UNIVERSITÉ JOSEPH KI-ZERBO (UJKZ)

****

INSTITUT BURKINABÈ DES ARTS ET MÉTIERS (IBAM)

MÉMOIRE POUR L'OBTENTION DU MASTER

Option : **Ingénierie des Systèmes d'information en Entreprise (ISIE)**

THÈME :

Authentification de documents administratifs à l'aide de la blockchain

Présenté par : **M. HIEN Zilèdem Pierre Canisius**
Période de stage : 01/01/2025 au 30/05/2025

Sous l'encadrement et la supervision de : **Dr. Yaya TRAORE**, *Maître de Conférences en informatique*.

Année universitaire : 2023-2024

DÉDICACE

A ma fille, W. Candice Urielle !

REMERCIEMENTS

Nous tenons tout d'abord à rendre grâce à Dieu Le Tout Puissant, par qui nous vivons.

Au cours de ce cycle de Master, nous avons eu le privilège de collaborer avec des personnes de marques qui nous ont fourni tout le nécessaire afin que nous parvenions à ce résultat. C'est donc le lieu pour nous, à travers ces lignes, de leur traduire notre profonde gratitude.

Nos distincts remerciements vont particulièrement à l'endroit de :

- **Dr. Yaya TRAORE**, notre Directeur de mémoire, qui a bien voulu nous encadrer sans hésitation, et pour sa disponibilité malgré un calendrier chargé. Ses critiques constructives et ses partages d'expériences nous ont été d'une très grande utilité.
- **Pr. Sadouanouan MALO**, pour ses conseils bien avisés, ses multiples encouragements et accompagnements.
- **Monsieur le Directeur de l'IBAM, le corps professoral et tout le personnel de l'IBAM** pour la formation reçue et leur accompagnement.
- **notre famille, nos collègues et camarades** pour leurs soutiens et encouragements.
- **toutes les personnes dont les noms n'ont pu être cités.**

RÉSUMÉ

Les documents administratifs ont toujours joué un rôle central non seulement dans le fonctionnement de l'Administration mais aussi dans la relation entre l'Administration et ses usagers/clients. Le document administratif sert entre autres à la traçabilité et constitue un support de décision, de communication, et de droit. Cela signifie que tout document administratif doit être fiable et d'origine incontestable. Cependant, il est de plus en plus fréquent d'avoir à faire à des documents administratifs numériques falsifiés ou d'origine douteuse. Ce phénomène crée une crise de confiance entre l'usager/client et l'administration. Il fait perdre du temps énorme aux structures administratives qui restent en veille pour démentir d'éventuels faux documents à elles attribués, au lieu de se consacrer exclusivement aux services administratifs règlementaires. En plus de la perte de temps, des ressources financières et humaines sont parfois utilisées par des structures pour procéder à des vérifications ou à des opérations d'authentification de certains documents administratifs auxquels elles font face.

Pour résoudre cette problématique, nous proposons une application décentralisée (DApp) hybride dénommée ADOBLOCK qui interagie avec la blockchain publique Ethereum 2.0 (utilisant la preuve d'enjeu – PoS – comme protocole de consensus) à travers un contrat intelligent que nous avons implémenté. Nous nous sommes intéressés particulièrement à la technologie blockchain pour ses propriétés de non-répudiation, de transparence et de stockage décentralisé. Avec ADOBLOCK, il est possible de vérifier l'authenticité d'un document administratif (PDF ou Word à téléverser) en temps réel.

Dans notre approche, un acteur d'une administration peut préalablement stocker un document administratif sur la blockchain. Précisément, il s'agit de l'empreinte numérique (SHA-256) signée numériquement à l'aide de la paire de clés cryptographiques ECDSA de l'acteur. Ces clés sont obtenues à travers ADOBLOCK et la signature numérique est calculée sur la base de la courbe elliptique secp256r1 (P-256). Nous combinons à ce procédé, la technique d'horodatage électronique. Ainsi, la DApp permet de s'assurer de l'intégrité du contenu du document et de la fiabilité de la signature associée audit document.

Techniquement, ADOBLOCK est conçu avec les Frameworks Angular 15, Spring-boot 3.4.1, MetaMask, Truffle et Ganache 2.7.1, les langages HTML5, TypeScript, JavaScript, Java 17, Solidity ^0.8.28 et les technologies API REST, Hardhat ^2.24.0, Ethers.js ^6.14.0, etc.

ABSTRACT

Administrative documents have always played a central role, not only in the functioning of the Administration, but also in the relationship between the Administration and its users/customers. Administrative documents are used, among other things, for traceability, and as a basis for decision-making, communication and law. This means that all administrative documents must be reliable and of indisputable origin. However, it is becoming increasingly common to have to deal with digital administrative documents that have been falsified or are of dubious origin. This phenomenon creates a crisis of confidence between the user/customer and the administration. It wastes an enormous amount of time for structures, which remain on standby to disprove any false documents attributed to them, instead of devoting themselves exclusively to statutory administrative services. In addition to wasting time, financial and human resources are sometimes used by structures to carry out verifications or authentication operations on certain administrative documents they are faced with.

To solve this problem, we propose a hybrid decentralized application (DApp) called ADOBLOCK that interacts with the Ethereum 2.0 public blockchain (using proof-of-stake as a consensus protocol) through a smart contract that we have implemented. We were particularly interested in blockchain technology for its properties of non-repudiation, transparency and decentralized storage. With ADOBLOCK, it is possible to verify the authenticity of an administrative document (PDF or Word to be uploaded) in real time.

In our approach, an actor from an administration can first store an administrative document on the blockchain. Specifically, this is the hash (SHA-256) digitally signed using the actor's ECDSA cryptographics keys pair. These keys are obtained through ADOBLOCK and the digital signature is calculated on the basis of the secp256r1 (P-256) elliptic curve. We combine this process with electronic time-stamping. In this way, the DApp ensures the integrity of the document content and the reliability of the signature associated with the document.

Technically, ADOBLOCK is designed with the Angular 15, Spring-boot 3.4.1, MetaMask, Truffle and Ganache 2.7.1 frameworks, HTML5, TypeScript, JavaScript, Java 17, Solidity ^0.8.28 languages and API REST, Hardhat ^2.24.0, Ethers.js ^6.14.0 technologies, etc.

TABLES DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES SIGLES ET ABREVIATIONS

ABI	Application Binary Interface (interface binaire d'application)
API	Application Programming Interface (interface de programmation d'application)
CSS	Cascading Style Sheets ou Feuilles de Style en Cascade
DApp	Decentralized Application ou application décentralisée
HTML	HyperText Markup Language ou Langage de balises pour l'hypertexte
HTTP	Hypertext Transfer Protocol ou Protocole de Transfert Hypertexte
IDE	Integrated Development Environment (environnement de développement intégré)
JSON	JavaScript Object Notation
P2P	Peer-to-Peer
PoA	Proof of Autority ou Preuve d'Autorité
PoS	Proof of Stake ou Preuve d'Enjeu
PoW	Proof of Work ou Preuve de Travail
RPC	Remote Procedure Call ou Protocole d'appel de procédure distante
SHA-256	Secure Hash Algorithm (algorithme de hachage sécurisé) 256 bits

CHAPITRE 1 :

INTRODUCTION GENERALE

CHAPITRE 1 : INTRODUCTION GENERALE

La blockchain est une révolution technologique en plein essor au cours de ces dernières années. Reconnue pour ses propriétés de non-répudiation, de transparence, et de stockage décentralisé, elle présente de nombreux avantages dans plusieurs aspects de la vie humaine, et s'avère un moyen efficace pour améliorer les services publics gouvernementaux en particulier. En effet, dans le domaine des services gouvernementaux, il est de nos jours très récurent de retrouver sur la place publique numérique, des projets de documents administratifs et même des documents administratifs falsifiés, causant ainsi beaucoup de dommages. Malheureusement au Burkina Faso, ce phénomène prend une allure inquiétante. De ce fait, et sachant les possibilités qu'offre la blockchain, nous pensons que l'utilisation de cette technologie dans les processus d'authentification de documents administratifs, pourrait apporter plus de facilité, de fiabilité et de sécurité.

Dans ce chapitre, nous présentons le contexte général dans lequel est né le sujet, et le problème que l'on se propose de résoudre. En plus d'énoncer des hypothèses de recherche, nous y présentons les objectifs et les résultats attendus du sujet. Ce chapitre fourni également une vue d'ensemble du déroulement de notre étude.

1.1. Contexte et justification

La fiabilité de certains documents administratifs est de plus en plus controversée. Cela pourrait être dû en partie, au développement et à l'exploitation malsaine des multiples outils basés par exemple sur l'Intelligence Artificielle (IA). En effet, ce phénomène est caractérisé par le fait que de nombreux cas réels de faux « documents administratifs » ont été retrouvés sur des espaces numériques publics, mettant en déroute bon nombre de citoyens. Parmi ces cas, la majorité a nécessité des démentis officiels venant des structures étatiques que nous qualifions de « victimes » de faux. Par exemple, nous avons constaté entre autres que :

- le 23 janvier 2025, le Ministère de l'Économie, des Finances et de la Prospective, à travers sa page Facebook, a alerté le public comme suit : « *Des avis au public, faussement attribués au ministère de l'Économie et des Finances faisant état de l'ouverture de sessions d'investissement sur les cryptomonnaies sont diffusés sur les réseaux, aux fins de spoliation*

des citoyens. Le ministère de l'Économie et des Finances tient à rassurer l'opinion qu'il n'est nullement associé à cette initiative qui n'est autre que de l'arnaque et invite les citoyens à la vigilance et à dénoncer les auteurs et les complices de telles pratiques auprès des autorités compétentes. » [22].

- le 16 août 2024, le Ministère de la Fonction Publique, du Travail et de la Protection Sociale publiait un démenti, signé DCRP/MFPTPS, sur sa page Facebook en ces termes « *Une loi portant statut général des agents publics et un projet de loi portant statut général des agents publics circulant sur les réseaux sociaux sont faux. Ces textes ne proviennent pas des services techniques du Ministère de la Fonction Publique, du Travail et de la Protection Sociale, ni du Gouvernement ou de l'Assemblée législative de Transition* » [12].
- le 18 janvier 2024, le Service de Communication et des Relations Publiques de la Direction Générale des Douanes a, de même, publié un démenti sur sa page Facebook comme suit : « *Le Service de Communication et des Relations Publiques de la Direction Générale des Douanes informe le public que les communiqués, ci-dessous, sur une supposée vente aux enchères de véhicules n'émanent nullement des services des Douanes* » [14].
- le 06 octobre 2023, le Ministère des Affaires Étrangères du Burkina Faso, à travers la DRCP/MAECR-BE, avait également publié sur sa page Facebook, ce qui suit : « *Depuis un certains temps un communiqué relatif à une bourse canadienne et impliquant le Ministère des Affaires Étrangères du Burkina Faso circule sur les réseaux sociaux. Le Ministère des Affaires Étrangères du Burkina Faso, apporte un démenti à ce communiqué qui est certainement l'œuvre d'individus mal intentionnés.* » [11].
- le 30 novembre 2021, l'Institut national de la statistique et de la démographie (INSD) avait aussi démenti sur sa page Facebook, un faux recrutement d'étudiants dont il serait l'auteur [13].

Le plus souvent, une copie de chaque document en question a été marquée d'insigne de faux par les structures « victimes », puis annexée aux différents démentis. De ce fait, ces documents pourraient être classés « administratifs », car par définition, peut être vu comme document administratif, tout acte produit ou reçu, dans le cadre de la mission de service public, par l'État, les collectivités territoriales ainsi que par les autres personnes de droit public ou les personnes de droit privé chargées d'une telle mission [7]. Il peut s'agir d'un communiqué, un certificat (de prise /cessation de service, administratif, …), un diplôme, un extrait d'acte de naissance, etc.

Ainsi, au regard de la fréquence progressive et élevée de la falsification et/ou fraude de ces types de document, les citoyens, et même des acteurs de l'Administration posent de nombreuses préoccupations sur l'authenticité, ou l'originalité de tel ou tel acte (document) administratif qui se présente à eux.

A l'image de l'IA, la Blockchain est une technologie novatrice en pleine croissance. C'est une technologie de registre numérique distribué qui s'applique dans divers domaines. En outre, la blockchain étant immuable, décentralisée et transparente, pourrait être une alternative aux préoccupations susmentionnées.

C'est pourquoi, dans ce contexte général, nous nous intéressons donc à l'utilisation de la technologie blockchain, notamment ses protocoles de consensus, de vérification et de validation, pour résoudre la question d'authentification des documents administratifs, afin de réduire les risques de fraudes et de falsifications de ces documents. D'où le thème du présent mémoire « ***Authentification de documents administratifs à l'aide de la blockchain*** » que nous nous proposons d'étudier.

1.2. Problématique et hypothèses

La falsification (ou fraude) documentaire est un problème d'actualité auquel est confrontée particulièrement l'administration publique. Afin de conserver leur image, d'assurer une bonne gouvernance et d'éviter l'usage du faux, les structures publiques et privées s'efforcent de vérifier elles-mêmes, manuellement les dossiers des usagers/clients. A défaut, elles délèguent et suivent une longue procédure de vérifications de ces dossiers par des tiers, moyennant des ressources (financières, humaines, temps, …) considérables. Elles sont également contraintes de rester plus ou moins en veille constante pour démentir d'éventuels faux documents à elles attribués. Cela diminue non seulement le temps consacré aux activités règlementaires des services administratifs, mais augmente les coûts de contrôle et les délais de prestations de services.

De ce fait, quel outil ou quelle technologie peut être mise en œuvre pour faire face à cette situation ? pour nous, la problématique qui se dégage, c'est comment la technologie blockchain peut-elle garantir l'authenticité des documents administratifs de façon plus sécurisée, transparente et efficace ? Autrement dit, comment la blockchain peut-elle aider les administrations à prévenir et/ou à détecter les tentatives de falsification de documents administratifs ? Cette technologie, peut-elle

permettre à un service public destinataire d'un document administratif de savoir si oui ou non, il s'agit bien d'un document authentique ?

Pour traiter cette problématique, nous posons les hypothèses suivantes :

Hypothèse 1 : l'intégration de la blockchain dans une solution d'authentification de documents permet de renforcer la sécurité et l'intégrité des documents administratifs.

Hypothèse 2 : la blockchain permet de vérifier l'authenticité en temps réel des documents administratifs.

Hypothèse 3 : l'utilisation de la blockchain dans le processus d'authentification de documents diminue les cas de falsifications et fraudes des documents administratifs.

Pour tester ou mesurer ces hypothèses, nous recourons à des variables tels que le temps moyen de vérification des documents administratifs avant et après l'implémentation de notre solution, et le niveau de satisfaction d'un échantillon de services administratifs. L'on pourra également s'appuyer sur le taux d'incidents de sécurité liés à la manipulation ou à la falsification des documents administratifs sur une période donnée.

1.3. Objectif du sujet

L'objectif principal de notre sujet est de proposer une approche d'authentification de documents administratifs à l'aide de la technologie blockchain.

De façon spécifique, il s'agit pour nous, de :

- faire une revue de littérature de la technologie blockchain et son utilisation dans les processus d'authentification des documents ;
- proposer une approche d'authentification de documents administratifs à l'aide de la blockchain ;
- implémenter une plateforme numérique basée sur l'approche et qui offre la possibilité de vérifier l'authenticité de documents administratifs.

1.4. Résultats attendus

Les travaux devront aboutir à des résultats qui présentent la manière dont les structures pourront utiliser la blockchain pour résoudre, entre autres, leurs problèmes de falsification et de fraude documentaire. Concrètement, les résultats suivants sont attendus au terme de notre étude :

- une revue de littérature de la technologie blockchain et son utilisation dans les processus d'authentification des documents est faite ;
- une approche d'authentification de documents administratifs à l'aide de la blockchain est proposée ;
- une plateforme numérique basée sur l'approche et qui offre la possibilité de vérifier l'authenticité de documents administratifs est implémentée.

1.5. Organisation du travail

Le présent mémoire est l'aboutissement de plusieurs travaux suivant un projet de chronogramme spécifique illustré par la figure 1 ci-dessous.

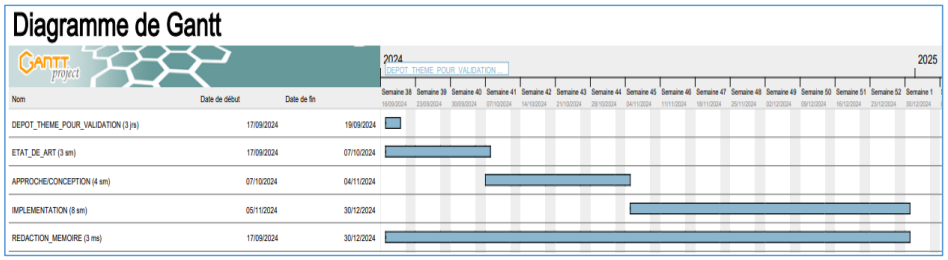


***Figure 1** : Projet de chronogramme des travaux*

Ce mémoire est organisé en cinq (05) principaux chapitres. En dehors de ce présent **chapitre 1**, le reste du présent mémoire est structuré comme suit :

- **le chapitre 2 est intitulé « technologie blockchain ».** Il est consacré à l'historique et à la définition de la blockchain et ses concepts dérivés. En plus des types de blockchain, nous y présentons l'architecture de la blockchain, les contrats intelligents et les protocoles de consensus. Nous y présentons particulièrement un exemple de blockchain populaire, à savoir Ethereum.
- **le chapitre 3 a pour titre « état de l'art sur l'authentification des documents à l'aide de la blockchain ».** Ce chapitre fait une synthèse des connaissances sur l'authentification de documents, les méthodes existantes d'authentification de documents, et les travaux existants sur les processus d'authentification de documents à l'aide de la blockchain. Nous menons, à la suite de cette synthèse, une discussion sur les travaux existants.

- **le chapitre 4 « approche d'authentification de documents à l'aide de la blockchain »** est dédié à la présentation de notre approche de résolution de la problématique. Nous y listons les étapes à suivre pour la mise en œuvre de notre solution. Nous y fixons aussi le type de document administratif sur lequel nous travaillons.
- **le chapitre 5 « implémentation de l'approche »** est réservé à la mise en œuvre concrète de notre approche. Nous y présentons le protocole d'implémentation à savoir l'environnement d'implémentation de la solution, les éléments de conception, les outils et technologies, etc. C'est également le lieu de présenter la solution développée et faire une discussion sur les résultats obtenus.
- **la conclusion** générale dans laquelle nous dressons un bilan des principales actions réalisées dans le cadre de nos recherches sur le sujet. Ce bilan est consolidé de perspectives. Le présent mémoire prend fin avec la présentation des références bibliographiques/webographiques et des annexes.

Au terme de ce premier chapitre, nous avons décrit le contexte de notre sujet de recherches « *Authentification de documents administratifs à l'aide de la blockchain* ». Nous y avons également énoncé la problématique à résoudre, tout en formulant les hypothèses et les objectifs de recherche. Dans le chapitre suivant, nous faisons un état des connaissances sur la technologie blockchain.

CHAPITRE 2 :

TECHNOLOGIE BLOCKCHAIN

CHAPITRE 2 : TECHNOLOGIE BLOCKCHAIN

On ne peut parler de Blockchain sans aborder la notion de transaction. En générale, la transaction est une opération d'échanges qui implique plusieurs parties [3]. Dans ce sens, l'opération peut être commerciale ou boursière, un contrat ou accord, etc. Parmi les parties impliquées dans une transaction, occupe en bonne position un tiers de confiance qui permet d'une part de sécuriser la transaction et d'autre part de certifier la validité de ladite transaction. En effet, le tiers de confiance est une entité neutre et indépendante telle qu'une institution financière ou un notaire.

La mondialisation de ce système de transaction à partir des années 1960 a fait augmenter de manière fulgurante, le nombre de transactions. Et cela a entrainé l'accroissement des risques liés à l'authenticité des transactions, qui, jusque-là étaient transcrites dans des registres (documents) physiques. Mais grâce à l'avènement et à l'évolution rapide de la Technologie, notamment dans les domaines du web et de la cryptographie, le principe du tiers de confiance disparait progressivement au profit de ce qu'il convient d'appeler « Blockchain ».

Dans ce chapitre, nous faisons une revue de littérature sur la technologie blockchain. En plus d'aborder les origines et définitions de la blockchain, nous présentons sa classification et son architecture. Nous y parlons également des différents mécanismes de consensus, des contrats intelligents et de la blockchain Ethereum en particulier.

2.1. Historique et définitions de la blockchain

2.1.1. Historique de la blockchain

Dès les années 1990, la perspective de digitalisation des documents sous format numérique soulevait déjà la question de savoir comment certifier la date à laquelle un document a été créé ou modifié pour la dernière fois. Comment faire en sorte qu'un utilisateur ne puisse pas antidater ou modifier la date d'un document mis sur support numérique ? dans [17], *Stuart Haber et W. Scott Stornetta* ont proposé dans ce sens, des procédures informatiques pratiques pour l'horodatage numérique de documents sous forme numérique. Leurs réflexions sur entre autres le hachage, et les signatures numériques, les ont permis d'incorporer, en 1992, le concept d'arbre de Merkle au

système d'horodatage de documents avec le concours de *Dave Bayer*. Cette innovation a amélioré l'efficacité du système en permettant à plusieurs documents d'être regroupés en un seul bloc [6].

