République Algérienne Démocratique Et Populaire
Ministère De l'Enseignement Supérieur Et De La Recherche Scientifique
Université Echahid Cheikh Larbi Tébessi - Tébessa
Faculté Des Sciences Exactes Et Sciences De La Nature Et De La Vie
Département De Mathématiques Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de MASTER

Domaine: Mathématiques et Informatique

Filière: Informatique

Option: Réseaux et Sécurité Informatique

Thème

Un modèle décentralisé basé sur la Blockchain pour les environnements IoT sécurisés

Présenté par:

HAFDALLAH Iheb

Devant le jury:

Dr. KHEDIRI Abderrazak	MCB	Université Echahid Cheikh Larbi Tébessi	Président
Dr. BOUROUGAA Salima	MCB	Université Echahid Cheikh Larbi Tébessi	Examinatrice
Dr. SAHRAOUI Abdelatif	MCA	Université Echahid Cheikh Larbi Tébessi	Encadreur

Date de soutenance: 7 juin 2023

Un modèle décentralisé basé sur la Blockchain pour les environnements IoT sécurisés

HAFDALLAH Iheb

Université Echahid Cheikh Larbi Tébessi

Remerciement

Tout d'abord, je tiens à remercier ALLAH le tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce modeste travail. Un remerciement particulier et sincère à mes parents pour les efforts qu'ils ont fournis. Ils ont toujours été présents et ce travail est un témoignage de ma gratitude et de mon profond respect envers eux.

Je souhaite exprimer mes sincères remerciements à Docteur SAHRAOUI Abdelatif pour ses précieux conseils et directives pédagogiques. Je tiens également à exprimer ma gratitude envers tous les professeurs qui m'ont enseigné et qui, par leurs compétences, m'ont soutenu dans la poursuite de mes études.

Je voudrais exprimer ma profonde gratitude envers les membres du jury pour l'intérêt qu'ils ont porté à ma recherche, en acceptant d'examiner mon travail et de l'enrichir par leurs propositions.

Enfin, je souhaite remercier toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce modeste travail. Votre soutien et votre collaboration ont été inestimables, et je vous exprime ici ma profonde gratitude et mon profond respect.

Merci à tous et à toutes.

Dédicace

À ma mère et mon père,

À ma grand-mère Aïcha,

À mes amis,

À mon encadreur Dr. SAHRAOUI Abdelatif.

Résumé

La croissance exponentielle des appareils de l'Internet des objets (IoT) a créé de nouvelles opportunités mais a également introduit des défis majeurs en termes de sécurité et de confidentialité des données. La sécurisation des données critiques générées par ces appareils est devenue une préoccupation cruciale. Dans l'infrastructure centralisée conventionnelle, les données sensibles des appareils IoT étaient stockées de manière centralisée. Cependant, cette approche présente le risque d'exposer les données privées des appareils IoT, car l'accès à l'intégralité des données stockées est possible.

Cette étude présente un modèle décentralisé basé sur la technologie Blockchain (BC) pour garantir la sécurité des environnements IoT. Ce modèle introduit la transparence et la résistance à la falsification dans le stockage et la récupération des données au sein des réseaux IoT, en utilisant des informations cryptées qui sont protégées contre tout accès ou utilisation non autorisés. La structure de la Blockchain proposée assure la sécurité du système en fonction d'un ensemble de contrats intelligents, adaptés en fonction des stratégies de stockage et de traitement des données.

Au cours de la phase d'expérimentation, nous avons conçu une application décentralisée basée sur la technologie Blockchain afin de renforcer la sécurité des politiques des maisons intelligentes. Cette application est associée à un prototype Arduino, ce qui permet une collecte de données en temps réel et un contrôle automatisé de la sécurité. Grâce à cette approche, il devient possible de créer des scénarios personnalisés pour automatiser la sécurité des politiques quotidiennes.

Les résultats expérimentaux obtenus dans ce travail montrent d'adoption du système décentralisé proposé avec la BC pour un environnement IoT. La solution fournie renforce la sécurité dans ces environnements IoT tout en assurant des mécanismes de confiance, de transparence qui peut aider à surmonter les problèmes de sécurité actuels.

Les mots Clés :

Internet des objets (IoT) • Blockchain • Décentalisé • Sécurité • Maisons intelligente.

Abstract

The exponential growth of Internet of Things (IoT) devices has created new opportunities but also introduced major challenges in terms of data security and privacy. Securing the critical data generated by these devices has become a critical concern. In the conventional centralized infrastructure, sensitive data from IoT devices was stored centrally. However, this approach has the risk of exposing the private data of IoT devices, as access to the entire stored data is possible.

This study presents a decentralized model based on Blockchain (BC) technology to guarantee the security of IoT environments. This model introduces transparency and tamper resistance to data storage and retrieval within IoT networks, using encrypted information that is protected from unauthorized access or use. The structure of the proposed Blockchain ensures the security of the system based on a set of smart contracts, adapted according to the data storage and processing strategies.

