000001 ETH. Ce mécanisme de frais est utilisé pour atténuer le spam de transaction et allouer des ressources sur le réseau.

Ethereum a été proposé fin 2013 par Vitalik Buterin, un chercheur et programmeur de crypto-monnaie. Le développement a été financé par un crowdsale en ligne qui a eu lieu entre juillet et août 2014. Le système a ensuite été mis en service le 30 juillet 2015.

En 2016, à la suite de l'exploitation d'une faille dans le logiciel de contrat intelligent du projet DAO et du vol subséquent de 50 millions de dollars d'Ether, Ethereum a été divisé en deux blockchains distinctes. La nouvelle version séparée est devenue Ethereum (ETH) avec le vol inversé, et la chaîne originale a continué comme Ethereum Classic (ETC).

Ethereum développe et prévoit actuellement de mettre en œuvre une série de mises à niveau appelée Ethereum 2.0.

Les spécifications actuelles d'Ethereum 2.0 incluent une transition vers la preuve de participation et une augmentation du débit des transactions à l'aide de la technologie de partitionnement.

3.4. Ethereum Virtual Machine [33]

La machine virtuelle Ethereum (EVM) est un système dit “Turing Complet” (en hommage à Alan Turing, créateur de la machine et du test de son nom). Cela signifie que le dit système possède la puissance de calcul de la machine de Turing, et est capable de réaliser des calculs lambda, fonctions récursives, comparaison, lectures/écritures ... Il permet déployer des smart-contracts et de garantir leur immuabilité et leur fonctionnement sur la blockchain Ethereum en échange de frais de déploiements, et d’utilisations que l’on appelle gas.

L’EVM est intégrée dans chaque nœud Ethereum et responsable de l'exécution du bytecode du contrat. Les contrats sont généralement écrits dans des langages de niveau supérieur, comme Solidity, puis compilés en bytecode EVM.

Cela signifie que le code machine est complètement isolé du réseau, du système de fichiers ou de tout processus de l'ordinateur hôte. Chaque nœud du réseau Ethereum exécute une instance EVM qui leur permet de s'entendre sur l'exécution des mêmes instructions.

Les machines virtuelles Ethereum ont été implémentées avec succès dans divers langages de programmation, notamment C ++, Java, JavaScript, Python, Ruby et bien d'autres.

L'EVM est essentiel au protocole Ethereum et joue un rôle déterminant dans le moteur de consensus du système Ethereum. Il permet à n'importe qui d'exécuter du code dans un écosystème sans confiance dans lequel le résultat d'une exécution peut être garanti et est entièrement déterministe (c'est-à-dire) l'exécution de contrats intelligents.

Pour chaque instruction implémentée sur l'EVM, un système qui assure le suivi du coût d'exécution, attribue à l'instruction un coût associé en unités de gas, Lorsqu'un utilisateur souhaite lancer une exécution, il réserve de l'éther, qu'il est prêt à payer pour ce coût de gas.

En utilisant le mécanisme gas, deux problèmes majeurs sont résolus : Un validateur est assuré de recevoir le montant prépayé initial, même si l'exécution échoue. Une exécution ne peut pas durer plus longtemps que le montant prépayé le permettrait. Au lieu de boucler indéfiniment, l'exécution s'exécuterait jusqu'à ce qu'elle soit à court de gas.

Lorsqu'une transaction est envoyée au réseau, les validateurs peuvent prendre la transaction en exécutant le code associé.

Le validateur s'assurera que :

- L'expéditeur dispose de suffisamment de fonds pour payer l'exécution de la transaction.
- Toutes les informations sur la transaction sont valides.
- L'EVM n'a rencontré aucune exception lors de l'exécution.

3.5. Microsoft Azure Blockchain service [6]

Le service blockhain de Microsoft Azure va nous permettre de Créer, gérer et développer des réseaux blockchain à grande échelle. Le service Azure Blockchain simplifie la formation, la gestion et la gouvernance des réseaux blockchain de consortium afin de nous permettre de nous concentrer sur la logique métier et le développement de notre application.

3.6. Firebase[35] [36]

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification mobile (2FA), sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel.

Le service est racheté par Google en octobre 2014. Il appartient aujourd'hui à la maison mère de Google : Alphabet.