Dans [30], le chercheur *Ittai Abraham* a affirmé : "*The longest running blockchain started in 1995 and is still running strong today. (…)*" ; ceci pour indiquer que le premier système de certification décentralisé est celui de la société Surety, qui publie chaque semaine depuis 1995 un certificat cryptographique de sa base de données dans la rubrique « Annonces et objets trouvés » du « New York Times ». Pour en venir, le concept de Blockchain en lui-même, basé sur la cryptographie, a été évoqué pour la première fois au début des années 1980. Mais de nos jours, il est impossible de dissocier les concepts de blockchain et de crypto-monnaies car elles constituent le point essentiel d'émergence de la blockchain. En effet, la crypto-monnaie est une monnaie virtuelle dont l'implémentation repose sur des algorithmes cryptographiques permettant de générer de la monnaie et de faire des transactions anonymes entre des paires sur internet. Dans ce sens, le bitcoin, une crypto-monnaie qui s'est rapidement imposée de manière non triviale, a été annoncé en 2008 par son mystérieux – mystérieux, car « Satoshi Nakamoto » est largement considéré comme un pseudonyme, et la véritable identité de l'inventeur du bitcoin reste un inconnue – développeur, **Satoshi Nakamoto** [27]. Le Bitcoin[1], selon *S. Nakamoto* dans [23] (plus détaillé par *G. Ferréol et R. Romain* dans [16]), repose sur trois fondamentaux à savoir : le réseau pair-à-pair sans autorité centrale, les transactions et le triple protocole de vérification-consensus-validation. Ces éléments constituent une chaîne de blocs (ou blockchain en anglais).

2.1.2. Définitions de la blockchain

La blockchain se définit de plusieurs manières.

De manière basique, la blockchain est une technologie numérique de stockage chronologique et de transmission d'informations sous forme de blocs reliés les uns aux autres de manière sécurisée et sans autorité centrale. En termes plus simple, le Mathématicien *Jean-Paul Delahaye*, la définit comme étant : « *un très grand cahier, que tout le monde peut lire librement et gratuitement, sur lequel tout le monde peut écrire, mais qui est impossible à effacer et indestructible* ».

De manière technique, la blockchain est assimilable à une base de données distribuée, dont les informations envoyées par les utilisateurs et les liens internes à la base sont vérifiés, puis groupés

[1] Bitcoin est le réseau et le bitcoin est la crypto-monnaie.

à intervalles de temps réguliers (environ 10 minutes dans le cas de Bitcoin) en blocs. L'ensemble de ces blocs est sécurisé par cryptographie et forme ainsi une chaîne de plus en plus longue. Par extension, une chaîne de blocs est une base de données distribuées qui gère une liste d'enregistrements théoriquement 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 la création de la transaction initiale [51].

On peut donc comprendre que cette technologie est basée sur le concept de grand livre distribué ou de base de données partagées. Cela implique que dans le réseau blockchain, chaque participant (nœud) au réseau possède sa propre copie de la base de données. Pour y parvenir, **un mécanisme (ou algorithme) de consensus** est utilisé. En effet, le mécanisme de consensus est essentiellement caractérisé par :

- **un accord unanime sur le contenu des données**, afin de permettre à tous les nœuds (ou du moins la majorité des participants) du réseau de valider et de s'accorder sur quelles données doivent être inscrites dans la blockchain, du fait de la présence de données contradictoires. La cryptographie est utilisée lors de la validation des transactions.
- **une conformité des copies des données convenues,** afin de s'assurer que toutes les transactions ajoutées à la blockchain sont les mêmes au niveau de chaque utilisateur.
- **une absence de tricherie par altérations des données ou tentative de fraude**, qui est garanti grâce à des mécanismes cryptographiques qui rendent toute tentative de modification ultérieure des données pratiquement impossible. En effet, dans une chaîne de blocs, les transactions sont horodatées en permanence. Cette sorte d'archivage empêche la suppression ou l'inversion des transactions une fois ajoutées à la chaîne de blocs, et dès que d'autres blocs ont été ajoutés à la suite.

Tout se passe dans un réseau distribué de sorte à ce que si un nœud tombe en panne, les autres peuvent continuer à fonctionner de façon transparente ; ce qui garantit la disponibilité et la fiabilité dans les transactions. On parle précisément de système distribué décentralisé. Dans ce système, chaque participant peut vérifier les informations de manière indépendante car les processus de vérification ne dépendent d'aucune autorité centralisée [4].

La technologie blockchain, dans son évolution, se distingue en différentes formes.

2.2. Types de blockchain

La classification de la technologie blockchain est possible du fait de son évolution générationnelle. Dans [5], *Imran Bashir* discute de quatre (04) générations (ou niveaux) de la blockchain. Il s'agit de la :

- Blockchain 1.0 : cette génération ne concernait que les crypto-monnaies car elle a été introduite avec l'invention du Bitcoin. Elle inclut donc les applications de base telles que les paiements et les applications servant à effectuer de simples transferts de valeurs.
- Blockchain 2.0 : elle est une évolution de la première génération à travers l'intégration des contrats intelligents et autres applications dérivées des services financiers.
- Blockchain 3.0 : les blockchains de la troisième génération ont permis d'envisager, au-delà de l'industrie des services financiers, de nombreuses autres applications à usage général telles que les médias, la santé, le gouvernement, etc.
- Blockchain X : la génération X permet de se projeter dans une vision où la blockchain va fournir des services dans tous les domaines de la société.

Outre cette évolution générationnelle, il existe une classification de la blockchain basée sur le réseau. Les principaux types de réseau blockchain sont les blockchains publiques, les blockchains privées et les blockchains de consortium.

Les blockchains publiques

Ces blockchains sont également appelées « blockchains sans permission ». C'est des blockchains qui sont ouvertes au public – donc accessibles à tous, au point de ne requérir aucune permission spécifique à l'entrée, ni au moment de réaliser une transaction – et chaque utilisateur peut conserver sans autorisation préalable, une copie du registre sur son nœud local. Les blockchains publiques sont aussi caractérisées par le fait que toute personne, en tant que nœud du réseau distribué, peut participer au processus de prise de décision. La prise de décision sur l'état de ce type de blockchain nécessite l'utilisation d'un mécanisme de consensus distribué. Il peut arriver qu'un nœud participant soit récompensé pour sa participation. Ce type de blockchain est en générale open source. Ethereum et Bitcoin[2] sont des exemples de blockchains publiques [18].

[2] Détails à : https://www.blockchain.com/fr/explorer

Les blockchains privées

Les blockchains privées ne sont pas ouvertes au public. Elles sont exclusivement accessibles sur invitation. Tous les membres participants de ce réseau blockchain se connaissent et se font confiance. Les membres participants peuvent être un groupe d'individus ou d'organisations qui ont décidé de partager le registre entre eux. Dans ce type de blockchain, un mécanisme de consensus est utilisé pour valider l'écriture des données parmi ses participants privilégiés. Par exemple, cette approche est plus appropriée lorsque plusieurs succursales d'une même entreprise (organisation) décident de l'utiliser pour se partager directement des informations. Aussi appelées blockchains permissionnées ou blockchains à autorisation, Hyperledger et Ripple sont des exemples de blockchain privée fréquemment cités [18].

Les blockchains semi-privées

Encore appelé blockchain de consortium, ce type de blockchain constitue un mixage entre la blockchain publique et la blockchain privée.

Dans ce type de blockchain, seuls quelques nœuds sélectionnés sont prédéterminés à se partager la responsabilité de la maintenance et de la sécurisation du réseau blockchain. Ils ont la responsabilité de déterminer les droits d'accès aux données. Les nœuds participants, eux, sont invités. Les décisions sont prises par la majorité des acteurs présélectionnés. Cela signifie qu'en dehors des données spécifiques stockées et contrôlées dans la blockchain, le reste des données sont accessibles au public. Ainsi, des membres publics peuvent vérifier (à l'aide de contrats intelligents) si les transactions privées ont été effectuées. Le fait d'avoir des droits de lecture pouvant être publics ou limités aux participants permet de préserver la confidentialité des données, comme dans les blockchains privées. BigchainDB, EEA et R3 sont des exemples de blockchain de consortium [2].

Outre ces trois (03) principaux types de blockchain selon les attributs réseau, il existe des blockchains dérivées telles que les sidechains (chaînes secondaires ou chaîne de transactions gérée par une sous-communauté) [3], les grands livres autorisés, les grands livres distribués, les grands livres partagés, les blockchains entièrement privées et propriétaires, les blockchains à jetons, les blockchains sans jetons, etc. [5]. Nous nous intéressons aux grands livres autorisés et aux blockchains entièrement privées et propriétaires.

Un grand livre autorisé est une blockchain dans laquelle l'utilisation d'un mécanisme de consensus distribué n'est pas nécessaire, car les utilisateurs sont connus et se font confiance. Les participants au réseau du grand livre autorisé peuvent utiliser un protocole d'accord pour maintenir une version partagée de la vérité sur l'état des enregistrements dans la blockchain. Dans ce cas, il n'est pas nécessaire que le grand livre autorisé soit privée, car elle peut être publique avec un contrôle d'accès réglementé. Les grands livres autorisés sont aussi appelés blockchains ou registres avec permission.

Comme leur nom l'indique, les **blockchains entièrement privées et propriétaires** ne sont pas ouvertes au grand public. Mais, dans des contextes privés spécifiques au sein d'une organisation, il pourrait être nécessaire de partager des données et de fournir un certain niveau de garantie quant à l'authenticité des données. Ces blockchains pourraient être utiles, par exemple, pour la collaboration et le partage de données entre différents départements gouvernementaux [5].

Bien qu'il y ait plusieurs types de blockchains, la structure et le mode fonctionnement de l'ensemble des blockchains qui permettent de garantir la sécurité des transactions restent quasiment les mêmes.

2.3. Architecture de la blockchain

Pour une meilleure compréhension de cette section, il semble indispensable de revenir sur les termes « hash », « arbre/racine de Merkle », et « mineur » qui sont des composants fondamentaux de la technologie blockchain.

Un hash est le résultat d'une fonction mathématique (SHA-256, Ethash, MD5, etc.) qui permet d'obtenir une empreinte numérique unique de taille fixe à partir d'une donnée d'entrée (fichier, texte, …) de taille non bornée. Par exemple, l'algorithme SHA-256 (32 octets) produit toujours une valeur de sortie hexadécimale de 64 caractères. En cryptographie, l'empreinte numérique générée est difficilement (voire impossible) devinable par un humain. De plus, la fonction de hachage est irréversible en ce sens qu'on ne peut pas retrouver la donnée d'origine à partir du hash.

L'arbre de Merkle ou arbre de Hash (ou hachage) a été créé en 1979 par le cryptographe informaticien *Ralph Merkle*. Il est beaucoup utilisé dans la blockchain et la cryptographie [32]. C'est une structure de stockage de données sous forme d'arbre binaire où chaque nœud de l'arbre est identifié par un hash. En effet, les nœuds initiaux (nœuds enfants ou feuilles) sont associés à un

nœud supérieur appelé nœud parent. Ainsi, un nœud parent a un identifiant unique résultant du hachage de ses deux (02) nœuds enfants. Cette structure est répétée jusqu'au nœud racine ou racine Merkle (Merkle root), dont l'empreinte est associée à tous les nœuds de l'arbre. Cela permet de rechercher efficacement des transactions dans la chaîne de bloc et d'empêcher les falsifications [25]. L'annexe 1 présente une illustration de l'arbre de Merkle où les hash sont nécessairement couplés par paire de nœud – le nœud EF y étant seul et différent de la racine, a été dupliqué et couplé avec lui-même.

Un mineur ou validateur est un nœud (ordinateur doté de grande puissance de calcul) actif du réseau de blockchain qui sélectionne des transactions, les vérifie et participe à leur validation. Il se distingue des autres nœuds du réseau tels que les clients SPV (Simple Paiement Verification) et les clients Web [25]. Aussi, il convient de rappeler que *"Tous les mineurs sont des nœuds, mais tous les nœuds ne sont pas nécessairement des mineurs." P.73 [25].*

2.3.1. Structure de la blockchain

La blockchain, comme son nom l'indique, est une chaîne ou liste chainée reliant des **blocs en retour** et hébergée dans des nœuds en réseau. Un bloc regroupe plusieurs transactions.

En effet, la structure de la blockchain repose sur l'architecture réseau distribué Peer to Peer (P2P) – aussi appelé pair-à-pair en français – qui est un réseau d'égal à égal. Ce type de réseau regroupe un ensemble d'ordinateurs appelés nœuds qui partagent les informations ou fichiers entre eux de manière directe, rapide et abordable. Ces nœuds contiennent chacun, une copie de la blockchain et fournissent un consensus sur l'état de celle-ci à tout moment. La figure 2 ci-dessous présente un schéma de réseau décentralisé P2P où chaque utilisateur ou nœud possède à la fois le rôle de serveur et de client ; ce qui est différent pour les réseaux classiques.

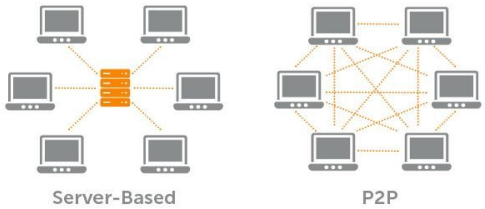


***Figure 2** : Réseau basé sur les Serveurs vs Réseau P2P*
Source figure : https://www.researchgate.net/figure/Reseau-base-sur-les-Serveurs-vs-Reseau-P2P_fig5_335174496

Dans une blockchain, les blocs sont plus ou moins importants en fonction du nombre de données qu'ils renferment. En effet, chaque bloc contient deux (02) parties à savoir l'entête (header) et le corps (facts) du bloc.

Le header contient plusieurs informations clés telles que [38] :

- **la version** qui indique le protocole de validation des règles ;
- **le hash du bloc précédent** qui assure la liaison entre les blocs afin de constituer la chaîne ;
- **le hash de la racine de Merkle** qui synthétise les informations que renferment toutes les transactions du bloc ;
- **le timestamp (date et heure de création)** pour l'horodatage du bloc qui précise le temps de minage ou de validation ;
- **les bits** qui indiquent la valeur actuelle de la difficulté de minage ;
- **le nonce** qui est un numéro aléatoire utilisé lors du minage pour trouver un hash valide.

En pseudo-code, un entête d'un bloc peut ressembler au contenu de la figure 3 ci-dessous.

```
BlocHeader: {
    Version: 1,
    PreviousBlockHash: "00000000000004X8G...",
    MerkleRoot: "3a5bc234ad...",
    Time: 1234567890,
    Bits: 1703ddf8c3,
    Nonce: 2085406893
}
```

Figure 3 *: Exemple d'entête d'un bloc*

Le corps du bloc contient les transactions qui doivent être stockées dans les bases de données. Ces transactions sont appelées « facts » ou « faits ». La transaction est l'élément de base de la blockchain Bitcoin (primitive des autres blockchains). La figure 4 ci-dessous présente un exemple simplifié d'une chaîne de blocs.

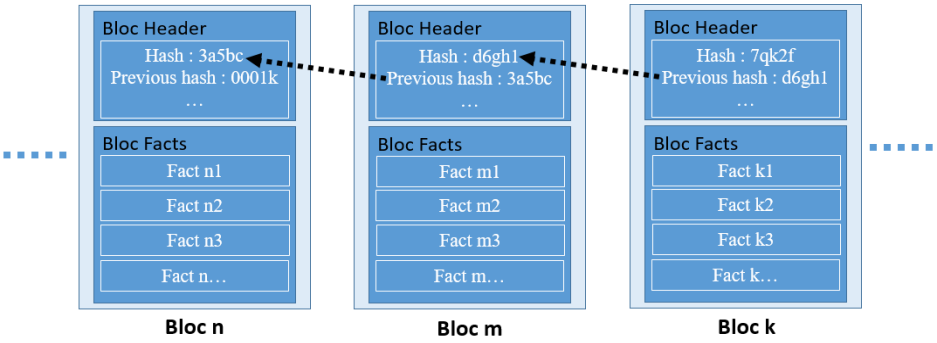


Figure 4 *: Schéma simplifié d'une chaîne de blocs*

Dans ce schéma, nous pouvons constater que ***bloc_m.previous_hash = bloc_n.hash*** *et* ***bloc_k.previous_hash = bloc_m.hash*** ; ce qui justifie la notion de blocs chaînés « en retour ». Cependant, il peut arriver qu'il y ait des chaînes de blocs orphelines (chaînes secondaires). Dans ce cas, la chaîne principale est composée de la plus longue suite de blocs après le bloc initial (ou bloc de genèse).

En générale, les facts sont organisés de manière séquentielle, de la plus ancienne à la plus récente, et peuvent être des transactions monétaires, des données médicales, des informations industrielles, des logs systèmes, etc. [25].

La logique de chaînage des blocs peut être décrite comme suit :
Soient B_0, B_1, B_2, les blocs représentant respectivement les bloc n, bloc m et bloc k de la figure 4 ci-dessus ; où B_0 est supposé bloc de genèse, B_1 le bloc fils de B_0 et B_2 le petit fils de B_0 et fils de B_1.

Chaque bloc de la chaîne est identifié de manière unique par un hash obtenu en SHA-256 (pour Bitcoin), Ethash (pour Ethereum), etc. Par exemple, la donnée (ou le texte) d'entrée α = ***Exemple de hash d'un bloc dans une chaîne de blocs*** a comme valeur de hash SHA-256, la sortie β = ***16958df5ae0040030217620a52f49e4398588cbbfa1f8bc1ef751fd2ba384ba5***. Et la moindre modification de α engendre obligatoirement un changement de β. Chaque bloc fait référence au bloc précédant à travers le hash de celui-ci. En effet, le hash de B_0 est référencé (ou inscrit comme previous_hash) dans l'entête de B_1 et celui de B_1 dans l'entête de B_2, formant ainsi une chaîne.

Dans une chaîne donnée, si l'identité du bloc de genèse ou d'un bloc parent change, l'identité des blocs enfants changera obligatoirement. Autrement dit, si un utilisateur modifie B_0, cette modification entrainera un changement du hash de B_0. Ce changement du hash de B_0 imposera un changement du pointeur « previous_hash » dans B_1 ; ce qui entraînera un changement du hash de B_1, qui, à son tour changera le hash de B_2, et ainsi de suite. De ce fait, cette opération de cascade implique le recalcul des hash de tous les blocs suivants dès qu'un bloc parent (ayant plusieurs descendants) viendrait à être modifier. Et plus la chaîne de blocs est longue, plus le recalcul devient énorme et coûteux, et plus l'historique (horodatage via la clé **timestamp** ou **time** du header) devient profond. Ceci explique le caractère immuable de la blockchain. Voir une simulation sur [1].

En sus de cette structuration, comment fonctionne la technologie blockchain ?

2.3.2. Fonctionnement de la blockchain

Le mécanisme global de fonctionnement de la blockchain passe par l'initiation d'une transaction, la validation de bloc via un consensus, et l'ajout du bloc validé à la chaîne précédente. Dans la figure 5 ci-dessous nous présentons le circuit d'une transaction pour illustrer ce mécanisme.

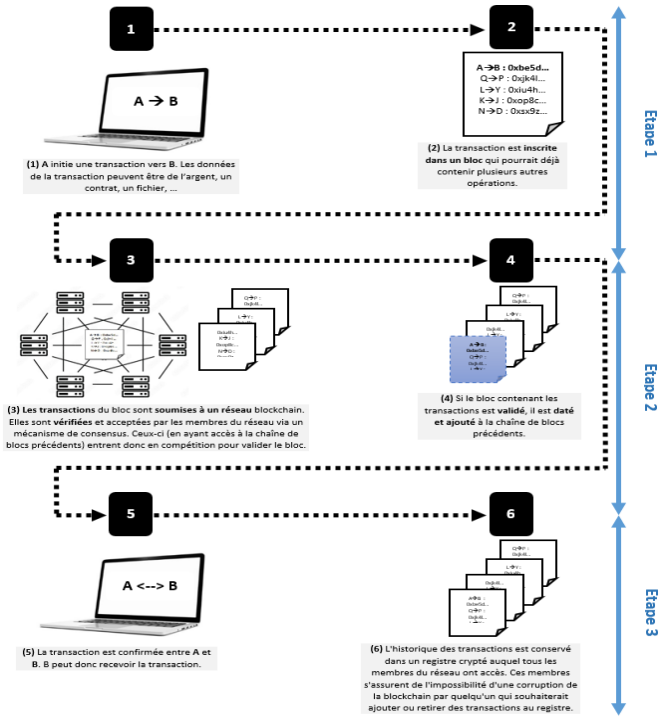


Figure 5 : *Circuit d'une transaction blockchain*

En effet, le mécanisme de fonctionnement est résumé en six (6) étapes à savoir :

1- Un utilisateur de la blockchain initie une transaction ***Tx***.
2- *Tx* est publiée (ou diffusée) sur le réseau de blockchain.
3- Les mineurs entrent en compétition, vérifie et valident *Tx*. Dans ce cas, une empreinte digitale (hash) est générée et associée à *Tx*.

4- Si *Tx* est vérifiée et validée, elle est ajoutée à d'autres transactions dans un bloc en construction. Ce processus permet de garantir que toutes les transactions sont légitimes. Légitimes pour indiquer que les transactions sont authentiques, valides (respect des tailles du bloc), non falsifiées, et acceptées par le réseau.
5- Après ce mécanisme de consensus, le bloc, désormais construit, est ajouté à la chaîne de blocs précédents (blockchain existante). Avant qu'un bloc ne soit ajouté à la chaîne existante, il est diffusé à tous les nœuds du réseau blockchain, suivi de l'acceptation de sa validité par ceux-ci. Ce bloc devient alors sécurisé et inaltérable. Cependant, si un bloc n'est pas entièrement validé, il ne peut pas être ajouté à la chaîne. Concernant les mécanismes de consensus, les plus populaires et les plus utilisés sont le Proof of Work (PoW) et le Proof of Stake (PoS) [38].
6- Ainsi, *Tx* est confirmée et considérée comme effectuée avec succès.

Dans le fonctionnement global de la blockchain, le concept de mécanisme (ou méthode) de consensus occupe une place importante. Qu'en est-il exactement ?

2.4. Protocoles de consensus

Dans le contexte de blockchain, le concept de consensus est indispensable et constitue l'épine dorsale de la technologie.

En générale, c'est une procédure qui consiste à dégager un accord sans procéder à un vote formel, ce qui évite de faire apparaître les objections et les abstentions [20]. Dit autrement par [41], le consensus désigne toute situation où plusieurs parties se mettent d'accord, sans possibilité d'opposition et sans que les intérêts de l'une ou l'autre des différentes parties ne se trouvent lésés. Le consensus s'établit généralement à l'unanimité, ou du moins à la majorité. Il est indissociable du mot voisin « consentement » : il ne revêt pas un caractère irréfutable, il s'agit de quelque chose que l'on admet, sur laquelle on s'accorde, et que l'on accepte comme une vérité ou comme une solution, en réponse à une question ou à un problème donné.

Appliqué à la Blockchain, le consensus est un processus sécurisé par lequel un groupe de pairs (ou nœuds) sur un réseau de blockchain parviennent à un accord unanime pour déterminer quelles transactions de la blockchain sont valides et lesquelles ne le sont pas. Ainsi, il permet de garantir que chaque nouveau bloc ajouté à la chaîne est la seule et unique version de la vérité vérifiée et

acceptée par tous les nœuds du système décentralisé et distribué. On parle alors de mécanisme de consensus ou d'algorithme de consensus [33]. D'une blockchain à une autre, la méthode utilisée pour parvenir à cet accord peut certes varier, mais comment parvenir à se mettre d'accord sur la blockchain ?

[34] *Satoshi Nakamoto*, en voulant vérifier l'authenticité d'un réseau blockchain et éviter les doubles dépenses dans le contexte de la crypto-monnaie, a mis en place en 2009, le premier consensus blockchain appelé « consensus de Nakamoto ». L'algorithme de consensus de Nakamoto intègre la tolérance aux pannes byzantines (BFT[3]) et combine une énigme informatique nécessitant des calculs extrêmement complexes (appelés Proof of **work**-Po**W** ou preuve de travail) afin de dissuader les acteurs malveillants du réseau. Par la suite, le consensus de Nakamoto a évolué en utilisant des ressources « **X** » rares et difficiles à obtenir (**X** pouvant désigner **W**ork, **S**take, **A**utority, **A**ctivity, **H**istory, etc.). En effet, toute Proof of **X** (Po**X**) incorpore généralement :

- la tolérance aux pannes byzantines afin permettre au réseau de continuer à fonctionner, même en cas de pannes de certains nœuds ou actions malveillantes de ceux-ci ;
- la communication synchrone entre les nœuds de sorte à ce que les messages ou transactions soient livrés dans un délai régulier ;
- une probabilité sur laquelle les nœuds s'accordent sur l'état du réseau ;
- la gouvernance qui indique les nœuds leaders responsables des validations des transactions.

Il existe plusieurs types de protocoles de consensus PoX à savoir : le Proof of Work (PoW), le Proof of Stake (PoS), le Proof of Autority (PoA), le Proof of Activity (PoA), le Proof of History (PoH), le Proof of Importance (PoI), le Proof of Alepsed Time (PoET), le Delegated Proof of Stake (DPoS), le Proof of Capacity/Proof of Space (PoC/PoSpace), et le Proof of Burn (PoB) [33].

En dehors des types de consensus PoX, il existe des consensus classiques basés sur le vote tels que la Tolérance pratique aux pannes byzantines (pBFT), la Tolérance aux pannes byzantines déléguée (dBFT), l'Accord de la Fédération Byzantine (AFB), le Radeau, etc. [34].
Sachant qu'il n'y pas que les types de protocoles de consensus PoX et que chacun de ces protocoles présente des avantages et des inconvénients, nous nous intéressons particulièrement aux PoW, PoS, et PoA qui sont les plus utilisés dans les réseaux blockchain.

[3] Détails à : https://river.com/learn/what-is-the-byzantine-generals-problem/ et https://crypto.com/glossary/byzantine-fault-tolerance-bft

La Preuve de travail (Proof of Work – PoW) [33] est le premier algorithme de consensus implémenté dans une crypto-monnaie et utilisé par la blockchain Bitcoin. Ce mécanisme nécessite une puissance de calcul considérable pour résoudre les problèmes mathématiques complexes et valider un bloc. Cela intervient lorsque parmi plusieurs mineurs (qui se concourent pour gagner la récompense du bloc ou coinbase transactions), chacun s'investi à trouver le Nonce correspondant au hash du bloc candidat à validation. La PoW est efficace en termes de sécurité. Mais son insuffisance est le fait qu'elle exige une quantité importante d'électricité et de ressources matérielles (Application-Specific Integrated Circuits, Graphics Processing Units, serveurs puissants, équipements de data center, ...) pour fonctionner.

La preuve d'enjeu (Proof of Stake – PoS), contrairement à la PoW, ne requiert pas une puissance de calcul pour valider les transactions et créer de nouveaux blocs. Dans ce mécanisme, les nœuds validateurs sont sélectionnés en fonction de la quantité de monnaie qu'ils sont prêts à mettre en jeu comme garantie. Le validateur est donc récompensé par des frais de transactions. Ainsi, les nœuds détenteurs de monnaie sont encouragés à agir honnêtement et à sécuriser le réseau de la blockchain (surtout contre les attaques à 51% [54]) au risque de perdre leur mise. Par exemple, la blockchain Ethereum est passé de la PoW à la PoS en 2022 [33].

La preuve d'autorité (Proof of Authority – PoA) quant à elle, est un algorithme de consensus approprié pour les blockchains d'entreprise du fait de sa faible consommation en énergie. Elle a été proposée en 2017 par *Gavin Wood*. Ce mécanisme de consensus oblige généralement les utilisateurs de la blockchain à dévoiler leur identité. Ainsi, au lieu de mettre des monnaies en jeu ou de disposer d'une énorme puissance de calcul comme dans les PoS et PoW respectivement, les validateurs mettent leur réputation en jeu pour obtenir le droit de valider les blocs dans la PoA [33]. De ce fait, même si la PoA a l'avantage d'être écologique, elle fait tout de même l'objet de critiques pour le fait qu'elle fonctionne sur le principe de centralisation de droits de validation sur la base de la réputation.

Outre les protocoles de consensus, le concept de contrat intelligent a émergé et fait désormais partie intégrante de la technologie blockchain.

2.5. Smart contracts (contrats intelligents)

Le terme « **smart contracts** » (ou **contrats intelligents** en français) a été utilisé pour la première fois en 1994 par *Nick Szabo* [29]. Il le défini dans [47] comme "*un ensemble de*

promesses, spécifiées sous forme numérique, comprenant les protocoles dans le cadre desquels les parties exécutent d'autres promesses". En terme simple, un contrat intelligent est un programme informatique implémenté, déployé et exécuté sur une blockchain lorsque certains critères ou spécifications sont remplis. Les contrats sont dits intelligents pour le fait qu'ils s'exécutent de façon autonome lorsque les conditions prédéfinies sont remplies. Leur récente popularité provient de l'émergence des blockchains (même s'ils ne sont pas disponibles dans toutes les blockchains), et notamment d'Ethereum. Les contrats intelligents sont applicables dans plusieurs domaines tels que la santé, les finances, le gouvernement, etc.

Selon [5], l'application des contrats intelligents constitue une caractéristique révolutionnaire de la blockchain pour le fait qu'ils permettent une flexibilité, une autonomisation (par programmation) et un contrôle très souhaitable des actions que les utilisateurs de la blockchain doivent effectuer en fonction de leurs exigences commerciales ou administratives spécifiques.

Techniquement, un contrat intelligent est un code source implémenté dans un langage de programmation de haut niveau tel que Solidity (orienté objet à typage statique), Vyper (de type python bien compatible avec l'EVM), etc., en combinant d'autres outils et technologies web. Ce code est ensuite compilé en bytecode à l'aide de Yul (langage intermédiaire pour compiler du code Solidity) par exemple. Une fois compilé, le contrat est déployé sur une version spécifique d'une machine virtuelle (machine virtuelle Ethereum – EVM) pour y être exécuter. La figure 6 ci-dessous est un exemple de contrat intelligent écrit en Solidity via l'environnement de développement Remix-Ethereum-IDE[4].

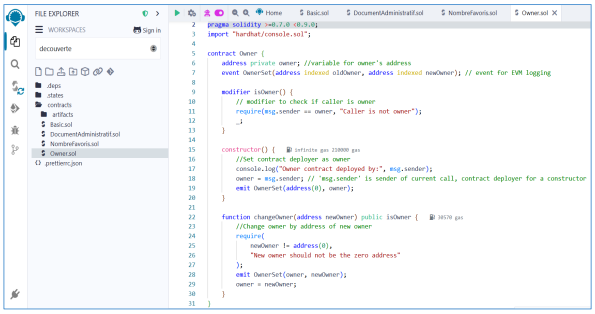


Figure 6 *: Exemple d'un contrat intelligent dénommé « Owner »*

[4] https://remix.ethereum.org/

Les contrats intelligents regorgent d'importants avantages mais également révèlent des insuffisances dans leurs applications. Il s'agit entre autres :

- **Avantages [50]** :
  - **Rapidité, efficience et exactitude :** le contrat est exécuté immédiatement dès que les conditions (if/when...then…) prédéfinies sont respectées. Aucun temps n'est consacré à rectifier les erreurs.
  - **Confiance et transparence** : aucun tiers n'est impliqué dans le contrat entre deux (02) utilisateurs. De plus, les enregistrements chiffrés des transactions sont partagés entre les participants, évitant ainsi la modification des informations à des fins personnelles.
  - **Sécurité et économies** : les enregistrements de transaction dans la blockchain sont chiffrés, donc impossible à détourner. De plus, chaque enregistrement est relié aux enregistrements précédents et suivants dans un grand livre distribué. Aussi, l'absence d'intermédiaire pour gérer les transactions conduit à l'absence ou à la réduction des délais et frais associés aux transactions.
- **Inconvénients [37]** :
  - **Erreur de l'oracle** : en générale, le contrat intelligent récupère la donnée d'un oracle (source de données) pour s'exécuter. Si cette donnée n'est pas fiable à la source, le contrat va tout de même s'exécuter sans possibilité de revenir en arrière.
  - **Piratage** : le contrat est exécuté par un programme informatique qui est lui-même sur la blockchain. Ainsi, si ce programme contient des failles ou des bugs, un hacker qui les découvre, pourrait les utiliser à d'autres fins malsaines.
  - **Immuabilité de la blockchain :** la blockchain étant par nature immuable, si le programme a fait une erreur ou s'il a été piraté, il n'est pas possible d'annuler une opération.

Les contrats intelligents sont une illustration parfaite de la possibilité qu'une blockchain soit programmable, contrairement aux premières versions de blockchain. Ethereum que nous présentons dans le point suivant est l'exemple populaire de blockchain programmable pour autres usages que les crypto-monnaies.

2.6. Exemple de blockchain : Ethereum

2.6.1. Historique et évolution d'Ethereum

La blockchain Bitcoin, depuis son déploiement en 2009, est un système décentralisé dédié exclusivement aux paiements digitaux en s'appuyant sur la crypto-monnaie bitcoin. En plus de cette exclusivité, *Vitalik Buterin* (programmeur russo-canadien et co-fondateur de Bitcoin Magazine) a publié en 2013, un livre blanc dans lequel il souligne les autres limitations de Bitcoin et décrit la vision et la conception technique d'Ethereum. En effet, selon *Buterin*, la technologie blockchain pourrait bénéficier de bien d'autres applications que les crypto-monnaies. Cette ainsi qu'Ethereum a été annoncé lors de la Conférence nord-américaine sur le Bitcoin à Miami en janvier 2014, puis lancé le 30 juillet 2015 [53].

Ethereum, conçue dans le but de lancer des applications décentralisées (DApps), est devenue tout de même la deuxième plus grande blockchain en termes de capitalisation boursière après Bitcoin. Pour être plus précis, il est énoncé dans le livre blanc [9] ceci : "*L'objectif d'Ethereum est de créer un autre protocole pour développer des applications décentralisées, en offrant un ensemble différent de compromis qui sera, nous le pensons, très utile pour une large gamme d'applications décentralisées. Il sera principalement axé sur les situations dans lesquelles le développement rapide, la sécurité des petites applications rarement utilisées et la possibilité pour les différentes applications d'interagir ensemble de façon très efficace sont importants.*". Cela fait d'Ethereum, une blockchain Turing Complete (peut effectuer tout calcul mathématique, y compris les boucles infinies et conditions) car elle permet à tout développeur d'écrire et d'y déployer des contrats intelligents et des DApps selon ses propres règles et logiques métiers d'entreprise [19]. Et c'est là l'innovation majeure qu'apporte Ethereum par rapport à Bitcoin. Pour rendre le réseau Ethereum plus solide et diversifié, les clients d'exécution (anciennement appelés « clients Eth1 » ou « client Ethereum ») ont été développés en utilisant plusieurs langages de programmations à savoir, Go, Java, Rust, C# et TypeScript [45].

Ethereum, dans son évolution et surtout suite à son piratage à travers la faille du code de The DAO le 17 juin 2016 [53], a connu plusieurs mutations. En effet, la gestion de cette attaque a abouti à la création de deux (02) chaînes de blocs distincts à savoir l'Ether (ETH) qui est la chaîne officielle et l'Ether classique (ETC) qui ne représente que 15% de la puissance de calcul des mineurs d'Ethereum avant piratage. Cela dit, nos recherches sont orientées vers l'ETH. Outre la scission de

la chaîne initiale, Ethereum a été critiqué (au même titre que d'autres blockchain) pour son émission de gaz à effet de serre très élevée (moins écologique). Et pour y remédier, Ethereum a changé son mécanisme de consensus de la preuve de travail (PoW) à la preuve d'enjeu (PoS) le 15 septembre 2022 (Ethereum vers Ethereum 2.0), réduisant ainsi sa consommation énergétique d'environ 99,95% [53].

2.6.2. Architecture et fonctionnement d'Ethereum

A l'image de Bitcoin, Ethereum fonctionne sur un réseau P2P. L'univers d'Ethereum est constitué de plusieurs éléments à savoir le réseau, les mécanismes de consensus, les nœuds et clients, l'EVM (Ethereum Virtual Machine), les comptes, les transactions et blocs, l'ETH (Ether native), les contrats intelligents et DApps, etc. Ces éléments sont regroupés en composants globaux permettant de schématiser Ethereum de manière simplifiée tel qu'illustre la figure 7 ci-dessous.

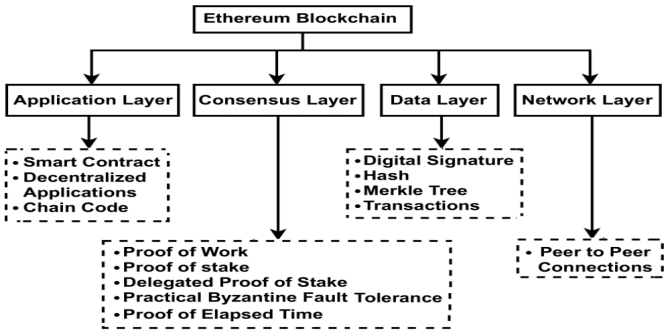


***Figure 7** : Architecture en couches de la blockchain Ethereum [19]*

Chaque couche de cette architecture joue un rôle spécifique dans le fonctionnement global du réseau Ethereum. La couche réseau (*Network Layer*) correspond au réseau P2P où les différents nœuds se communiquent directement. C'est dans la couche donnée (*Data Layer*) que sont sauvegardées les données issues des transactions. Il s'agit des signatures numériques qui garantissent l'authenticité et l'intégrité des transactions via la cryptographie asymétrique, les empreintes numériques utilisées pour identifier les transactions et les blocs, la structure arborescente des transactions stockées dans les blocs, et les valeurs des transferts. La couche consensus (*Consensus Layer*) met en œuvre les différents mécanismes de validation des transactions, et assure la mise à jour de l'état de la blockchain. Elle est aussi responsable de la sécurité. Quant à la couche applicative (*Application Layer*), elle permet aux utilisateurs d'interagir

à souhait avec la blockchain. C'est là que sont stockés et exécutés les contrats intelligents et les DApps.

Pour mieux appréhender Ethereum, nous définissons le rôle des différents éléments ou terminologies indispensables à son fonctionnement, comme suit :

a. **Nœuds** (comme fondation du réseau)

Selon [45], un nœud Ethereum est un ordinateur (ou machine réelle) sur lequel sont exécutés nécessairement deux (02) logiciels distincts appelés clients (client d'exécution et client de consensus), en vue de valider les blocs et les données de transactions. Le réseau Ethereum est donc simplement le regroupement de tous ses nœuds qui se communiquent directement. Le client d'exécution capture les nouvelles transactions publiées sur le réseau, les exécute dans l'EVM, et contient la dernière base de données et l'état de toutes les données Ethereum à jour ; tandis que le client de consensus implémente le PoS et la sécurisation du réseau. La figure 8 ci-dessous représente un nœud de moteur d'exécution couplé au client de consensus dans un réseau Ethereum.

Ainsi, dès qu'une transaction est demandée depuis un nœud, elle est publiée auprès de tous les autres nœuds du réseau qui la vérifient avant que le processus de validation ne s'enclenche. A la fin d'une validation, chaque nœud sauvegarde et met à jour la copie de chaîne de blocs.

Il existe trois (03) types de nœuds Ethereum à savoir le nœud complet, le nœud d'archive et le nœud léger.

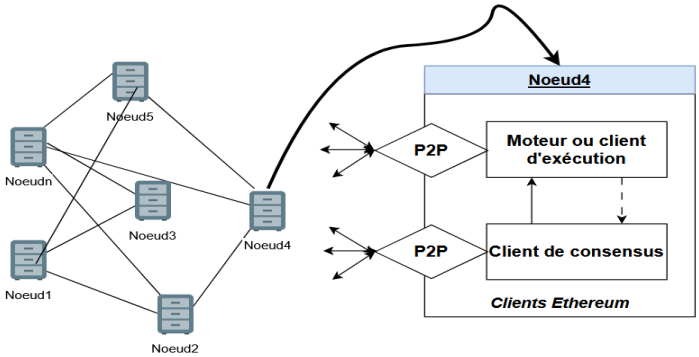


***Figure 8** : Schéma simplifié d'un nœud de moteur d'exécution couplé au client de consensus dans un réseau Ethereum*

b. EVM (comme environnement d'exécution)

[48] L'EVM (Ethereum Virtual Machine) est l'ordinateur virtuel unique et décentralisé sur lequel sont exécutés (sans demande de permission) les contrats intelligents ou toutes autres applications décentralisées ou (DApps). Il exécute ces codes de manière cohérente et sécurisée sur tous les nœuds Ethereum. Autrement dit, l'exécution d'une quelconque transaction ou d'un code spécifique modifie l'état de l'EVM. Et à chaque modification de l'état de l'EVM, tous les nœuds (à l'écoute permanente) du réseau Ethereum approuvent le nouvel état et gardent une copie de celui-ci. Cela permet de garantir la transparence, la décentralisation et l'immuabilité de la blockchain.

En outre, l'EVM dispose d'une fonction de transition d'état formellement décrite dans [28]. Il implémente également plusieurs opérations spécifiques à la blockchain telles que les address, balances, blockhash, etc.

c. Comptes (acteurs de base pour les transactions)

Les comptes Ethereum sont des entités disposant d'un solde en ETH. Les comptes et leur solde font partie de l'état global de l'EVM. Il existe deux (02) types de comptes Ethereum : le **compte de propriété externe (EOA- Externally Owned Account)** qui est contrôlé par toute personne ayant les clés privées et le **compte de contrat (CA- Contract Account)** qui est contrôlé par le code d'un contrat intelligent déployé sur le réseau [42]. En effet, ces deux types de comptes ont des caractéristiques différentes.

En effet, [42] un EOA est composé d'une paire de clés cryptographiques publique/privée qui permettent de contrôler les activités du compte. Ce compte interagit avec la blockchain en créant et en signant les transactions à l'aide de sa clé privée. La création de ce type de compte est gratuite et il offre la possibilité d'initier des transactions. Mais les transactions entre des comptes externes ne peuvent être que des transferts, réceptions, et détentions de valeurs en ETH et en jetons.