Toute l'implémentation et la gestion serveur de Firebase est à la charge exclusive de la société Alphabet. Les applications qui utilisent Firebase intègrent une bibliothèque qui permet les diverses interactions possibles.

Firebase Authentification

Firebase Authentication fournit des services de backend, des SDK faciles à utiliser et des bibliothèques d'interface utilisateur prêtes à l'emploi pour authentifier les utilisateurs auprès de notre application. Il prend en charge l'authentification à l'aide de mots de passe, de numéros de téléphone (2 Factor Auhentification ou 2FA), de fournisseurs d'identité fédérés populaires tels que Google, Facebook et Twitter, etc.

Cloud Firestore

Cloud Firestore est une base de données flexible et évolutive pour le développement mobile, Web et serveur de Firebase et de Google Cloud Platform. Ce service maintient les données synchronisées entre les applications clientes via des écouteurs en temps réel et offre une prise en charge hors ligne pour le mobile et le Web afin de pouvoir créer des applications réactives qui fonctionnent indépendamment de la latence du réseau ou de la connectivité Internet.

4. Application client

4.1. Authentification et vérification par SMS (2FA)

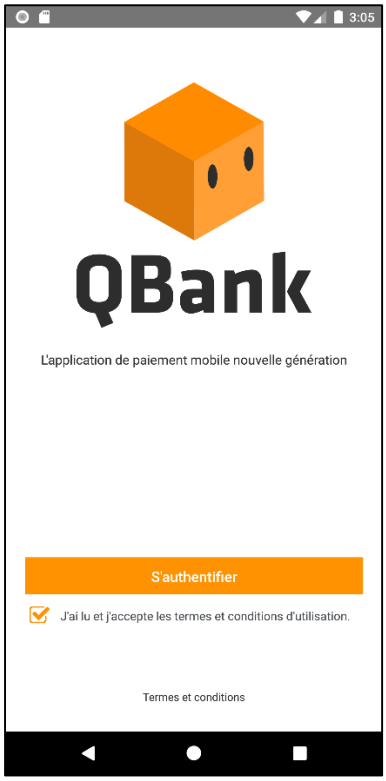


Figure 4.5 : Vue Accueil

Figure 4.6 : Vue Authentification

Figure 4.7 : Vérification code SMS

Les vues ci-dessous représentent la partie authentification de l'application, ou l'utilisateur est invité à s'authentifier à l'aide de son numéro de téléphone et un code de vérification qui lui sera envoyé par SMS. Si l'utilisateur s'authentifie correctement, il sera automatiquement redirigé vers la page d'accueil.

En cas de première authentification, l'utilisateur sera redirigé vers la page profile pour remplir ses informations de base.