Quant au compte de contrat, sa création se fait moyennant un coût dû à l'utilisation de stockage du réseau. Ce type de compte n'a pas de clé privée et on ne peut qu'envoyer des transactions en réponse à la réception d'une transaction [42] provenant d'un EOA ou d'un autre CA. C'est-à-dire que les CA ne peuvent pas initier de nouvelles transactions par eux-mêmes. Mais, il est possible d'effectuer des transactions depuis un EOA vers un CA. Ces transactions peuvent déclencher un code pouvant

exécuter plein d'actions différentes, comme transférer des jetons ou même créer un nouveau contrat.

Outre ces principales différences, l'état de tout type de compte Ethereum est structuré en quatre (04) champs illustrés par la figure 9 ci-dessous.

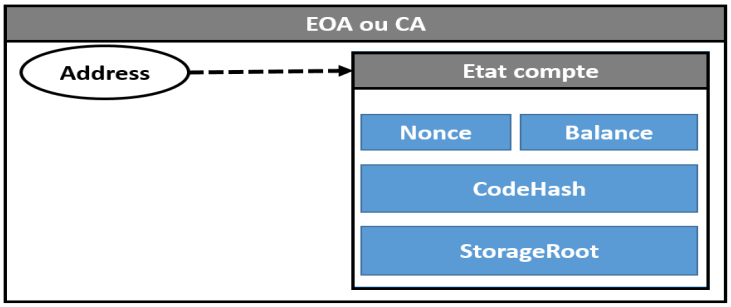


***Figure 9** : État simplifié d'un compte Ethereum*

Un compte Ethereum est identifié par une adresse unique (**Address**) qui est un code haché sur 42 caractères hexadécimaux ayant comme préfix 0x. On peut avoir comme exemple d'adresse *Account = 0x5e97870f263700f46aa00d968721199b9bc5a129*. Le champ **Nonce** représente d'une part le nombre de transactions envoyées à partir de l'adresse du compte s'il s'agit d'un EOA et d'autre part le nombre de contrats créés par le compte si c'est un CA. **Balance** est le nombre d'ETH exprimé en Wei (ou le solde actuel) possédé par cette adresse. Le **CodeHash** est l'empreinte numérique (hash) du code du compte (EOA ou CA) dans l'EVM. Contrairement aux autres champs du compte, le CodeHash n'est pas modifiable. Pour les EOA, ce champ CodeHash contient le hachage d'une chaîne vide. Mais pour le cas des CA, le code source du contrat intelligent sont hachés et stockés dans le CodeHash. Dans Ethereum, le contenu du compte est structuré sous forme d'arbre de Merkle (expliqué dans 2.3 précédemment) dont la racine est hachée sur 256 bits. Le champ **StorageRoot** (ou hachage de stockage) représente l'encodage de cet arbre. Il est vide par défaut.

d. Transactions (interactions entre comptes)

Une transaction ou demande de transaction (terme officiel) est une action initiée par un EOA (compte géré par un humain et non par un contrat). Cette action est une suite d'instructions signées cryptographiquement et dont l'exécution peut modifier l'état de l'EVM. Et pour qu'une demande de

transaction modifie l'état convenu de l'EVM, elle doit être validée, exécutée et diffusée sur le réseau par un autre nœud. [43] Cette demande peut être l'envoie d'ETH depuis un compte A vers un autre compte B, la publication d'un code de contrat intelligent sur l'EVM, l'exécution du code d'un contrat intelligent à une adresse spécifique dans l'EVM, etc.

Toute transaction contient les principales informations suivantes (voir exemple à l'annexe 2) :

- **from** : c'est l'adresse (émettrice) de l'expéditeur qui signe la transaction.
- **to** : c'est l'adresse du destinataire de la transaction. Dans le cas des transactions de création de contrat, l'adresse du compte du contrat n'existe pas encore et une valeur vide est donc utilisée.
- **signature** : elle est générée lorsque la clé privée de l'expéditeur signe la transaction, et confirme que l'expéditeur autorise ladite transaction.
- **nonce** : c'est le numéro de transactions dans la liste des transactions émises par l'expéditeur. On dira aussi que c'est le nombre de transactions envoyées par l'expéditeur [4].
- **value** : c'est le montant de l'ETH (en Wei) à transférer de l'expéditeur au destinataire. Dans le cas des transactions de création de contrat intelligent, cette valeur correspond au solde (Balance) de départ dans le compte de contrat nouvellement créé.
- **input data** : est un champ facultatif qui est souvent utilisé pour inclure des données arbitraires (par exemple, les nom, prénoms d'un étudiant dans le cas d'un contrat intelligent servant à enregistrer un diplôme universitaire).
- **gasLimit** : c'est la quantité maximale de gaz (estimée par l'EVM) pouvant être consommée pour l'exécution complète de la transaction. Ici, le gaz représente le coût informatique ou l'unité de mesure de la quantité d'efforts de calculs requis pour exécuter une opération sur le réseau Ethereum [52]. Une transaction simple de transfert requière 21 000 unités de gaz.
- **maxPriorityFeePerGas** (ou pourboire) : c'est la quantité maximale de gaz à inclure comme pourboire pour le validateur (uniquement) de la transaction. Ce montant incite le validateur à traiter la transaction plus rapidement.
- **maxFeePerGas** : c'est le montant maximum que l'expéditeur est prêt à payer par unité de gaz pour la transaction. Donc *maxFeePerGas* = *baseFeePerGas* + *maxPriorityFeePerGas* ; où baseFeePerGas = frais de base du réseau qui est ajusté dynamiquement en fonction de la congestion du réseau. Mais si à la fin de l'opération, le montant total réel payé ne vaut

pas le maxFeePerGas initial, le reliquat (*maxFeePerGas – (baseFeePerGas + maxPriorityFeePerGas)*) est retourné à l'expéditeur.

e. Blocs (comme conteneurs des transactions)

Un bloc est un regroupement (généralement des dizaines à des centaines) de transactions. Sur Ethereum, les blocs sont créés et engagés toutes les 12 secondes. Un bloc Ethereum a deux (02) parties : l'entête et le corps.

[40] L'entête d'un bloc contient les informations telles que slot (créneau auquel appartient le bloc), proposer_index (ID du validateur proposant le bloc), parent_root (hachage du bloc précédent), state_root (hachage racine de l'objet état), et body (qui contient aussi plusieurs autres champs). Le corps du bloc contient aussi plusieurs champs d'information tels que randao_reveal, eth1_data, attestations, voluntary_exits, execution_payload, etc.

Chaque bloc a une taille cible dont la limite est de 15 millions de gaz. Mais cette limite peut être ajustée à la hausse ou à la baisse par un facteur de 1/1024 par rapport à la limite de gaz du bloc précédent [40]. En tout état de cause, la quantité de gaz dépensée par toutes les transactions d'un bloc doit être inférieure à la limite de gaz dudit bloc.

f. ETH (comme monnaie pour payer et interagir)

[49] C'est la crypto-monnaie native d'Ethereum. L'ETH est créé par le protocole Ethereum, et non pas par un utilisateur. Chaque utilisateur d'Ethereum doit disposer d'une quantité suffisante d'ETH. Pour chaque transaction, une certaine quantité d'ETH (frais de gaz) est obligatoirement fournie par l'initiateur de ladite transaction. Cette quantité d'ETH est utilisée d'une part comme ressources pour effectuer les calculs afférant à la transaction et d'autre part comme prime pour simultanément récompenser les nœuds validateurs et empêcher les participants d'être animer de mauvaises intentions (bloquer le réseau par exemple). Étant donné que les validateurs misent (mettent en jeu) une partie de leurs ETH pour être favoris validateur, si un d'entre eux venait à mal se comporter, ses ETH sont probablement détruits par le réseau.

L'Ether dispose de plusieurs unités de valeur. Les plus importants sont le « Wei » qui est la plus petite quantité possible d'Ether et le « Gwei » (Giga-Wei) qui est couramment utilisé pour décrire les frais de gaz lisibles par l'humain. **1** ETH = $\mathbf{10^9}$ Gwei = $\mathbf{10^{18}}$ Wei.

g. Contrats intelligents

Bien que cet élément ait été présenté précédemment dans le point *2.5* du présent mémoire, rappelons tout de même que n'importe quel développeur peut créer un contrat intelligent, le stocker sur la blockchain et le rendre public sur le réseau, moyennant des frais (ETH) payés au réseau. Aussi, tout utilisateur peut invoquer et exécuter ledit contrat intelligent, encore moyennant des frais (ETH) payés au réseau.

Selon [55], les contrats intelligents sont assimilables à des API ouvertes pour le fait qu'un contrat intelligent peut appeler d'autres contrats et que certains d'entre eux, peuvent même déployer d'autres contrats. Par défaut, la taille maximale d'un contrat intelligent est de 24 Ko.

Les contrats intelligents sont très souvent confondus aux DApps. Cependant, pour être plus précis, une application décentralisée (DApp) est une application dont le code backend s'exécute sur un réseau décentralisé P2P, contrairement à une application traditionnelle, dont le code du backend est exécuté sur des serveurs centralisés. Une DApp combine un contrat intelligent (code backend) et une interface utilisateur (code frontend implémenté dans n'importe quel langage de programmation). C'est donc une application plus complexe des contrats intelligents [39].

En somme, nous avons proposé dans ce chapitre, un historique et des définitions de la technologie blockchain. Nous y avons également passé en revue les types de blockchain, la structuration, le fonctionnement et les mécanismes de consensus de la blockchain. Dans cette revue, nous avons découvert que les contrats intelligents ont révolutionné la manière d'effectuer les transactions blockchains, surtout dans les blockchains publiques telles qu'Ethereum qui est assez populaire.

Nous faisons dans le chapitre suivant, un état des connaissances sur les méthodes d'authentification de documents et les travaux existants en matière d'authentification de documents à l'aide de la blockchain.

CHAPITRE 3 :

ÉTAT DE L'ART SUR L'AUTHENTIFICATION DES DOCUMENTS À L'AIDE DE LA BLOCKCHAIN

CHAPITRE 3 : ÉTAT DE L'ART SUR L'AUTHENTIFICATION DES DOCUMENTS À L'AIDE DE LA BLOCKCHAIN

Dans quel cas peut-on conclure qu'un document est authentifié ou dit authentique ? afin d'y répondre, plusieurs éléments doivent être réunis.

C'est pourquoi dans ce chapitre consacré à l'état de l'art sur l'authentification des documents à l'aide de la blockchain, nous définissons d'abord la notion d'authentification de document. Ensuite, nous présentons différentes méthodes d'authentification de documents. Cette présentation est assortie d'une étude comparative de ces méthodes. Aussi, nous menons une discussion sur cet état des connaissances. Cette discussion est précédée particulièrement d'une présentation des travaux existants sur l'authentification des documents avec la blockchain.

3.1. Authentification de documents

L'authentification d'un document est un processus par lequel un système informatique ou un humain prouve ou certifie qu'un document est authentique. Et un document est dit authentique s'il s'agit de l'original, ou d'une copie conforme à/de l'original après *vérification* et *validation* par un sujet habilité ou compétent. Le sujet (humain), pour le cas du Burkina Faso, peut être un Officier de l'État Civil, un Officier de Police, l'Autorité officielle ayant délivré ledit document, le Greffe des cours et tribunaux, et le Notaire. Mais cette forme de certification n'a pas pour vocation de prouver la véracité du contenu du document. Elle a pour but de s'assurer que les signature et cachet visibles sur le document émanent d'une autorité officielle et que la date de signature ou délivrance, le nom et la fonction du signataire sont aussi lisibles.

L'authentification est un composant de la sécurisation qui, elle-même, est un ensemble de mesures à prendre et à mettre en œuvre pour garantir la traçabilité liée aux accès, et la protection des informations sensibles (électroniques ou physiques). La sécurisation vise à empêcher que les données soient manipulées ou reproduites de manière illicite ou non autorisée. Ainsi, qu'est-ce qu'un document administratif ?

Un document désigne une information conservée sur papier ou sur un support électronique. En effet, selon l'article 4 de [7], sont considérés comme **documents administratifs** : "*les documents produits ou reçus, dans le cadre de la mission de service public, par l'État, les collectivités*

territoriales ainsi que par les autres personnes de droit public ou les personnes de droit privé chargées d'une telle mission" p.4. Il s'agit par exemple des notes de service, des décisions, des instructions, des circulaires, des directives, des journaux, des délibérations, des rapports, des comptes rendus, des procès-verbaux, des croquis, des plans, des schémas, des avis, des prévisions, des communiqués officiels, des certificats (de prise-reprise-cessation de service, …), des bulletins, des décrets, des arrêtés, etc.

Cela dit, quels peuvent être les méthodes d'authentification de documents basées sur un système informatique ?

Commenté [AA1]: Pas necessaire

3.2. Méthodes d'authentification de documents

Il existe plusieurs méthodes d'authentification de documents basées sur des systèmes informatiques. Nous avons entre autres méthodes, la signature numérique (Digital Signature), le hachage et empreinte numérique (Document Hashing), le filigrane numérique et tatouage électronique (Digital Watermarking), l'horodatage électronique (Timestamping), la Blockchain et preuve d'existence, les RFID et codes QR. Nous présentons ci-dessous les principales techniques.

a. Technique de signature numérique

La signature numérique (ou signature électronique) est un moyen sécurisé qui permet d'authentifier l'auteur d'un document électronique et de garantir l'intégrité dudit document [46]. Elle permet ainsi d'assurer la non-répudiation c'est à dire, la quasi impossibilité de remettre en cause le document. De façon opérationnelle, l'émetteur (ou auteur), à l'aide de sa clé privée, génère d'abord la signature en appliquant un algorithme cryptographique à clé publique (RSA, ECDSA, etc.) sur le document. Et le destinataire peut ensuite vérifier l'authenticité du document en décryptant la signature avec la clé publique de l'émetteur. Pour ce faire, une PKI (Infrastructure à Clé Publique) est aussi souvent utilisée pour la gestion des certificats numériques. [24] Une signature numérique doit nécessairement remplir les conditions suivantes :

- authentique : l'identité du signataire doit pouvoir être retrouvée de manière certaine ;
- infalsifiable : une autre personne ne peut pas se faire passer pour le vrai signataire. La signature ne peut pas être falsifiée ;
- non réutilisable : la signature fait partie du document signé et ne peut être déplacée sur un autre document ;

- inaltérable : une fois que le document est signé, on ne peut plus le modifier ;
- irrévocable : la personne qui a signé ne peut le contester.

La technique de signature numérique est bien adaptée pour signer rapidement et facilement les documents électroniques administratifs et juridiques depuis n'importe où.

b. Technique de hachage ou empreinte numérique

Le hachage d'un document consiste à appliquer un algorithme (SHA-256, SHA-3) à sens unique sur le document pour générer une empreinte numérique unique. Ainsi, toute modification (aussi minime soit-elle) du document entraîne un changement considérable de cette empreinte.

Tout algorithme idéal de hachage doit avoir les propriétés suivantes [36] :

- le déterminisme : un message aura toujours la même valeur de hachage ;
- l'illisibilité : la valeur de hachage d'un message ne doit pas être déchiffrable ou compréhensible par un humain. Aussi, il ne doit pas être possible de générer (ou retrouver) le message d'origine à partir du hash ;
- la sécurité en cas de collision : la même valeur de hachage ne peut pas être attribuée à des messages différents. Cela réduit les points d'attaques et augment la sécurité.
- la continuité ou non-continuité : On parle de hachage continu lorsque l'algorithme est utilisé pour gérer des enregistrements et des messages similaires. Par contre, lorsque différents enregistrements et messages d'origine reçoivent autant de valeurs de hachage que possible, il s'agit de hachage non continu. Ceci est plus sécurisé.
- la vitesse : la valeur de hachage d'un message se calcule « facilement ».

La technique de hachage ne prouve pas l'identité de l'auteur ou du signataire, mais uniquement l'intégrité du document. C'est un complément à la signature numérique.

c. Technique d'horodatage électronique

L'horodatage électronique permet de prouver qu'un document existait à une date donnée et qu'il n'a pas été modifié. En effet, selon [8], l'horodatage a été défini pour la première fois comme le "*mécanisme associant une représentation de données à un instant donné et attestant de l'existence de la représentation de ces données à cet instant au moyen d'un jeton d'horodatage [qui] comporte un cachet du prestataire d'horodatage électronique établi à partir des données de signature du jeton d'horodatage*". Sa fiabilité est garantie par un certificat (jeton d'horodatage) qui contient à la

fois l'empreinte numérique du document (ou de la donnée), la date et l'heure UTC et le cachet du jeton d'horodatage. C'est donc une technique basée sur la signature numérique et le hachage.

d. Technique des RFID et codes QR

L'identification par radiofréquence (RFID – Radio Frequency Identification) et le code à réponse rapide (QR Code – quick response code) sont aussi des technologies utilisées dans de nombreux secteurs d'activités (commerce, administration, santé, ...) pour vérifier l'authenticité de certains documents ou pour suivre certains biens.

Une RFID est composée d'étiquette RFID, de lecteur RFID et d'un logiciel de gestion de données. En effet, la RFID, initialement inventée en 1980 par *Charles Walton*, utilise des ondes radio pour transmettre des informations marquées sur des étiquettes RFID à un lecteur RFID [35].

Les QR Codes ont, quant à eux, été développés en 1994 et sont plus simples à mettre en œuvre par rapport aux RFID. Du même coup, ils sont assez faciles à reproduire ou à manipuler, ouvrant certains problèmes de sécurité [31]. Les RFID et QR code ont des restrictions sur le type et le volume de données qu'elles peuvent stockées. Ces technologies sont utilisables sur des documents papier ou adhésifs (QR Code) ou du matériel physique palpable (RFID).

e. Technique Blockchain

La Blockchain, telle que présentée précédemment, est une alternative intéressante pour l'authentification de documents. En plus d'autres éléments tels que les mécanismes de consensus et la cryptographie asymétrique, elle combine l'horodatage électronique et le hachage. En effet, les empreintes numériques des documents peuvent être stockées dans une blockchain pour assurer une authentification décentralisée et inaltérable, offrant ainsi une preuve d'existence et d'intégrité desdits documents. D'un point de vue fonctionnelle, la technique de Blockchain offre la possibilité de faire publiquement des vérifications (sans autorité centrale), non seulement de l'intégrité d'un document, mais aussi de l'identité associée à la signature dudit document issu de la chaîne de blocs. Pour y parvenir, plusieurs plateformes et services dédiés tels que Ethereum, Blockcerts[5], les contrats intelligents, etc. peuvent être utilisés.

En s'intéressant particulièrement à la technologie Blockchain, nous constatons qu'elle peut être utilisée de manières différentes dans les processus d'authentification et de sécurisation de

[5] https://www.blockcerts.org/

documents électroniques. A cet effet, nous présentons, ci-après, deux (02) résultats de travaux antérieurs qui ont consisté à intégrer la blockchain respectivement dans un système gouvernemental d'authentification de documents électroniques [21] et dans un système de sécurisation de E-livrets scolaires [4].

Dans [21], *Isyak Meirobie et al.* ont présenté le résultat de leurs recherches qui ont abouti à la mise en place d'une plateforme d'authentification des documents électroniques à l'aide de la technologie Blockchain dans le système gouvernemental d'Indonésie. Les problèmes ayant suscités ces recherches sont le manque de sécurité dans le stockage de toutes les données des documents, les redondances profondes de données et la présence de tierces parties qui interfèrent dans les transmissions de documents. Afin de minimiser la falsification des documents et de rendre plus modernes et sécurisés les documents électroniques du gouvernement, la méthode a été de combiner la blockchain, des smarts contracts (contrats intelligents) et des Decentralized Autonomous Organization (DAO ou type de plus complexe des smarts contracts).

Selon l'approche, les gouvernements pourraient charger un ensemble de données et de documents sur une blockchain publique (sans nécessiter d'autorisation) et utiliser des signatures pour signer les transactions. Les signatures sont librement accessibles via le site web de l'institut. Et chaque gouvernement qui souhaite confirmer l'authenticité d'un document via la blockchain peut s'en assurer grâce à sa transcription numérique, tout en vérifiant que la transaction qui l'intègre à la blockchain est signée par le gouvernement lui-même. Au lieu de stocker toutes les données complètes sur la blockchain, seul le hachage de la signature SHA256 des données est stocké. Cela élimine la nécessité d'un stockage massif tout en garantissant l'intégrité et la vérification de toutes les données. La blockchain publique utilisée est sans licence, basée le processus de consensus PoA.

Concrètement, les auteurs de cette étude ont mis en place une interface utilisateur dénommé Go-Chain (Government Blockchain). Go-Chain est construit en 3 couches essentielles à savoir la couche de vérification, la couche des services de logique métier, et la couche d'accès/persistance de données. En amont, les documents (pdf ou word) gouvernementaux peuvent être téléversés, signés (via clé privée), transcrits numériquement en json et stockés (après calcul de la racine de Merkle) sur la blockchain. La transcription sous forme json peut être distribué au public. En retour, le public peut présenter le document haché à toute entreprise ou institution comme preuve valable. Cependant, pour tout de même vérifier la validité ou l'authenticité d'un document via la

blockchain, le public peut téléverser le document numérique gouvernemental dans Go-Chain, en y saisissant une clé privée. Après recalcule de la racine de Merkle, le cadre compare cette racine recalculée avec la racine de Merkle auparavant stockée sur la blockchain et signale si elle a été signée par une institution légitime. Pour la signature, l'auteur a utilisé un Digital Signature Algorithm (DSA) avec une courbe P-256. Et lorsque le document chargé par le public est valide, la clé publique, l'empreinte digitale SHA265 et d'autres données apparaissent sur l'écran de vérification Go-Chain.
En termes d'outils et de technologies, les auteurs ont utilisé HTML5, CSS3, JavaScript (ES6), Python 3, le microframework Flask et des serveurs HTTP.

Dans [4] *Ana BAKHOUM* a proposé, au profit du système d'enseignement moyen et secondaire du Sénégal, la dématérialisation du livret scolaire (d'où le E-livret). Le livret scolaire est un document administratif au format papier qui permet de répertorier les notes des élèves de la classe de sixième à la classe de terminal. Le même livret scolaire est transféré dans chaque établissement d'enseignements fréquenté par l'élève. Cette dématérialisation a constitué à la mise en place d'un système de recueil et de stockage (dans une base de données relationnelle MySQL hébergée par un serveur) des informations qui étaient dans le livret en papier. Dans cette dynamique, la problématique majeure traitée par l'auteur est comment assurer la fiabilité, l'authenticité, la transparence et la sécurité des E-livrets ? quelle architecture idéale, quel type de stockage utilisé ?

Dans ses travaux en lien avec cette problématique, l'auteur a fait une revue de littérature sur la technologie blockchain en générale et la blockchain Ethereum en particulier. Il a aussi passé en revue, la question de la sécurité informatique et celle notamment appliquée à la technologie blockchain. De ce qui est de la sécurité informatique en générale, il s'agit des obligations d'authentification, d'intégrité, de confidentialité, de disponibilité et de non-répudiation. Celles-ci pourraient faire face partiellement aux vulnérabilités, menaces, risques et attaques dans la blockchain. Car dans la pratique, il existe de multiples attaques qui manipulent directement ou indirectement le mécanisme de récompense (des mineurs), donnant ainsi d'injustes avantages aux mineurs de plus grandes tailles au détriment des petits mineurs.

En matière de sécurité selon les standards de la norme ISO/TC 307 [56], plusieurs propriétés sont intégrées dans la blockchain, notamment dans les applications basées sur les Distributed Ledger Technologies (DLT). Ce sont entre autres les propriétés :

- d’intégrité qui assure la protection de données contre toute modification après création ;
- d’authenticité qui permet de vérifier, qui enregistre une transaction dans le registre ;
- de confidentialité qui garantit que le registre est uniquement consultable par ceux qui y sont autorisés ;
- de disponibilité qui permet d’assurer la disponibilité à tout moment de toute transaction déjà enregistrée ;
- d’ordonnancement des événements rendant impossible le changement d’ordre des enregistrements dans le registre avec l’horodatage ;
- de « trusted-server less » permettant à la blockchain de toujours fonctionner malgré l’absence de serveur de confiance ;
- etc.

Dans la même logique, des mécanismes de sécurité ont été intégrés dans la blockchain tels que la cryptographie (surtout asymétrique), la signature numérique, le hachage.

L'étude a, à terme, permis de mettre en place un système décentralisé de sécurisation des E-livrets scolaires (SDSEL) en s’appuyant sur la technologie Blockchain, particulièrement sur Ethereum. Outre leur sécurisation, le SDSEL permet de valider ou de vérifier les E-livrets scolaires des élèves. La démarche a été de :

- développer le système (déjà existant et utilisé dans l’étude) de gestion des livrets électroniques (SGLE) qui permet de saisir et traiter, des informations des livrets scolaires à savoir les établissements, les élèves, les notes, les appréciations du conseil, etc.
- développer l’application décentralisée (DApp) dénommée « SDSEL » pour la sauvegarde des E-livrets dans la blockchain ;
- importer les informations des E-livrets depuis la base de données du SGLE vers la Blockchain des E-livrets ;
- consulter les listes et statistiques des élèves et leur livret ; cela permet de vérifier la conformité avec le livret généré par le SGLE ;
- développer (en perspective de l’étude) les contrats qui permettront à l’office du bac de générer automatiquement la liste des élèves inscrits en terminale qui servira à l’organisation de l’examen de baccalauréat ;

— déployer (en perspective de l'étude) le SDSEL dans la blockchain publique Ethereum afin qu'elle soit accessible par tous les établissements d'enseignements.

En termes d'outils et de technologies pour la mise en place de la DApp SDSEL, l'auteur a utilisé l'API JavaScript Web3, l'API JSON RPC, le langage de programmation Solidity, l'IDE Remix-IDE, le framework Truffle Framework qui intègre GANACHE, les frameworks Angular et Spring.

A la suite de cette revue de littérature, nous proposons un résumé comparatif de l'analyse de ces méthodes d'authentification de documents. Ce résumé est présenté à travers le tableau 1 ci-dessous.

***Tableau 1** : Analyse comparative de quelques méthodes d'authentification de documents*

Méthodes	Technologies & Protocoles	Avantages	Limites
Signature numérique (ou signature électronique)	PKI + Algorithmes (RSA, ECDSA, …) + PAdES + CAdES + OpenSSL + autres	• adaptée pour documents électroniques juridiques et administratifs • prouve l'identité de l'auteur et pratiquement l'intégrité du document	• nécessité d'avoir une PKI et un cadre règlementaire
Hachage	Algorithmes de hachage (SHA-2, SHA-3) + OpenSSL + autres	• prouve l'intégrité du document	• ne prouve pas l'identité de l'auteur (ou signataire) • utilisé en complément avec une signature numérique
Horodatage électronique	RFC 3161 + Autorité de certification + Adobe Timestamp Server + Blockchain + autres	• adapté pour documents électroniques juridiques et administratifs • prouve qu'un document existait à une date donnée et qu'il n'a pas été frauduleusement modifié ou antidaté	• utilisé en combinaison avec une signature numérique et une empreinte numérique du document
RFID et QR Code	QR Code + Base de données + puce RFID/NFC + autres	• adaptés pour documents papier nécessitant une vérification rapide	• mise en œuvre relativement complexe et coûteuse (RFID) avec de potentielles failles de sécurité (QR Code)

Commenté [AA2]: En faire une section : 3.3. Discussion des méthodes existantes. On peux fussionner cette section avec la section discussion et la section positionnement

			• non adapté pour documents électroniques
Blockchain	Ethereum + Smart contracts + P2P + autres	• adaptée pour documents électroniques de plusieurs types • prouve l'horodatage et l'immuabilité (donc l'existence du document) • prouve l'intégrité de document • vérification possible par le public et sans autorité centrale	• implique plusieurs technologies émergentes • complexe à mettre en œuvre

Les résultats des travaux déjà réalisés contribuent considérablement à l'avenir de la technologie blockchain qui est, elle-même, une révolution pour les processus informatiques. Quelle analyse synthétique peut-on faire de ces études ? comment y apporter de l'innovation si l'on souhaite adopter la blockchain pour résoudre la problématique d'authentification (voire sécurisation) de documents administratifs ?

3.3. Discussion

a mis en forme : Surlignage

Après analyse de la synthèse des travaux réalisés par *Ana BAKHOUM* [4] et les co-auteurs *Isyak Meirobie et al.* [21], nous décelons quelques points communs.

En effet, ces auteurs ont utilisé une **blockchain publique** dans le cadre de leurs travaux. Par exemple, *Ana BAKHOUM* a explicitement utilisé Ethereum dans ses travaux.

Aussi, les deux (02) acteurs ont utilisé la notion de **contrat intelligent**. Des contrats intelligents ont été développés afin d'automatiser des traitements sur la blockchain. Il s'agit par exemple de :

- la transcription numérique de documents, le hachage, la signature numérique et le stockage des données dans la blockchain ;
- l'importation et le stockage des E-livrets dans la blockchain ;
- la vérification de l'authenticité des E-livrets et des documents gouvernementaux depuis la blockchain ;
- etc.

Cela dénote de l'utilisation des propriétés de stockage décentralisé, de transparence (fiabilité) et d'intégrité de la technologie blockchain.

De plus, [4] et [21] ont travaillé sur des **documents électroniques,** même s'il s'agit de différents types de documents. Pour revenir à notre sujet du présent mémoire, cela pourrait être intéressant dans la mesure où l'Administration utilise couramment des documents électroniques ; sachant qu'un document est dit électronique, s'il est créé directement au format numérique (sans lien direct avec un support physique – on parle de document numérique) ou converti (numérisé).
En outre, dans les différentes solutions mises en place, toutes les données sujettes à authentification et devant être sécurisées **ne sont pas entièrement stockées** sur la blockchain, évitant ainsi les stockages massifs.

Hormis ces points communs, quelques applications suscitent des réflexions d'ordre technique et juridique. En effet, comment opérer le choix du processus de consensus qui puisse cadrer avec le type de blockchain adopté ([21] a utilisé le PoA) ? le cadre juridique national permet-il d'exploiter des signatures numériques de documents administratifs ? stocker des informations issues de documents administratifs dans la blockchain ne met-il pas en cause la souveraineté de l'État ? le cadre juridique étant évolutif, et la blockchain intégrant hautement les techniques de hachage et de cryptographie, ces inquiétudes semblent résolubles.

3.4. Positionnement

a mis en forme : Surlignage

Au regard des deux (02) points précédents (3.2 et 3.3), la méthode RFID et QR Code n'est pas appropriée pour résoudre notre problématique car d'une part, elle ne prend pas efficacement en compte l'authentification de documents au format électroniques de plusieurs natures. D'autre part, sa mise en œuvre s'avère complexe, coûteuse et comporte des potentielles failles de sécurité. Les méthodes de hachage et d'horodatage électronique, quant à elles, sont intrinsèquement liées à celle de la signature numérique et la rendent relativement robuste. Cependant, la Blockchain présente des bénéfices nettement plus attrayants avec quelques limites surmontables. Tout comme la signature numérique, elle prouve non seulement l'intégrité du document et l'identité de son auteur, mais est aussi utilisable pour l'authentification de documents électroniques juridiques et administratifs. De plus, la méthode Blockchain permet de prouver l'existence d'un document spécifique de manière inaltérable et indéniable. Applicable au processus d'authentification de tout type de document électronique administratif, la méthode Blockchain offre également la possibilité au public ou à une entité administrative de vérifier en temps réel l'authenticité d'un document sans passer par une autorité centrale.

De ce fait, nous retenons que parmi ces différentes méthodes, la Blockchain est la méthode la mieux indiquer pour assurer l'authentification et la sécurisation de documents administratifs.
Nous comptons nous inspirer des expériences de [4], [21] afin d'exploiter le potentiel de la blockchain Ethereum avec son protocole de consensus (PoS) pour la mise en place de notre approche. Cette approche doit respecter une logique précise que nous formulons comme suit :

$$\boldsymbol{Authenticit}\acute{e}(\boldsymbol{da}) \Leftrightarrow \boldsymbol{\mathcal{A}}(\boldsymbol{da}) = \boldsymbol{Vrai}\ \ et\ \ \boldsymbol{\mathcal{R}}(\boldsymbol{da}) = (\boldsymbol{\omega}, \boldsymbol{\lambda}, \boldsymbol{T})$$

Avec :

- $\boldsymbol{da}$: *un document administratif donné*
- $\boldsymbol{\omega}$: $l'ensemble\ spécifique\ d'informations\ contenues\ dans\ \boldsymbol{da}$
- $\boldsymbol{\lambda}$: $l'auteur\ ou\ le\ signataire\ de\ \boldsymbol{da}$
- $\boldsymbol{T}$: $la\ date\ et\ l'heure\ de\ signature\ de\ \boldsymbol{da}$
- $\boldsymbol{\mathcal{A}}$: $la\ méthode\ ou\ la\ technique\ d'authentification$
- $\boldsymbol{\mathcal{R}}(\boldsymbol{da})$: $le\ résultat\ de\ l'application\ de\ la\ méthode\ d'authentification\ \boldsymbol{\mathcal{A}}\ sur\ \boldsymbol{da}$

Où :

- si $\boldsymbol{\mathcal{A}}(\boldsymbol{da}) = \boldsymbol{Vrai}$, alors la méthode d'authentification appliquée au document administratif $\boldsymbol{da}$ retourne un résultat valide pour confirmer que $\boldsymbol{da}$ est authentique ;
- et si $\boldsymbol{\mathcal{R}}(\boldsymbol{da}) = (\boldsymbol{\omega}, \boldsymbol{\lambda}, \boldsymbol{T})$, alors la méthode d'authentification approuve que le document $\boldsymbol{da}$ contient bien l'ensemble d'informations $\boldsymbol{\omega}$, et qu'il est effectivement signé par l'auteur $\boldsymbol{\lambda}$ à la date $\boldsymbol{T}$.

Ainsi, le document administratif $\boldsymbol{da}$ est dit authentique si et seulement si, après avoir appliqué la méthode d'authentification $\boldsymbol{\mathcal{A}}$, le résultat $\boldsymbol{\mathcal{R}}(\boldsymbol{da})$ correspond effectivement à l'ensemble $\boldsymbol{\omega}$, à l'auteur $\boldsymbol{\lambda}$ et la date $\boldsymbol{T}$.

En conclusion, nous retenons que l'adoption de la blockchain est la mieux indiquée pour l'authentification de documents administratifs. Cette position découle de l'appréhension des notions sur les documents administratifs et l'authentification de document puis de l'étude comparative des méthodes d'authentification qui ont été abordées dans ce troisième chapitre. Dans le chapitre suivant, nous présentons, en tenant compte de ce qui précède, notre approche qui permet d'authentifier un type spécifique de document administratif.

CHAPITRE 4 :

APPROCHE D'AUTHENTIFICATION DE DOCUMENTS ADMINISTRATIFS À L'AIDE DE LA BLOCKCHAIN

CHAPITRE 4 : APPROCHE D'AUTHENTIFICATION DE DOCUMENTS ADMINISTRATIFS À L'AIDE DE LA BLOCKCHAIN

Le présent chapitre est le lieu de proposer la démarche pour la mise en place de la solution. Dans ce chapitre, nous fixons parmi tant d'autres, un type de document administratif sur lequel se basent nos travaux. Nous montrons également les principales exigences fonctionnelles prises en compte dans l'approche d'authentification de documents administratifs à l'aide de la blockchain. Enfin, nous y présentons les différentes étapes constitutives de l'approche.

4.1. Type de document administratif et exigences fonctionnelles

En se référant aux contexte et justification de notre présent sujet de recherche, nous choisissons le Communiqué officiel comme type de document administratif pour nos expériences. Ce type de document est relativement plus orienté grand public. De ce point de vue, en situation de faux communiqué, l'impact et les dommages sur les victimes sont assez énormes.

En termes d'exigences fonctionnelles, notre approche doit permettre :

- de générer et de réutiliser des paires de clés cryptographiques pour les besoins de signatures numériques à appliquer aux documents ;
- d'enregistrer un document ou une collection de documents administratifs électroniques sur la blockchain Ethereum ;
- de vérifier l'authenticité d'un document administratif électronique.

Ces éléments sont explicitement décrits dans le point suivant portant sur la méthode d'authentification.

4.2. Méthode d'authentification de documents administratifs à l'aide de la blockchain

L'analyse documentaire des recherches précédentes est un passage crucial et indispensable pour l'élaboration et l'implémentation de l'approche d'authentification de documents administratifs. Elle soutient la possibilité de créer une solution applicative d'authentification de documents en utilisant la technologie blockchain.

En effet, notre approche est basée sur la blockchain publique Ethereum, utilisant la preuve d'enjeu (Proof of Stake – PoS) comme protocole de consensus. Elle combine à la fois l'utilisation

de contrat intelligent, d'algorithmes de hachage (SHA-256), de signature numérique à l'aide de paire de clés cryptographiques avec l'ECDSA[6] et de technique d'horodatage électronique. Pour la signature numérique, nous utilisons spécifiquement la courbe elliptique secp256r1 (aussi appelée P-256) qui est compatible avec Ethereum et les normes du NIST[7] [15] [10].

Notre approche est composée de deux (02) principales exigences fonctionnelles ordonnées et complémentaires, à savoir :

- la possibilité d'enregistrer des documents administratifs numériques dans la blockchain (Phase 1) ;
- la possibilité de vérifier/authentifier des documents administratifs numériques sur la blockchain (Phase 2).

En effet, une entité administrative peut téléverser (upload) et stocker dans la blockchain Ethereum, un document administratif ou une collection de documents administratifs. Le stockage d'un document est précédé de hachage et de la signature numérique du contenu dudit document. Ainsi, tout usager (ou autre entité administrative) désirant vérifier l'authenticité d'un document administratif en sa portée, peut le téléverser également sur la blockchain Ethereum qui se chargera (après recherches et traitements) de confirmer ou d'infirmer l'authenticité dudit document. Ces deux (02) phases constitutives et complémentaires de l'approche sont illustrées par la figure 10 ci-dessous.

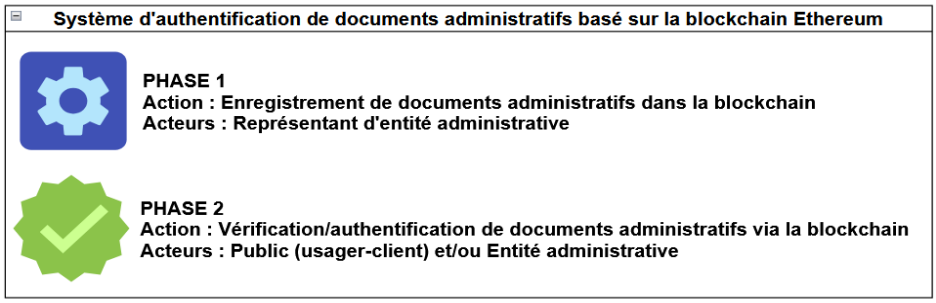


***Figure 10** : Phases constitutives de l'approche d'authentification de documents administratifs à l'aide de la blockchain Ethereum*

[6] ECDSA = Elliptic Curve Digital Signature Algorithm
[7] NIST = National Institute of Standards and Technology (https://www.nist.gov)

A leur tour, chaque phase est composée de plusieurs étapes. Pour qu'un utilisateur (public ou entité administrative) de l'approche puisse interagir avec la blockchain Ethereum, nous mettons en place une application décentralisée (DApp) à apparence d'application web classique. Il s'agit précisément d'une DApp hybride (de rigueur) car elle est constituée de backend et de frontend, et ne stocke aucune donnée dans une base de données ordinaire de type relationnelle, graphe ou NoSQL comme PostgreSQL, MariaDB, Oracle, MySQL, Cassandra, MongoDB, Apache Spark, Neo4j, etc. La DApp hybride sert uniquement de passerelle vers la blockchain Ethereum.

La PHASE 1 de notre approche est illustrée par la figure 11 ci-dessous.

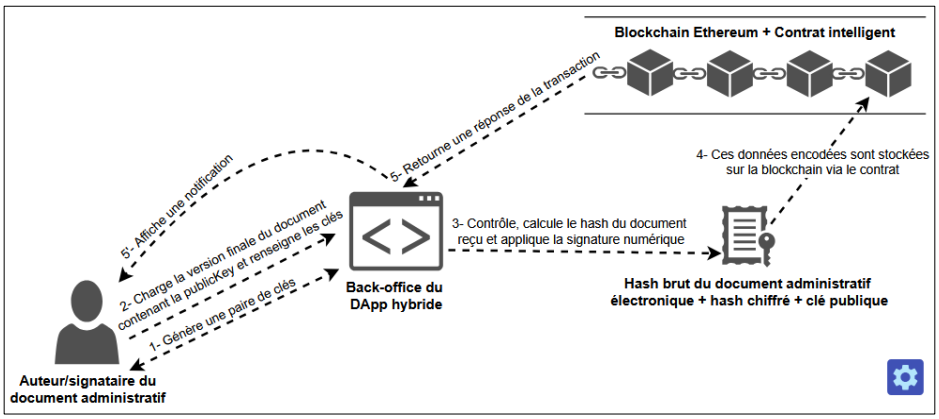


Figure 11 *: PHASE 1 - Enregistrement de document administratif dans la blockchain Ethereum en utilisant le hachage, la signature numérique et l'horodatage*

Selon ce schéma illustratif, les différentes étapes de la phase 1 sont décrites comme suit :

1. L'auteur génère une paire de clés (via le back-office de la DApp). Ces clés ne doivent pas être perdues et la clé privée doit être soigneusement conservée.
2. L'auteur intègre sa clé publique dans le document de sorte à ce qu'elle soit visible. Ensuite il upload (via le back-office de la DApp) la version finale en PDF ou Word du document contenant sa clé publique. Lors de l'upload, il renseigne la paire de clés. Le frontend transmet ces données au backend de la DApp.
3. Le backend extrait le contenu (textuel) du fichier uploadé, calcule un hash (ou empreinte numérique) de ce contenu. Le backend chiffre ensuite le hash calculé avec la clé privée de l'auteur. On obtient ainsi un hash signé – c'est la signature numérique du document.

Ces données obtenues (le hash brut, le hash signé et la clé publique de l'auteur) sont encodées et renvoyées au frontend de la DApp.