4.2. Application principale

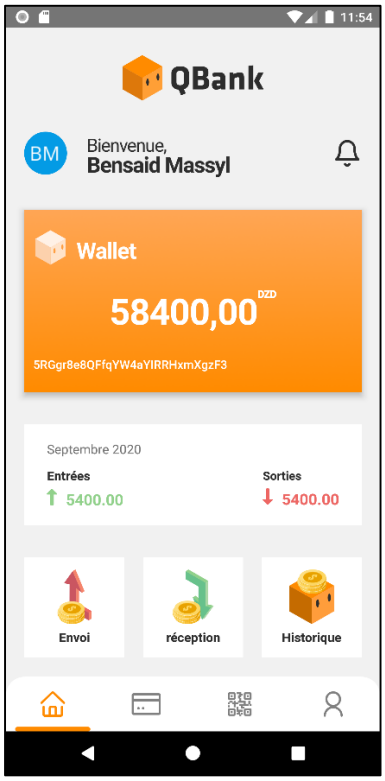


Figure 4.8 : Vue d'accueil

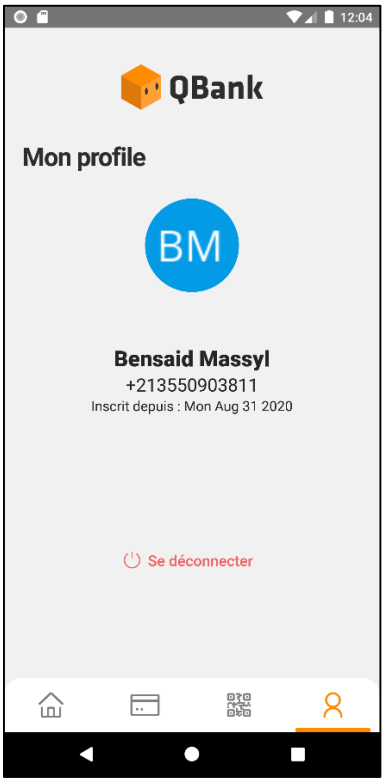


Figure 4.9 : Vue Profile

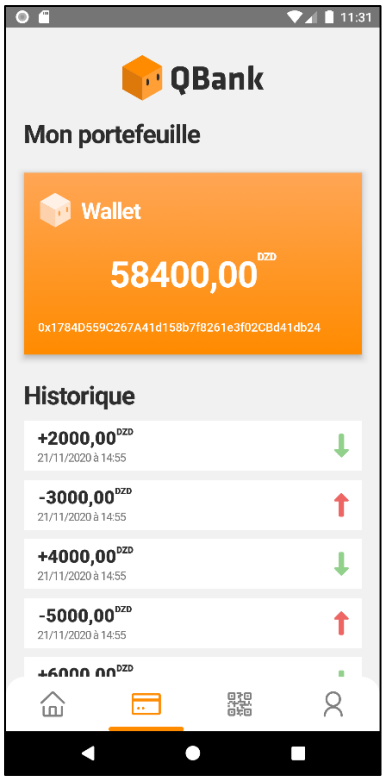


Figure 4.10 : Vue Historique

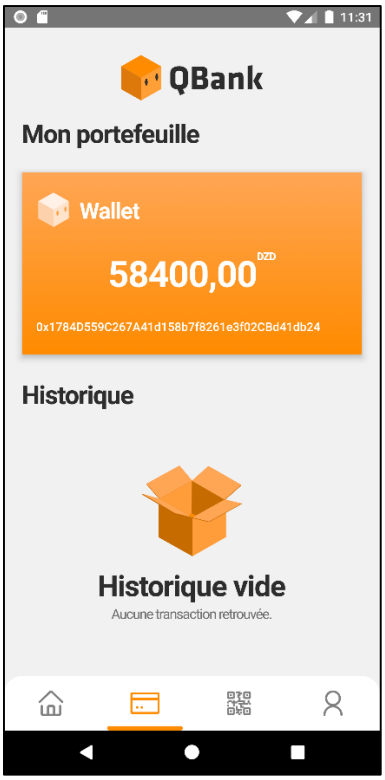


Figure 4.11 Vue historique vide

Après l'authentification de l'utilisateur, il sera redirigé vers la page d'accueil ou il pourra trouver un résumé de son compte et accéder à son historique de transferts. A partir d'ici l'utilisateur aura accès à l'envoi et à la réception d'argent ainsi qu'à son profile ou il pourra retrouver tous les détails de son compte.

4.3. Envoi et réception :

Figure 4.12 : Vue Code QR

Figure 4.13 Vue Transaction

Les deux vues ci-dessus représentent les fonctions d'envoi et de réception, le code QR au niveau de la figure 4.12 permet le démarrage d'une transaction rapide. Au niveau de la figure 4.13 On peut créer une nouvelle transaction en saisissant le numéro de téléphone du destinataire et en spécifiant le montant à transférer.

Conclusion générale et perspectives

A travers ce projet, nous avons vu comment on peut implémenter un réseau blockchain qui peut assurer la sécurité des transactions monétaires, Ainsi qu'une application mobile sécurisée pour gérer les transferts en ligne, le tout sans organe central car toutes les transactions sont enregistrées sur un registre global et totalement distribué et inviolable. On assure donc l'authenticité de toutes les transactions ainsi que leurs non-répudiations.

La blockchain est une technologie nouvelle et innovante, qui change radicalement la façon dont les données sont sécurisées. Grâce à cette technologie on apporte un niveau de sécurité plus élevé ainsi qu'une non-répudiation totale grâce à son aspect immutable et on est assurés que les données sont infalsifiables. A travers notre travail nous avons réalisé une dApp capable de gérer de manière sécurisée diverses transactions monétaires mais cette technologie ne se résume pas qu'à ça, les dApps peuvent s'implémenter dans divers autres domaines et peuvent aider à sécuriser dans plusieurs contextes différents, Par exemple, une dApp qui gère les votes des utilisateurs et qui ne garantit qu'aucun trafic au niveau des votes comptabilisé et des chiffres ait eu lieu.