4. Le frontend transfère ensuite les données encodées reçues du backend (hash brut, hash signé et clé publique) au contrat intelligent qui les stocke sur la blockchain Ethereum. Lors du stockage, l'horodatage est calculé (et associé aux données) par le contrat intelligent. Ainsi, lorsqu'un utilisateur veut vérifier l'authenticité, la clé publique correspondante est utilisée pour déchiffrer et vérifier la signature du document.
5. Une notification est retournée au frontend depuis la blockchain en passant par le backend. Cette notification est personnalisée et affichée à l'utilisateur.

La PHASE 2 de notre approche est schématisée comme suit (voir figure 12 ci-dessous) :

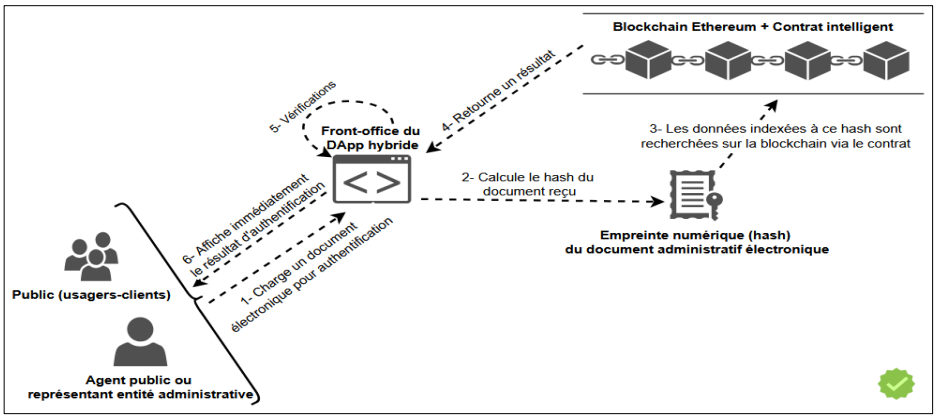


***Figure 12** : PHASE 2 - Vérification ou authentification de documents administratif sur la blockchain Ethereum en se basant sur le hachage, la signature numérique et la clé publique*

La phase 2 est également composée de plusieurs étapes pouvant être décrites comme suit :

1. L'utilisateur (public ou agent public) upload le document PDF ou Word (via le front-office de la DApp) dont il souhaite vérifier l'authenticité. Le frontend transmet ces données au backend de la DApp.
2. Le backend extrait le contenu (textuel) du fichier uploadé pour authentification et calcule le hash de ce contenu. Cette empreinte numérique (hash) est encodée et renvoyée au frontend de la DApp.

3. Le frontend utilise cet hash et interroge la blockchain Ethereum via le contrat intelligent (qui y est déployé) pour récupérer l'empreinte numérique brute, la signature numérique et la clé publique auparavant stockées dans la blockchain (à travers la phase 1).
4. La blockchain retourne une réponse à la DApp via le contrat intelligent. Cette réponse est en effet transmise au backend de la DApp pour vérifications.
5. Le backend décode les données reçues d'Ethereum et effectue deux (02) vérifications :
    a. Il compare le hash recalculé avec celui stocké. Si les hashs ne sont pas identiques, alors le document a été modifié et n'est donc pas authentique (c'est la vérification de l'intégrité du contenu document). Si les hashs correspondent, on passe à la seconde vérification (b).
    b. Il vérifie la signature numérique avec la clé publique stockée. Si cette vérification échoue, alors la signature ne provient pas du bon signataire, c'est-à-dire de l'auteur ayant validé la version finale du document. C'est la vérification de l'authenticité du document. Si la signature est valide, alors on déclare que le document est authentifié et authentique.
6. Après vérifications, une notification est faite au frontend. Ainsi, le résultat de l'authentification est immédiatement personnalisé et affiché à l'utilisateur.

La mise en place de cette approche doit respecter les principes de Design Pattern afin que les responsabilités soient situées à chaque étape des 02 phases. La figure 13 ci-dessous est un schéma simplifié de l'architecture en couches (ayant chacune une responsabilité) de notre approche.

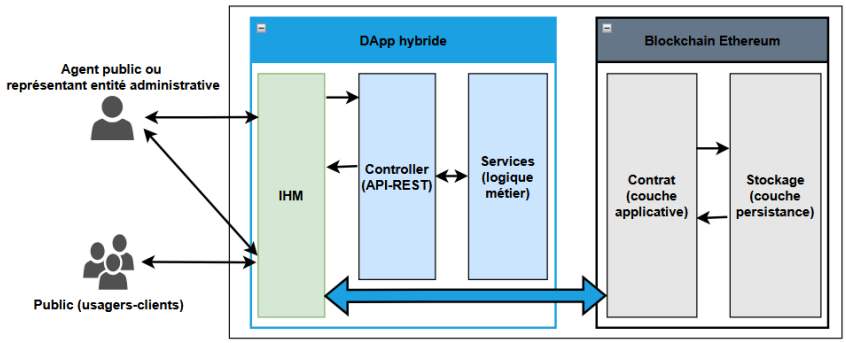


***Figure 13** : Architecture simplifiée en couches de notre approche d'authentification de documents adminitsratifs à l'aide de la blockchain Ethereum*

La couche IHM (frontend) de la DApp hybride est constituée de deux (02) espaces, à savoir le back-office qui est uniquement accessible par l'entité administrative pour les opérations de la phase 1 et le front-office destiné à tous (entités administratives et usagers publics) pour les opérations de la phase 2. Chacun de ces espaces peut interagir avec la couche Controller (1ère couche backend) de la DApp hybride. La couche Controller qui est la couche supérieure web du backend, a la responsabilité d'exposer des API REST au profit de la couche IHM. Elle communique également avec la couche Services (immédiatement inférieure du backend) de la DApp. La couche Services implémente la logique métier de chaque exigence fonctionnelle. Cette couche Service peut dans son fonctionnement, utiliser des utilitaires (chiffrer/déchiffrer, encoder/décoder, etc.). La responsabilité de la couche applicative de la blockchain Ethereum est d'héberger notre contrat intelligent et de tout mettre en œuvre en collaboration avec les autres couches de cette blockchain pour l'exécution du contrat. L'exécution du contrat intelligent peut se résumer par les demandes de transactions immuables qui renforcent la sécurité des documents administratifs et garantissent l'intégrité desdits documents. Cette couche applicative communique avec la couche IHM de la DApp hybride. Elle communique également avec la couche persistance d'Ethereum où sont stockées et recherchées les données hachées, chiffrées et encodées des documents administratifs. Cela permet d'avoir un stockage décentralisé des documents administratifs et de vérifier leur authenticité en temps réel.

En conclusion, nous choisissons le communiqué officiel comme type de document administratif à utiliser dans le cadre de nos travaux. Techniquement notre approche est une DApp hybride qui combine l'utilisation d'Ethereum, de contrat intelligent, de hachage, de la signature numérique et de l'horodatage électronique pour assurer l'authentification des communiqués officiels administratifs. Cette démarche permet de renforcer la sécurité et l'intégrité de ce type de documents administratifs, réduisant ainsi les cas de falsifications et fraudes documentaires. Elle offre également la possibilité de vérifier l'authenticité des communiqués en temps réel.

CHAPITRE 5 :

IMPLÉMENTATION DE L'APPROCHE

CHAPITRE 5 : IMPLÉMENTATION DE L'APPROCHE

Dans ce chapitre nous présentons les éléments de conception ayant servi la réalisation de notre approche d'authentification de documents administratifs à l'aide de la blockchain. En effet, il s'agit des acteurs et leurs rôles, des diagrammes de cas d'utilisation et de séquences, du diagramme d'interactions du contrat intelligent, des outils et technologies utilisés. Nous y présentons également des interfaces utilisateurs de la solution. La DApp hybride que nous avons implémentée suivant l'approche est dénommée « ADOBLOCK ».

5.1. Protocole d'implémentation

Les travaux de modélisation, d'implémentation, de déploiement en local et de tests de ADOBLOCK ont été effectués à l'aide d'un ordinateur de processeur Intel(R) Core(TM) i7-8565U @1.80 GHz fonctionnant sur Windows 10 Professionnel x64 avec 16 Go de RAM d'une vitesse de 2667 MHz et 256 Go SSD de stockage.

La suite des expérimentations de la solution et de la rédaction du présent mémoire a été faite à l'aide d'un ordinateur de processeur Intel Core i9 (8 cœurs) @2.4 GHz fonctionnant sur MacOs 15.5 (24F74) avec 32 Go 2667 MHz DDR4 de RAM et 1 To SSD de stockage.

Pour la mise en place d'une telle solution, nous estimons qu'il faut un ordinateur ayant les caractéristiques minimales suivantes : 1 processeur Core i7, 16 Go de RAM et 256 Go SSD de stockage.

5.1.1. Analyse et conception de l'approche

Commenté [AA3]: Ramener cette section à la fin du chapitre 4

La réalisation de l'approche nécessite une conception formalisée des spécifications fonctionnelles et techniques. Pour ce faire, nous identifions les acteurs sensés interagir avec la solution ainsi que leurs rôles. Nous concevons également dans cette partie, les principaux diagrammes.

a. Identification des acteurs et leurs rôles

Les acteurs humains destinés à interagir avec notre solution d'authentification sont les usagers/clients (le public ou encore tout citoyen) et les entités administratives (ou représentant

d'entité administrative). La figure 14 ci-dessous illustre les acteurs et les actions qu'ils peuvent effectuer à travers la solution.

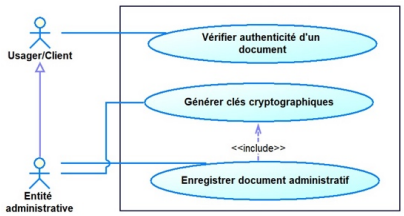


***Figure 14** : Diagramme de cas d'utilisation de la solution ADOBLOCK*

Le tableau 2 ci-dessous présente une description de ce diagramme de cas d'utilisation.

***Tableau 2** : Acteurs et rôles de la solution ADOBLOCK*

Acteurs	Rôles
Usager/Client	• Peut vérifier l'authenticité d'un document administratif électronique.
Entité administrative	• Peut générer une paire de clés cryptographiques pour les besoins de signatures numériques à appliquer aux documents. • Peut enregistrer un document ou une collection de documents administratifs électroniques sur la blockchain Ethereum. • Peut vérifier l'authenticité d'un document administratif électronique.

Relativement au fonctionnement de la blockchain, ces acteurs sont considérés comme un compte de propriété externe (EOA) caractérisé par (exemple) :

- ACCOUNT ADDRESS : *0x3b876edC4Efac286265Af907E81c9aB794B20530*
- PRIVATE KEY : *0xe8b99ad0cbfcdc23078f0fa201d16f1185913aee09cdf37eccc27d9bde6b9811*

Alors que le contrat intelligent faisant la jonction entre la blockchain Ethereum et notre DApp ADOBLOCK dispose nécessairement d'un compte de contrat (CA) caractérisé par (exemple) :

- CREATED CONTRACT ADDRESS : *0x3Db366A5908ec0384DEE67b0c37D6B3aB4003c35*.

b. Diagrammes de séquences

Ce type de diagramme permet de mieux appréhender les interactions entre les acteurs et la solution selon un ordre chronologique lors de l'exécution d'une fonction de celle-ci. A travers la figure 15 ci-dessous, nous présentons le diagramme de séquences d'enregistrement de document

administratif dans la blockchain Ethereum. Cette opération d'enregistrement de document nécessite d'avoir une paire de clé (privée/publique) à renseigner dans le formulaire d'ajout.

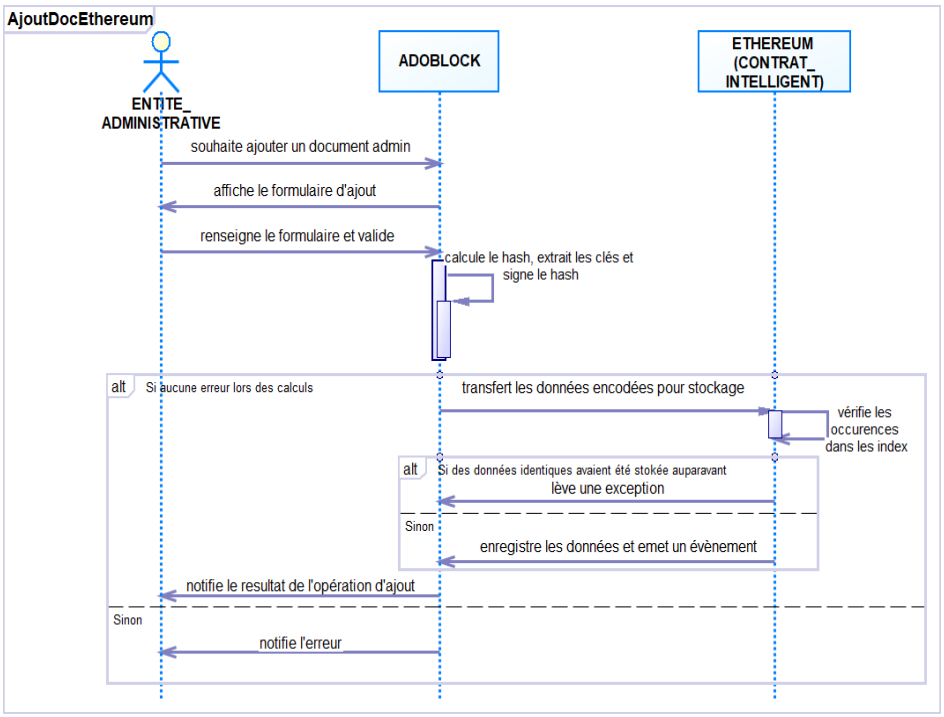


Figure 15 *: Diagramme de séquences d'enregistrement de document administratif dans Ethereum*

La figure 16 ci-dessous représente le diagramme de séquences d'authentification (ou de vérification) de document administratif depuis Ethereum. Pour cette opération de vérification, l'acteur a juste besoin d'avoir le document électronique à authentifier.

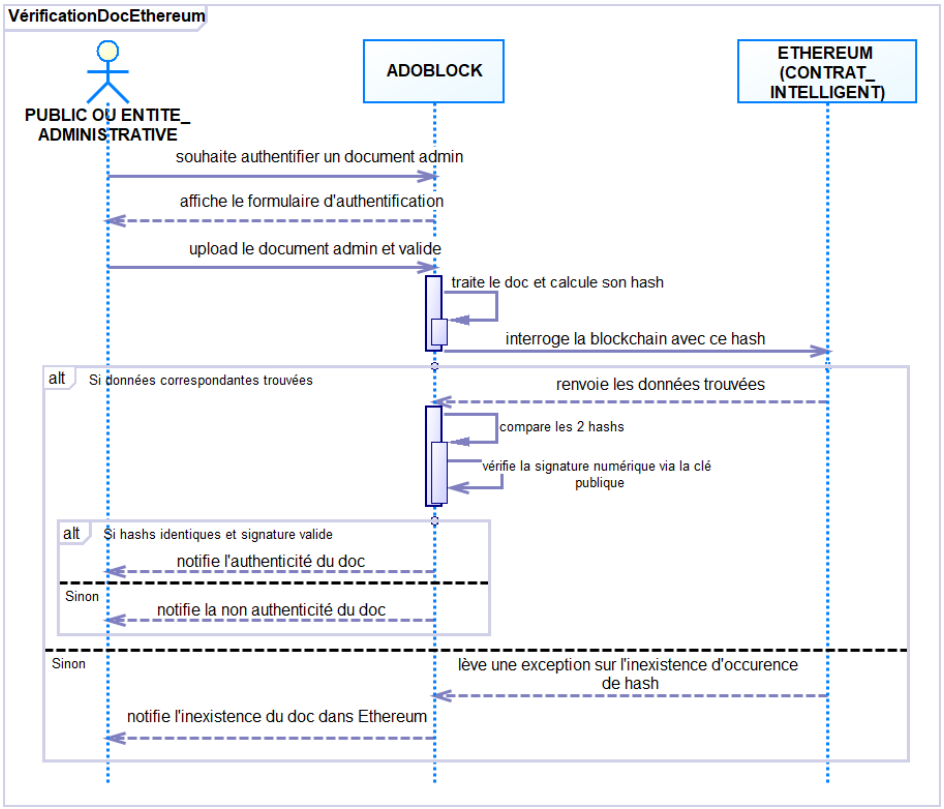


Figure 16 *: Diagramme de séquences d'authentification de document administratif depuis Ethereum*

c. Diagramme d'interactions du contrat intelligent

Ce diagramme offre une vue graphique du contrat intelligent (pouvant être assimilé à un objet dans une conception orientée objets-relationnel) et ses interactions avec les comptes EOA. L'utilité de notre contrat intelligent est de stocker un document administratif électronique sur Ethereum et de pouvoir le retrouver à tout moment. Le contrat est dénommé ***DocumentAdministratif*** (voir code source à l'annexe 3) et son déploiement peut coûter[8] environ 782016 GAS, soit 1487.818208 ETH (avec 894334 GAS ou 1701.50791 ETH comme coût estimatif de transaction de création du

[8] Opérations effectuées sur https://remix.ethereum.org/.
1 ETH = 525.613 GAS sur https://bitsgap.com/fr/converter/gas/eth (27/04/2025).

contrat). En tenant compte de notre approche, *DocumentAdministratif* est caractérisé par une structure d'informations, un mapping, un évènement et des fonctions. La structure d'informations regroupe quatre (04) attributs à savoir le hash du document, le hash signé (ou signature numérique), la clé publique de l'auteur (signataire), et le timestamp qui correspond à l'horodatage. Le mapping dénommé ***documents***, sert à mapper tout document administratif par son hash dans la liste des documents qui seront sur la blockchain. L'évènement ***DocumentStored*** nous permet d'avoir une réponse personnalisée après l'enregistrement réussi d'un document sur Ethereum. Il crée un index sur Ethereum en se basant principalement sur le hash du document. En cas d'échec d'enregistrement du document, cet évènement est quand même émis à l'attention de l'utilisateur. La fonction ***storeAdministrativeDocument(****string memory _hash, string memory _signedHash, string memory _publicKey****)*** sert à enregistrer un document sur Ethereum. A cette occasion, Ethereum se charge de calculer et d'associer l'horodatage à la structure d'informations. L'exécution de cette fonction déclenche une transaction qui peut coûter[9] environ 304004 GAS, soit 578.380349 ETH (avec 628.295438 ETH, et 379776 GAS comme respectivement le coût estimatif de transaction et le gas limit). La fonction ***getAdministrativeDocument(****string memory _hash****)*** permet de rechercher ou récupérer un document depuis la blockchain via le hash. Ceci est illustré par la figure 17 ci-dessous.

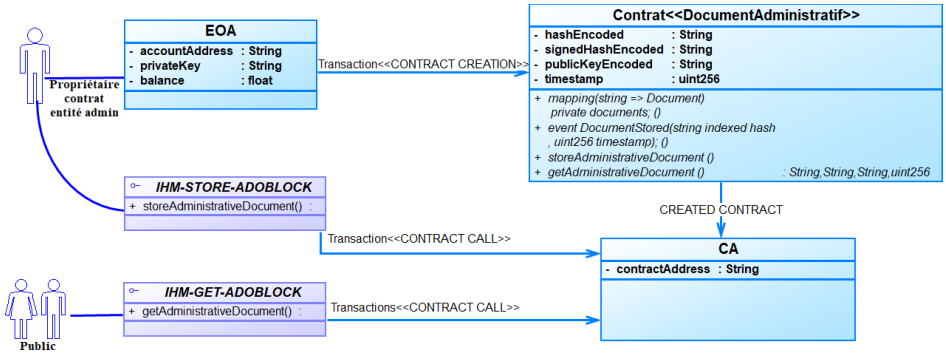


Figure 17 *: Diagramme d'interactions entre le contrat intelligent et les comptes*

[9] Idem au [8].

5.1.2. Outils et technologies utilisés

Plusieurs outils et technologies ont été utilisés pour concevoir et implémenter la DApp ADOBLOCK. Les principaux sont présentés comme suit :

- **Sybase PowerDesigner ® 16.5.0.3982**

est un logiciel propriétaire de conception qui offre la possibilité de modéliser les traitements informatiques. Il a été créé par SDP puis racheté par Sybase en 1995, qui lui-même appartient à SAP. Cet outil très populaire et intuitif permet également de travailler avec UML et la méthode Merise. Nous avons utilisé PowerDesigner (anciennement PowerAMC) pour modéliser les différents diagrammes UML, tandis que l'approche a été schématisée à l'aide du logiciel multiplateforme **draw.io** accessible sur https://app.diagrams.net.

- **Spring-boot 3.4.1 (Maven 3.8.1 + Java 17)**

est un framework open source populaire sous Licence Apache 2.0, connu pour sa robustesse. Publié en avril 2014 par *Rod Johnson*, Spring boot prend en compte des serveurs d'application web Tomcat (pour notre cas) et Jetty. Il peut être utilisé pour implémenter des microservices, des applications web et des applications console avec le langage Java (mais prend en charge également Kotlin et Apache Groovy). Nous l'utilisons pour l'implémentation du backend de la DApp car il ne nécessite pas de configuration particulière, gère automatiquement les dépendances et fournit plusieurs fonctionnalités prêtes pour la production.

- **Angular 15 (HTML5, TypeScript/JavaScript, SCSS/CSS)**

est un framework JavaScript frontend open source crée en septembre 2016 sous licence MIT. Angular recommande l'utilisation du langage TypeScript avec le SCSS par rapport au JavaScript et CSS. Il permet de créer des applications dynamiques complètes en monopage (SPA – pour de belles interfaces UX intuitives) telle que les front-office et back-office de notre DApp ADOBLOCK sans avoir à réécrire toute la structure du projet de bout en bout. Pour son utilisation, nous avons recouru à NPM (node package manager) v11.3.0 pour la gestion des packages officiels de Node.js (v22.14.0 pour notre cas).

- **Solidity (^0.5.16 et ^0.8.28)**

est un langage de programmation apparu pour la première fois en août 2014 sous licence GNU GPL v3.0. Il a été proposé par *Gavin Wood* pour permettre d'implémenter des contrats intelligents compatibles avec essentiellement la blockchain Ethereum (EVM précisément). Solidity est statiquement typé et prend en compte les héritages multiples (voire complexes), les mappages, les structures imaginaires. Il est Turing complet et permet aux contrats intelligents d'avoir de nombreuses propriétés de sécurité. Nous utilisons la version Solidity ^0.5.16 pour nos premières expérimentations avec les outils tels que Ganache et Truffle en console et sur Visual Studio Code. La suite des expérimentations se base sur Solidity ^0.8.28 avec les outils comme Remix – Ethereum IDE (en ligne), Visual Studio Code et Hardhat. Cela en raison de la différence des versions de compilateur de Truffle, Remix, et Hardhat.

- **Remix – Ethereum IDE**

est un environnement de développement intégré (IDE) de contrats intelligents en Solidity. [26] Il est utilisé par des utilisateurs de tous niveaux car ne nécessite aucune configuration, favorise un cycle de développement rapide et propose un riche ensemble de plugins avec des interfaces utilisateur intuitives. Cet IDE est disponible en deux versions (application web que nous utilisons ou application de bureau). Il intègre une quinzaine de comptes Ethereum (EOA) avec chacun 100 ETH fictifs, plusieurs versions de compilateurs, plusieurs versions d'EVM pour les déploiements, les débogages et les tests des contrats intelligents.

- **Visual Studio Code (VS Code) 1.96.2**

est également un IDE multiplateforme de codes sources produit par Microsoft à partir de novembre 2015 sous licence MIT et licence propriétaire. Il est open source et supporte plusieurs langages de programmations dont JavaScript, TypeScript, Java et même Solidity (via des plugins/extensions à installer). Il est majoritairement utilisé pour l'implémentation de la DApp ADOBLOCK.

- **Ganache 2.7.1**

[44] est un outil multiplateforme conçu par Truffle Suite (boîte à outils de développement) dans le but de simuler sur un ordinateur en local, le fonctionnement d'un réseau blockchain public tel qu'Ethereum. En effet, il offre la possibilité d'avoir une chaîne de blocs Ethereum locale pour le développement, et permet de déployer, d'interagir et

de tester des DApps et contrats intelligents dans un environnement sécurisé et contrôlé. Ganache est largement utilisé par les développeurs de blockchain Ethereum pour sa flexibilité et sa capacité à offrir des Ethers illimités et fictifs (aucun frais réel de gaz). Il s'intègre directement avec Truffle. Nous utilisons Ganache pour tester notre contrat intelligent afin de prétendre passer à un réseau Ethereum (réel en production) en direct.

- **Truffle v5.11.5**

est un outil très complet par rapport à Remix – Ethereum IDE. Le framework Truffle intègre automatiquement Ganache (ganache-cli v7.9.1 en mode console pour notre cas). Truffle et Ganache sont donc étroitement liés, ce qui facilite l'exécution de tests automatiques pour les contrats intelligents. En effet, la suite Truffle permet de générer facilement en ligne de commande une DApp avec le framework JavaScript React (le plus souvent). Ce genre de projet prédispose un package pour les futurs contrats intelligents, un package contenant le script qui permet la migration des contrats implémentés sur la blockchain de choix, et autre un package pour les futurs scripts de tests des contrats. Un fichier de configurations est également associé à la racine de la DApp générée afin de paramétrer les blockchains sur lesquelles déployer ($truffle migrate) les contrats et le compilateur utilisé ($truffle compile). Nous l'avons utilisé en ligne de commande lors des premières expérimentations.

- **MetaMask v2.132.0 (x)**

est un portefeuille de crypto-monnaie (Wallet) qui permet aux utilisateurs d'interagir avec des applications décentralisées. Il a été développé en 2016 par ConsenSys Software Inc. en deux versions à savoir : une extension web intégrable aux navigateurs Google Chrome, Mozilla, etc. et une application mobile (iOS et Android). A travers MetaMask, les utilisateurs peuvent y stocker et gérer les clés de leurs comptes. Il permet aussi aux utilisateurs de demander et de signer des transactions. Dans nos expérimentations, nous utilisons MetaMask[10] pour le fait qu'il est très pratique pour tester les DApps qui sont conçues sur la base des librairies Hardhat et Ethers.js comme dans notre cas.

- **Technologies fondamentales**

Aux nombres des technologies, nous avons expérimenté dans le frontend de ADOBLOCK, plusieurs modules de la bibliothèque **Web3** (ou Web3.js) pour récupérer le compte Ethereum local

[10] Services disponibles sur : https://metamask.io/

connecté (via Ganache) et s'en servir de Provider à l'Application Binary Interface (ABI) du contrat intelligent. Cette récupération est donc une interaction avec les nœuds Ethereum locaux (possible avec ceux distants) à l'aide des protocoles HTTP. Ensuite nous avons utilisé, en remplacement de Web3, le framework **Hardhat ^2.24.0** et la librairie **Ethers.js ^6.14.0** (ainsi que leurs sous modules hardhat-waffle, ethereum-waffle, chai, etc.) qui nous ont permis de disposer de vingt (20) de comptes Ethereum (importés dans MetaMask) contenant chacun 10000 ETH fictifs. Avec ce couple Hardhat et Ethers.js, nous pouvons configurer le réseau blockchain de test (network Sepolia – test online, network Localhost, …) afin d'y compiler et déployer le contrat intelligent, de connecter le contrat intelligent à Angular via l'**ABI** et de connecter également MetaMask à Angular. La communication entre MetaMask et Angular se fait via le protocole JSON Remote Procedure Call (**JSON RPC**).

L'**API REST** permet au frontend Angular de ADOBLOCK de faire appel aux ressources (clés cryptographiques, données calculées, etc.) du backend Spring-boot de ADOBLOCK qui sont aussi transmises en json.

L'algorithme de hachage sécurisé **SHA-256** est utilisé dans le backend pour calculer l'empreinte numérique du contenu de document administratif. Et afin de générer la paire de **clés cryptographiques ECDSA**, nous utilisons la dépendance Maven **bouncycastle** ; ce qui nous permet d'exploiter la **courbe elliptique spec256r1 (P-256)** pour les signatures numériques des empreintes numériques.

5.1.3. Démarche d'implémentation

La démarche pratique pour l'implémentation de la DApp ADOBLOCK est la suivante :

a. Conception et modélisation des différentes spécifications fonctionnelles et techniques.
b. Implémentation du backend Spring Boot (APP-ADOBLOCK) : principalement pour calculer les empreintes numériques, les clés cryptographiques, les signatures numériques, etc. côté serveur. Voir la structure du code source à l'annexe 4 (a) ci-dessous.
c. Implémentation du frontend Angular (UX-ADOBLOCK) : pour disposer d'interfaces clients (utilisateurs) basées globalement sur les modules PrimeNG afin de garantir une haute responsivité et convivialité. Voir la structure du code à l'annexe 4 (b) ci-dessous.
d. Implémentation, compilation, déploiement et tests rapides du contrat intelligent *DocumentAdministratif* sur Remix – Ethereum IDE. Voir le code du contrat à l'annexe 3.

e. Génération et configuration du projet basic JavaScript Hardhat dans le frontend Angular : pour la compilation et le déploiement du contrat intelligent. Les scripts, le contrat, les fichiers de configuration et les autres fichiers auto-générés de ce projet basic sont logés dans le package « *blockchain* » du frontend Angular créé à cet effet. Voir la structure du code à l'annexe 4 (b) ci-dessous.
f. Installation des librairies Ethereum nécessaires dont Ethers.js dans le frontend Angular : pour l'intégration du frontend Angular avec Ethereum.
g. Installation et configuration de MetaMask : pour la signature des transactions et la gestion des comptes Ethereum importés depuis Hardhat.
h. Tests et ajustements de la solution ADOBLOCK.

L'application de cette démarche a permis d'aboutir à la solution ADOBLOCK dont les différentes interfaces sont présentées dans la section ci-après.

5.2. Présentation de la solution

Notre solution ADOBLOCK est une DApp hybride semblable à une application web classique, accessible à travers un navigateur. Le frontend de ADOBLOCK est composé de deux (02) espaces à savoir le back-office (accès réservé aux entités administratives) et le front-office (accès grand public). Les principales interfaces utilisateur par espace sont ci-dessous présentée.

- **En back-office**

Nous avons principalement un menu latéral composé d'items « **Accueil** », « **Guide utilisation** », « **Obtenir clés privée/publique** », « **Enregistrer document** », et « **Authentifier document** ». Les interfaces des trois (03) derniers items sont illustrées par les figures 18, 19 et 20 ci-dessous.

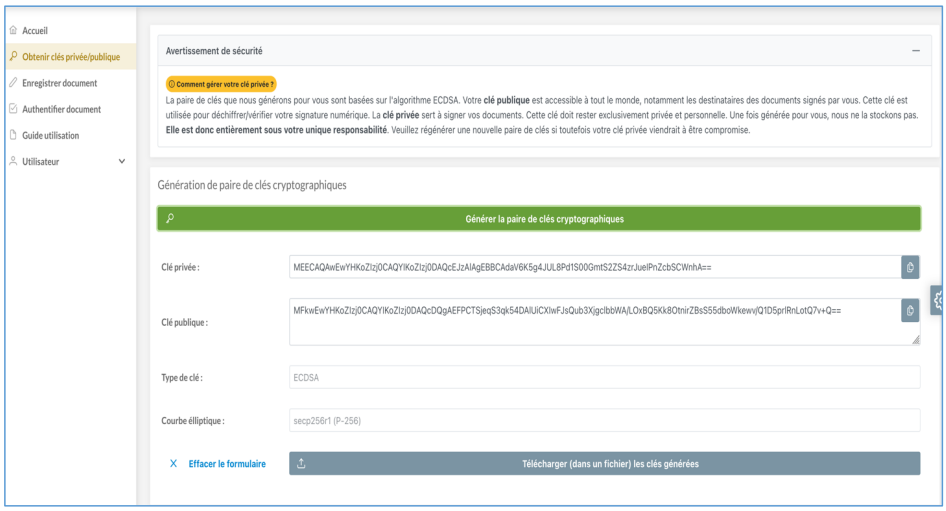


Figure 18 *: Interface du menu « Obtenir clés privée/publique »*

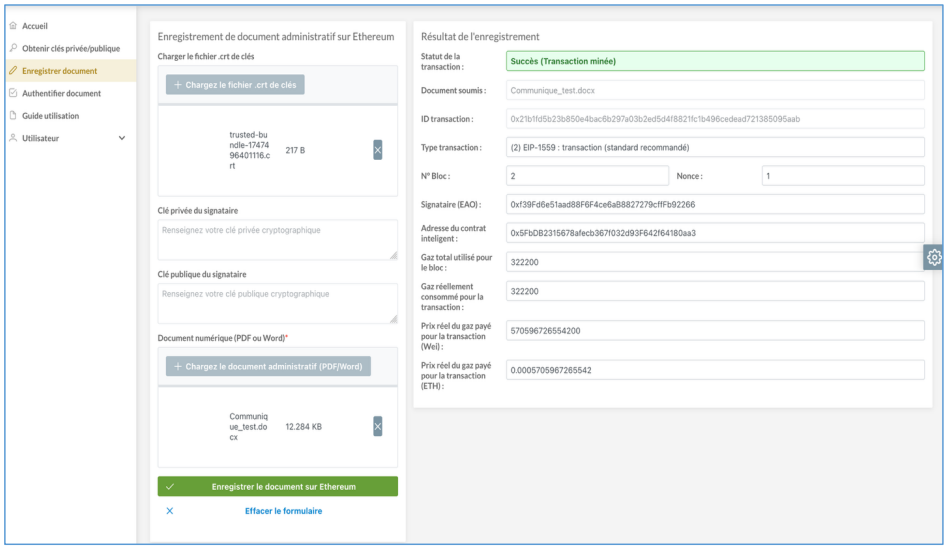


Figure 19 *: Interface du menu « Enregistrer document »*

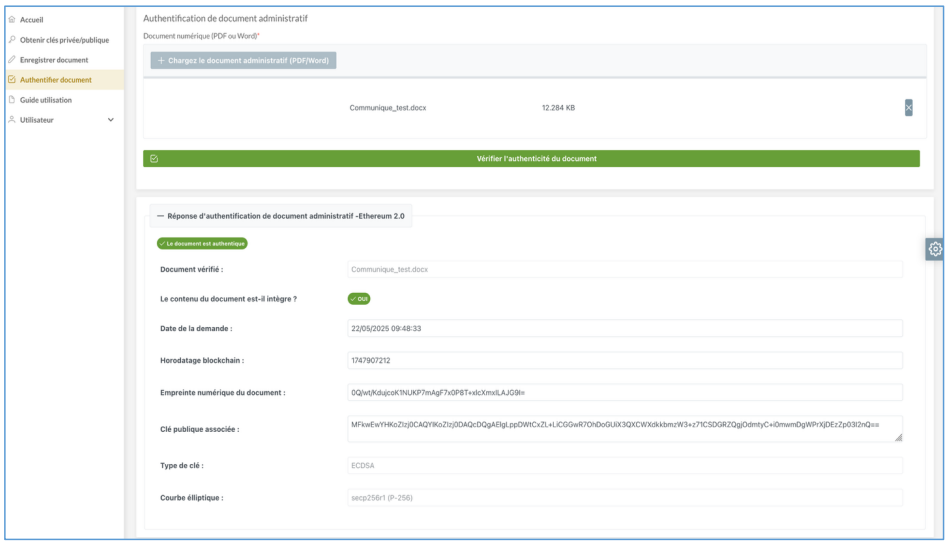


Figure 20 *: Interface du menu « Authentifier document »*

En front-office

Nous avons principalement un menu horizontal supérieur composé d'items « **Accueil** », « **Guide** », et « **A propos** ». L'interface le plus important est sur l'item « **Accueil** », illustré à travers la figure 21 ci-dessous. Il suffit d'y importer un document électronique et de soumettre sa vérification.

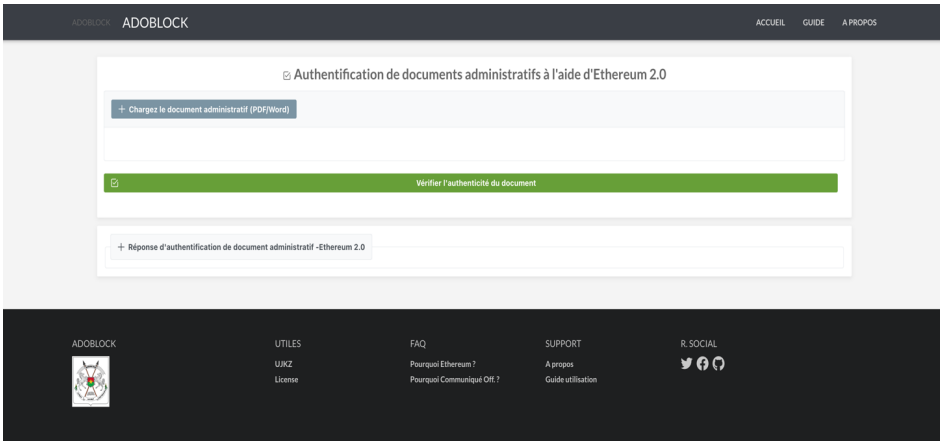


Figure 21 *: Interface grand public d'authentification de documents administratifs*

5.3. Discussion des résultats

Les résultats obtenus à l'issu de nos recherches permettent d'authentifier les documents administratifs électroniques, particulièrement les communiqués officiels, en utilisant la technologie Blockchain. L'accent mis sur les communiqués officiels découle du contexte de cette étude qui s'articule autour de l'authenticité ou la fiabilité des informations mises à la disposition du public par les structures administratives.

Concrètement, la DApp ADOBLOCK offre la possibilité de vérifier en temps réel, l'authenticité de documents administratifs électroniques. Aussi, elle intègre absolument la blockchain publique Ethereum 2.0 afin de renforcer la sécurité et l'intégrité des communiqués officiels. En effet, Ethereum introduit régulièrement de nombreuses évolutions comme c'est le cas de sa mise à jour du 7 mai 2025, via le *hard fork Pectra* ; ce qui rend ainsi le réseau plus sécurisé, accessible et convivial.

En se référant aux travaux antérieurs [4] et [21] en matière d'authentification et de sécurisation de documents à l'aide de la blockchain, ADOBLOCK est une innovation en raison de :

- l'absence d'utilisation de base de données ordinaires, contrairement à l'approche de [4] ;
- la simplicité du formulaire grand public d'authentification de documents (seul le fichier est requis au lieu de plusieurs champs comme dans [21]) ;
- la non manipulation directe des clés cryptographiques par les acteurs grand public ;
- la non nécessité d'avoir plusieurs comptes EOA ;
- l'utilisation de technologies plus avancées et flexibles telles que Hardhat, Ethers.js au lieu de Web3.js ;
- l'intégration de l'horodatage électronique, de la signature numérique des documents systématiquement hachés, et de la blockchain Ethereum via le contrat intelligent. Cela permet de prendre en compte plusieurs méthodes d'authentification afin de garantir une sécurité accrue de notre solution.

En dépit de ces avantages, une préoccupation résiduelle nécessite d'être examinée. Il s'agit de la prise en compte des documents administratifs électroniques non textuels (fichier PDF dont le contenu est un scan en image). En effet, il est évident que pour un document dont le contenu extrait est du texte, plusieurs opérations de hachage SHA-256 de ce contenu aboutiront à un hash identique

(la même empreinte numérique). Cependant, pour un document ayant fait l'objet de numérisations, les résultats de plusieurs opérations de hachage seraient probablement différents. Car les images scannées (voire avec le même appareil de scannage) ont des variations en termes de résolution, d'alignement, de luminosité, de compressions, etc. et contiennent des métadonnées (date de création, logiciel utilisé, etc.) différentes. Or dans un système d'authentification de documents et selon les propriétés des algorithmes de hachage, le même document doit nécessairement avoir la même valeur de hachage. Ainsi, la perspective sera d'explorer l'intégration de système de Reconnaissance Optique de Caractères (OCR ou autres pipelines) ou l'option de hachage direct du document, c'est-à-dire le fichier binaire complet (à défaut).

En somme, nous avons abordé dans ce dernier chapitre, la présentation du protocole d'implémentation de la DApp ADOBLOCK. Ce protocole est constitué des éléments d'analyse et de conception de l'approche précédemment proposé, des outils et technologies utilisés lors de nos expérimentations, et la démarche adoptée pour l'implémentation de notre DApp. Nous avons également présenté les principales interfaces (écrans) de la DApp puis mené une discussion sur l'ensemble des résultats obtenus à l'issus de nos recherches. Que peut-on retenir, en définitive, de nos recherches ?

CONCLUSION ET PERSPECTIVES

CONCLUSION ET PERSPECTIVES

En résumé, ce travail était axé principalement sur trois (03) éléments : les concepts de base en matière de blockchain, l'état de l'art sur l'authentification et la sécurisation de documents à l'aide de la technologie blockchain, et une approche scientifique proposée et implémentée par nous. Cela afin de répondre à la préoccupation initiale à savoir : « comment la technologie blockchain peut-elle garantir l'authenticité des documents administratifs de façon plus sécurisée, transparente et efficace ? ».

Ainsi, nos recherches ont été particulièrement orientées sur l'intégration de la blockchain Ethereum dans une solution d'authentification de documents administratifs, notamment les communiqués officiels, afin de renforcer leur sécurité et l'intégrité de leur contenu. Notre solution ADOBLOCK est une application décentralisée qui combine à Ethereum 2.0, l'utilisation d'un contrat intelligent, de la signature numérique en s'appuyant sur la technique à clés asymétriques ECDSA avec une courbe elliptique secp256r1, de l'horodatage électronique, et de la technique de hachage basé sur l'algorithme SHA-256.

A ce stade, nous pouvons affirmer que la DApp ADOBLOCK satisfait nos hypothèses de recherches car l'intégration des technologies suscité permettent de garantir plus de sécurité et d'intégrité des documents administratifs. Ce qui permet d'aboutir à une diminution considérable des cas de falsifications/fraudes des documents administratifs. Aussi, l'expérimentation de la solution nous permet d'obtenir des résultats instantanés (en temps réel) sur l'authenticité des exemples de documents administratifs électroniques aux contenus textuels.

En perspective, nous prévoyons d'abord d'implémenter le module de gestion des utilisateurs du back-office de la DApp de sorte à ce que chaque compte d'accès à la DApp soit systématiquement associé à un EOA. Ensuite, nous effectuerons la mise en échelle de la DApp sur un réseau public blockchain de tests plus réaliste tel que Testnets (Goerli, Sepolia, ou autre) avant d'envisager sa mise production sur le réseau public Ethereum Mainnet (coûteux en frais de gaz, mais sécurisé). Cela garantira que la solution soit totalement décentralisée. Cependant, une question résiduelle demeure : comment prendre en charge dans notre approche, les documents administratifs électroniques non textuels (fichier PDF dont le contenu est un scan en image) ? sachant que les images scannées (voire avec le même appareil de scannage) ont des variations en termes de résolution, d'alignement, de luminosité, de compressions, etc. et contiennent des métadonnées (date de création, logiciel utilisé, etc.) différentes.