Les dApps et la blockchain en général touchent à tous les domaines et peuvent aider grâce à un ou plusieurs de leurs aspects à améliorer ces domaines et à augmenter leurs efficacités.

Bibliographie et références

[1] “Architectures Réparties en JAVA” 3e édition, Annick Fron, Dunod Editions, 2015

[2] “Nouvelles technologies réseau : les réseaux peer-to-peer, Nathalie Budan, Benoit Tedeschi, Stéphane Vaubourg

[3] The NIST Definition of Cloud Computing (Special Publication 800-145)

[4] Cloud computing defined: Characteristics & service levels - IBM Bluemix Cloud

[5] “Architectural Styles and the Design of Network-based Software architectures”, Dr. Roy Thomas Fielding PhD in Information and computer Science.

[6] “Blockchain technology and Applications” Microsoft Azure “https://azure.microsoft.com/en-us/solutions/blockchain/ 20147

[7] NIST’s Guide to secure Web Services (Special Publication 800-95)

[8] “Cryptographic Standards and Guidelines” Dr. Lily Chen, National Institute of Standards and Technology, SP 800-175A/ SP 800-175B 2019

[9] “Bitcoin peer-to-peer electronic cash system” Satoshi Nakamoto Bitcoin Whitepaper 2009

[10] “Blockchain Technology Overview” Dylan J. Yaga, Peter M. Mell, Nik Roby, Karen Scarfone National Institute of Standards and Technology Internal Report (NISTIR) – 8202 Octobre 2018

[11] “Bitcoin’s Academic Pedigree” Arvind Narayanan, Jeremy Clark 2017

[12] “Blockchain the invisible technology that’s changing the world” PCMAG “www.pcmag.com/news/blockchain-the-invisible-technology-thats-changing-the-world”

[13] “Blockchain for Industrial Applications Community of Interest” National Institute of Standards and Technology, November 15, 2019

[15] “A Taxonomic Approach to Understanding Emerging Blockchain Identity Management Systems” National Institute of Standards and Technology NISTIR 8202 - 14 Janvier 2020

[16] “The difference between app coins and protocol tokens” blog.0xproject.com/the-difference-between-app-coins-and-protocol-tokens-7281a428348c

[17] “What Is Blockchain Technology” CoinDesk “www.coindesk.com/learn/blockchain-101/what-is-blockchain-technology”

[18] “Smart Contracts: Building Blocks for Digital Markets” Nick Szabo 1996

[19] “Token Economy” Shermin Voshmgir 2019

[20] “The future will be decentralized” Charles Hoskinson TEDx Conference 2014

[21] “Mastering Ethereum” Andreas M. Antonopoulos and Dr. Gavin Wood (O’Reilly) 2019

[22] “Enhanced distributed ledger technology” National Institute of Standards and Technology October 19, 2020

[23] Javascript Mozilla Developer Network, “https://developer.mozilla.org/en-US/docs/Web/JavaScript”.

[24] Node JS Official Docs, “https://nodejs.org/en/docs/”.

[25] NPMJS Official website, “https://www.npmjs.com/”.

[26] React Js Official Docs, “https://fr.reactjs.org/docs/getting-started.html”.

[27] React Native Official Docs, “https://reactnative.dev/”.

[28] Redux Official Docs, “https://redux.js.org/”.

[29] Redux thunk Github repo, “https://github.com/reduxjs/redux-thunk”.

[30] Ethereum Solidity Github repo, “https://github.com/ethereum/solidity”.

[31] Remix - Ethereum IDE & community “remix-project.org”

[32] Ethereum Blockchain Network “ethereum.org”

[33] Ethereum evm Illustrated, Takenobu Tani “https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf “, Mars 2018

[34] Web3 JS “https://web3js.readthedocs.io/en/v1.3.0/“

[35] Google cloud platform official documentation, “https://cloud.google.com/docs/”.

[36] Google Firebase official documentation, “https://firebase.google.com/docs”.