RÉFÉRENCES

[1] B. Anders, « Blockchain Demo ». Consulté le: 23 janvier 2025. [En ligne]. Disponible sur: https://andersbrownworth.com/blockchain/hash

[2] H. Anwar, « Blockchain Consortium: Top 20 Consortia You Should Check Out », *101 Blockchains*, 8 février 2021. Consulté le: 2 octobre 2024. [En ligne]. Disponible sur: https://101blockchains.com/blockchain-consortium/

[3] A. Back *et al.*, « Enabling Blockchain Innovations with Pegged Sidechains », p. 25, oct. 2014.

[4] A. Bakhoum, « La Blockchain pour la Sécurisation des E-livrets scolaires », 2019. Consulté le: 10 septembre 2024. [En ligne]. Disponible sur: http://rivieresdusud.uasz.sn/xmlui/handle/123456789/1803

[5] I. Bashir, *Mastering Blockchain*. Packt Publishing Ltd, 2017.

[6] D. Bayer, S. Haber, et W. S. Stornetta, « Improving the Efficiency and Reliability of Digital Time-Stamping », in *Sequences II*, R. Capocelli, A. De Santis, et U. Vaccaro, Éd., New York, NY: Springer, 1993, p. 329-334. doi: 10.1007/978-1-4613-9323-8_24.

[7] B. Bertrand, *Loi 051 portant sur l'accès à l'information publique - CSC - BURKINA FASO*. Consulté le: 16 septembre 2024. [En ligne]. Disponible sur: https://www.csc.bf/index.php/textes-de-reference/lois/item/76-loi-051-portant-sur-l-acces-a-l-information-publique

[8] R. Blancher, « Horodatage électronique : définition et fonctionnement », Evidency. Consulté le: 24 février 2025. [En ligne]. Disponible sur: https://evidency.io/horodatage-electronique-definition-et-fonctionnement/

[9] T. Bourbotte, « C'est quoi Ethereum ? Nos explications pour tout savoir sur cette blockchain et sa cryptomonnaie ETH », *Cryptoast*, 20 décembre 2024. Consulté le: 10 février 2025. [En ligne]. Disponible sur: https://cryptoast.fr/fiche-ethereum/

[10] C. de la sécurité des télécommunications Canada, « Conseils sur la configuration sécurisée des protocoles réseau (ITSP.40.062) », *Centre canadien pour la cybersécurité*, 15 octobre 2020. Consulté le: 18 avril 2025. [En ligne]. Disponible sur: https://www.cyber.gc.ca/fr/orientation/conseils-sur-la-configuration-securisee-des-protocoles-reseau-itsp40062

[11] DCRP/MAECR-BE, « Ministère des Affaires Etrangères du Burkina Faso ». Consulté le: 15 septembre 2024. [En ligne]. Disponible sur: https://www.facebook.com/share/p/yqFV712VwsjcHxne/?mibextid=oFDknk

[12] DCRP/MFPTPS, « Ministère de la Fonction Publique, du Travail et de la Protection Sociale ». Consulté le: 15 septembre 2024. [En ligne]. Disponible sur: https://www.facebook.com/share/p/kHPydXy6A1zMniK1/?mibextid=oFDknk

[13] Direction générale de l'INSD, « Démenti de recrutement d'étudiants ». Consulté le: 15 septembre 2024. [En ligne]. Disponible sur: https://www.facebook.com/share/p/DYayYQThHP2NLDbF/?mibextid=oFDknk

[14] Douanes du Burkina Faso, « Service de Communication et des Relations Publiques de la Direction Générale des Douanes ». Consulté le: 15 septembre 2024. [En ligne]. Disponible sur: https://www.facebook.com/share/p/15f4cKVZ3P/

[15] É. (gapz) Gaspar, « Standardisation des courbes elliptiques : à qui faire confiance ? | Connect - Editions Diamond », avril 2016. Consulté le: 18 avril 2025. [En ligne]. Disponible sur: https://connect.ed-diamond.com/MISC/mischs-013/standardisation-des-courbes-elliptiques-a-qui-faire-confiance

[16] Ferr. Godebarge et R. Rossat, « Principes clés d'une application Blockchain », déc. 2016. Consulté le: 11 septembre 2024. [En ligne]. Disponible sur:

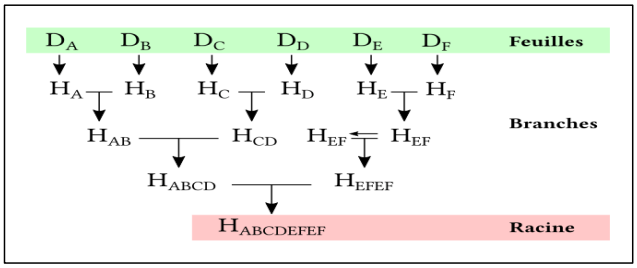
https://www.academia.edu/download/56098741/GODEBARGE_ROSSAT_Blockchain-version-finale.pdf
[17] S. Haber et W. S. Stornetta, « How to time-stamp a digital document », *J. Cryptol.*, vol. 3, nº 2, p. 99-111, janv. 1991, doi: 10.1007/BF00196791.
[18] J.-L. Jonnaert, « Quelles différences entre blockchain publique et blockchain privée ? », *Cryptoast*, 12 juin 2023. Consulté le: 29 septembre 2024. [En ligne]. Disponible sur: https://cryptoast.fr/differences-blockchain-publique-blockchain-privee/
[19] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur, et H.-N. Lee, « Systematic Review of Security Vulnerabilities in Ethereum Blockchain Smart Contract », *IEEE Access*, vol. 10, p. 6605-6621, 2022, doi: 10.1109/ACCESS.2021.3140091.
[20] É. Larousse, « Définitions : consensus - Dictionnaire de français Larousse ». Consulté le: 15 septembre 2024. [En ligne]. Disponible sur: https://www.larousse.fr/dictionnaires/francais/consensus/18357
[21] I. Meirobie, A. P. Irawan, H. T. Sukmana, D. P. Lazirkha, et N. P. L. Santoso, « Framework Authentication e-document using Blockchain Technology on the Government system », *Int. J. Artif. Intell. Res.*, vol. 6, nº 2, Art. nº 2, déc. 2022, doi: 10.29099/ijair.v6i2.294.
[22] Ministère de l'Economie et des Finances du Burkina Faso, « ALERTE VIGILANCE ». Consulté le: 26 janvier 2025. [En ligne]. Disponible sur: https://www.facebook.com/share/p/1BP46UYXF9/
[23] S. Nakamoto, « Bitcoin: A Peer-to-Peer Electronic Cash System », p. 9.
[24] F. Num, « La signature électronique : un outil devenu incontournable - francenum.gouv.fr », Direction générale des entreprises, 7 décembre 2020. Consulté le: 21 février 2025. [En ligne]. Disponible sur: https://www.francenum.gouv.fr/guides-et-conseils/pilotage-de-lentreprise/dematerialisation-des-documents/la-signature
[25] A. A. Oussama, « CHAPITRE III : État de l'art de la Blockchain », in *ResearchGate*, 2019. Consulté le: 21 octobre 2024. [En ligne]. Disponible sur: https://www.researchgate.net/publication/335174496_CHAPITRE_III_Etat_de_l'art_de_la_Blockchain
[26] Remix, « Bienvenue dans la documentation de Remix ! — Documentation de Remix - Ethereum IDE 1 ». Consulté le: 28 avril 2025. [En ligne]. Disponible sur: https://remix-ide.readthedocs.io/en/latest/
[27] R. Subramanian et T. Chino, « The State of Cryptocurrencies, Their Issues and Policy Interactions », *J. Int. Technol. Inf. Manag.*, vol. 24, nº 3, p. 40, janv. 2015, doi: 10.58729/1941-6679.1045.
[28] T. Takenobu, « Ethereum EVM illustrated », p. 116.
[29] « Blockchain : qu'est-ce qu'un Smart Contract et à quoi ça sert ? », *LeMagIT*, 23 novembre 2016. Consulté le: 5 février 2025. [En ligne]. Disponible sur: https://www.lemagit.fr/conseil/Blockchain-quest-ce-quun-Smart-Contract-et-a-quoi-ca-sert
[30] « La première blockchain de l'histoire date de 1995, et elle est imprimée sur papier », *Le Monde.fr*, 1 septembre 2018. Consulté le: 17 septembre 2024. [En ligne]. Disponible sur: https://www.lemonde.fr/big-browser/article/2018/09/01/la-premiere-blockchain-de-l-histoire-date-de-1995-et-elle-est-imprimee-sur-papier_5349082_4832693.html
[31] « QR vs. RFID, which is better? Find out which asset tags to use. », itemit. Consulté le: 25 février 2025. [En ligne]. Disponible sur: https://itemit.com/qr-vs-rfid-which-is-better/
[32] « Qu'est-ce qu'un arbre Merkle? », *Bit2Me Academy*, 11 novembre 2019. Consulté le: 18 décembre 2024. [En ligne]. Disponible sur: https://academy.bit2me.com/fr/qu%27est-ce-qu%27un-arbre-merkle/
[33] « Qu'est-ce que le consensus ? Guide du débutant », *Qu'est-ce que le consensus ? Guide du débutant*, 13 mai 2022. Consulté le: 15 septembre 2024. [En ligne]. Disponible sur: https://crypto.com/fr/university/consensus-mechanisms-explained

[34] « How to Agree: Different Types of Consensus for Blockchain », How to Agree: Different Types of Consensus for Blockchain. Consulté le: 1 février 2025. [En ligne]. Disponible sur: https://crypto.com/en/university/different-types-of-consensus-for-blockchain
[35] « AXEM Technology | Quelle est la différence entre la RFID et le code QR ? Quelle est la meilleure solution pour votre activité ? », *https://www.axemtec.com/*, 7 décembre 2022. Consulté le: 25 février 2025. [En ligne]. Disponible sur: https://www.axemtec.com/rfid-vs-qr-code-whats-the-difference-which-one-is-better-for-your-business/
[36] « Hashing : voici comment fonctionne le hachage », *IONOS Digital Guide*, 22 février 2023. Consulté le: 24 février 2025. [En ligne]. Disponible sur: https://www.ionos.fr/digitalguide/sites-internet/developpement-web/hachage/
[37] « Smart contract : définition et fonctionnement ». Consulté le: 7 février 2025. [En ligne]. Disponible sur: https://www.captaincontrat.com/contrats-commerciaux-cgv/contrats-commerciaux/smart-contract-definition-et-fonctionnement-me-beaubourg-avocats
[38] « Éléments Fondamentaux d'un Bloc dans la Blockchain - W3r.one Magazine », 9 février 2024. Consulté le: 21 octobre 2024. [En ligne]. Disponible sur: https://w3r.one/fr/blog/blockchain-web3/architecture-blockchain/conception-de-blocs/elements-fondamentaux-bloc-blockchain
[39] « Introduction aux dApps », *ethereum.org*, 23 avril 2024. Consulté le: 12 février 2025. [En ligne]. Disponible sur: https://ethereum.org/fr/developers/docs/dapps/
[40] « Blocs », *ethereum.org*, 26 mai 2024. Consulté le: 17 février 2025. [En ligne]. Disponible sur: https://ethereum.org/fr/developers/docs/blocks/
[41] « Consensus : Définition simple et facile du dictionnaire ». 9 juin 2024. Consulté le: 15 septembre 2024. [En ligne]. Disponible sur: https://www.linternaute.fr/dictionnaire/fr/definition/consensus/
[42] « Comptes Ethereum », *ethereum.org*, 4 juillet 2024. Consulté le: 12 février 2025. [En ligne]. Disponible sur: https://ethereum.org/fr/developers/docs/accounts/
[43] « Transactions », *ethereum.org*, 11 août 2024. Consulté le: 15 février 2025. [En ligne]. Disponible sur: https://ethereum.org/fr/developers/docs/transactions/
[44] « Ganache : Simuler un Environnement Blockchain pour le Développement - W3r.one Magazine », 11 septembre 2024. Consulté le: 11 mars 2025. [En ligne]. Disponible sur: https://w3r.one/fr/blog/blockchain-web3/smart-contracts/outils-environnements-developpement/ganache-simuler-environnement-blockchain-developpement
[45] « Nœuds et clients », *ethereum.org*, 9 octobre 2024. Consulté le: 12 février 2025. [En ligne]. Disponible sur: https://ethereum.org/fr/developers/docs/nodes-and-clients/
[46] « Signature numérique », *Wikipédia*. 21 octobre 2024. Consulté le: 21 février 2025. [En ligne]. Disponible sur: https://fr.wikipedia.org/wiki/Signature_num%C3%A9rique
[47] « Que sont les contrats intelligents ? | Les contrats intelligents expliqués | Kraken ». Consulté le: 8 février 2025. [En ligne]. Disponible sur: https://www.kraken.com/fr/learn/what-are-smart-contracts
[48] « Machine virtuelle Ethereum (EVM) », *ethereum.org*, 13 décembre 2024. Consulté le: 12 février 2025. [En ligne]. Disponible sur: https://ethereum.org/fr/developers/docs/evm/
[49] « Introduction à l'ether », *ethereum.org*, 13 décembre 2024. Consulté le: 12 février 2025. [En ligne]. Disponible sur: https://ethereum.org/fr/developers/docs/intro-to-ether/
[50] « Que sont les contrats intelligents sur la blockchain ? | IBM », 26 décembre 2024. Consulté le: 7 février 2025. [En ligne]. Disponible sur: https://www.ibm.com/fr-fr/topics/smart-contracts
[51] « Blockchain », *Wikipédia*. 11 janvier 2025. Consulté le: 19 septembre 2024. [En ligne]. Disponible sur: https://fr.wikipedia.org/w/index.php?title=Blockchain&oldid=222001944#Histoire
[52] « Gaz et frais », *ethereum.org*, 14 janvier 2025. Consulté le: 15 février 2025. [En ligne]. Disponible sur: https://ethereum.org/fr/developers/docs/gas/

[53] « Ethereum », *Wikipédia*. 22 janvier 2025. Consulté le: 10 février 2025. [En ligne]. Disponible sur: https://fr.wikipedia.org/wiki/Ethereum
[54] « Qu'est-ce qu'une attaque à 51% et quels sont les risques ? », *coinbase*. Consulté le: 3 février 2025. [En ligne]. Disponible sur: https://www.coinbase.com/fr/learn/crypto-glossary/what-is-a-51-percent-attack-and-what-are-the-risks
[55] « Introduction aux contrats intelligents », *ethereum.org*. Consulté le: 12 février 2025. [En ligne]. Disponible sur: https://ethereum.org/fr/developers/docs/smart-contracts/
[56] « ISO/TC 307 - Blockchain and distributed ledger technologies ». Consulté le: 31 octobre 2024. [En ligne]. Disponible sur: https://www.iso.org/committee/6266604/x/catalogue/p/1/u/1/w/0/d/0

ANNEXES

Annexe 1 : exemple d'arbre de Merkle dans Bitcoin



Source : *https://cryptoast.fr/wp-content/uploads/2020/08/merkle-tree-general.png*

Annexe 2 : détail de réponse d'une transaction sur un réseau de tests Hardhat

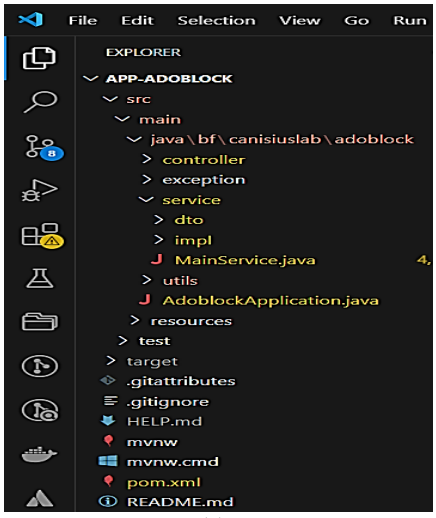
```
ContractTransactionResponse {
  provider: HardhatEthersProvider {
    _hardhatProvider: LazyInitializationProviderAdapter {
      _providerFactory: [AsyncFunction (anonymous)],
      _emitter: [EventEmitter],
      _initializingPromise: [Promise],
      provider: [BackwardsCompatibilityProviderAdapter]
    },
    _networkName: 'hardhat',
    _blockListeners: [],
    _transactionHashListeners: Map(0) {},
    _eventListeners: []
  },
  blockNumber: 2,
  blockHash: '0xdf964d8e9a46fba14634b1e7710a6b6f8d74cbcfcb5cfbb96a1a6641a21f9d7b',
  index: undefined,
  hash: '0x29b948b7c060c5080d8c1246d691c92003c4e3a34a829a6379ac68110bdde284',
  type: 2,
  to: '0x5FbDB2315678afecb367f032d93F642f64180aa3',
  from: '0xf39Fd6e51aad88F6F4ce6aB8827279cfffb92266',
  nonce: 1,
  gasLimit: 30000000n,
  gasPrice: 1766615449n,
  maxPriorityFeePerGas: 1000000000n,
  maxFeePerGas: 2533230898n,
  maxFeePerBlobGas: null,
  data: '0x8ec0423200000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000006',
  value: 0n,
  chainId: 1337n,
  signature: Signature { r: "0x521ba6a86697dc5854fd850264684c591e956456fdd5c5ba79f9354211fa6faa", s:
"0x62f4e1f332e7337c5d0ffb7aed7f6b4d4fda953a2e6452cc9d1bc0e34def5087", yParity: 1, networkV: null },
  accessList: [],
  blobVersionedHashes: null
}
```

Annexe 3 : source du contrat intelligent « DocumentAdministratif »

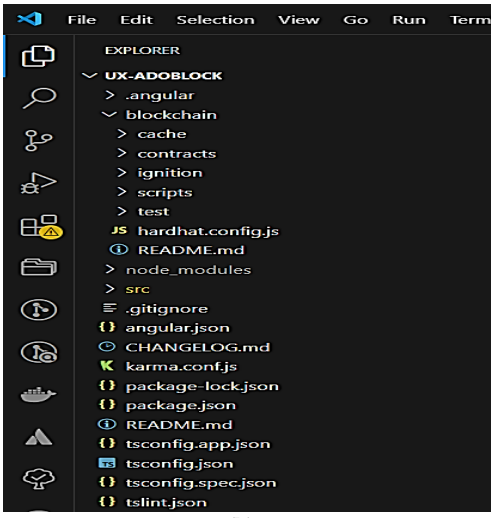
Home DocumentAdministratif.sol

```sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.28;

import "hardhat/console.sol";
/**
 * @author canisiushien@gmail.com
 *
 * @title CONTRAT INTELLIGENT "DocumentAdministratif".
 * @dev Stocke et recherche un document administratif (hash) dans Ethereum.
 * @dev Ce contrat est une intéraction entre deux parties : le document administratif et l'utilisateur.
 * @dev date-implCompilDeployAndTest: 10 mars 2025.
 */
contract DocumentAdministratif {
    //definition de la structure d'un document a stocker
    struct Document {
        string hashEncoded; //hash du document
        string signedHashEncoded; //signature numerique du document
        string publicKeyEncoded; //cle publique du signataire
        uint256 timestamp; //date et heure(horodatage) de stockage du document dans la blockchain
    }

    //mappage(appele "documents") de la structure du document avec une chaine (le hash)
    mapping(string => Document) private documents;

    //creation d'un evenement d'indexage (tel hash+horodatage correspond a tel ensemble de donnees w = signedHashEncoded+publicKeyEncoded)
    event DocumentStored(string indexed hashEncoded, uint256 timestamp);

    //fonction de stockage de document dans la blockchain
    function storeAdministrativeDocument(string memory _hashEncoded, string memory _signedHashEncoded, string memory _publicKeyEncoded) public {
        //controle de doublons de hash dans le stockage
        require(bytes(documents[_hashEncoded].hashEncoded).length == 0, "Ce document existe deja dans la blockchain.");

        //stocke le document dans une map
        documents[_hashEncoded] = Document({
            hashEncoded: _hashEncoded,
            signedHashEncoded: _signedHashEncoded,
            publicKeyEncoded: _publicKeyEncoded,
            timestamp: block.timestamp
        });

        //emettre l'evenement a l'attention des utilisateurs
        emit DocumentStored(_hashEncoded, block.timestamp);
    }

    //fonction de recherche/recuperation de document(hash) dans la blockchain
    function getAdministrativeDocument(string memory _hashEncoded) public view returns (string memory, string memory, string memory, uint256){
        //on leve une exception si un hash similaire n'avait pas ete stocké dans la blockchain
        require(bytes(documents[_hashEncoded].hashEncoded).length != 0, "Ce document n'existe pas dans la blockchain.");

        Document memory doc = documents[_hashEncoded];
        return (doc.hashEncoded, doc.signedHashEncoded, doc.publicKeyEncoded, doc.timestamp);
    }
}
```

Annexe 4 : structures des codes sources UX-ADOBLOCK et APP-ADOBLOCK



(a)



(b)