During the experimentation phase, we designed a decentralized application based on Blockchain technology to strengthen the security of smart home policies. This application is associated with an Arduino prototype, which allows real-time data collection and automated security monitoring. With this approach, it becomes possible to create custom scenarios to automate everyday policy security.

The experimental results obtained in this work show the adoption of the decentralized system that we propose with the BC for an IoT environment. The solution provided enhances security in these IoT environments while ensuring trust, transparency mechanisms that can help overcome current security challenges.

Keywords:

Internet of Things (IoT) • Blockchain • Decentralised • Security • Samrt Home.

ملخص

أدى النمو المتسارع لأجهزة إنترنت الأشياء (IoT) إلى خلق فرص جديدة ولكنه قدم أيضًا تحديات كبيرة من حيث أمان البيانات والخصوصية. أصبح تأمين البيانات الهامة الناتجة عن هذه الأجهزة مصدر قلق بالغ. في البنية التحتية المركزية التقليدية، يتم تخزين البيانات الحساسة من أجهزة إنترنت الأشياء مركزيًا. ومع ذلك، فإن هذا النهج ينطوي على خطر تعريض البيانات الخاصة لأجهزة إنترنت الأشياء، حيث يمكن الوصول إلى البيانات المخزنة بالكامل.

تقدم هذه الدراسة نموذجًا لامركزيًا يعتمد على تقنية Blockchain (BC) لضمان أمان بيئات إنترنت الأشياء. يقدم هذا النموذج الشفافية ومقاومة العبث لتخزين البيانات واسترجاعها داخل شبكات إنترنت الأشياء، باستخدام المعلومات المشفرة المحمية من الوصول أو الاستخدام غير المصرح به. يضمن هيكل Blockchain المقترح أمان النظام بناءً على مجموعة من العقود الذكية، والتي تم تكييفها وفقًا لاستراتيجيات تخزين البيانات ومعالجتها.

خلال مرحلة التجريب، قمنا بتصميم تطبيق لامركزي يعتمد على تقنية Blockchain لتعزيز أمان سياسات المنزل الذكي. يرتبط هذا التطبيق بنموذج أولي من Arduino، والذي يسمح بجمع البيانات في الوقت الفعلي والمراقبة الأمنية التلقائية. باستخدام هذا الأسلوب، يصبح من الممكن إنشاء سيناريوهات مخصصة لأمان القواعد اليومية.

تظهر النتائج التجريبية التي تم الحصول عليها في هذا العمل اعتماد النظام اللامركزي المقترح مع BC لبيئة إنترنت الأشياء. يعمل الحل المقدم على تعزيز الأمان في بيئات إنترنت الأشياء هذه مع ضمان الثقة وآليات الشفافية التي يمكن أن تساعد في التغلب على التحديات الأمنية الحالية.

الكلمات المفتاحية:

إنترنت الأشياء • بلوكتشين • لامركزية • الأمان • المنزل الذكي.

Table des matières

Liste des figures

Liste des tableaux

Liste des abréviations

AC: Autorité de certification

API: Application Programming Interface

BC: Blockchain

BTC: Bitcoin

CC: Cloud computing

DDoS: Distributed Denial of Service

DIC: Device, Identity, Connectivity

DoS: Denial of Service

DPoS: Delegated Proof of Stake

IIOT: Industrial Internet of Things

IdO: Internet des objets

IDE: Integrated Development Environment

IoT: Internet of Things

IP: Internet Protocol

IPSec: Internet Protocol Security

MEMS: Micro Electro Mechanical Systems

MITM: Man-in-the-middle

NFC: Near Field Communication

NIST: National Institute of Standards and Technology

P2P: Peer-to-Peer

PoA: Proof of Authority

PoC: Proof of Capacity

PoS: Proof of Stake

PoW: Proof of Work

QoE: Quality of Experience

QoS: Quality of Service

RFID: Radio Frequency Identification

SDD: Solid State Drive

SF: Sans Fil

TCP: Transmission Control Protocol

TIC: Technologies de l'information et de la communication

TLS: Transport Layer Security

TMN: Telecommunications Management Network

WSN: Wireless Sensor Network

Introduction Générale

Introduction

L'Internet des objets (IoT) est une technologie émergente qui permet la connectivité d'objets physiques à Internet, créant un réseau d'appareils interconnectés. Cette technologie a connu une croissance exponentielle ces dernières années, avec des milliards d'objets connectés dans le monde, tels que des capteurs, des caméras de surveillance, des dispositifs de suivi de la santé, des voitures intelligentes, etc.

Cependant, l'IoT fait face à plusieurs défis majeurs, notamment dans le domaine de la sécurité et de la confidentialité des données échangées entre les objets connectés. Les échanges de données sont souvent centralisés et vulnérables aux attaques malveillantes, mettant en péril la sécurité des données et la vie privée des utilisateurs.

Pour relever ces défis, la technologie Blockchain (BC) offre une solution potentielle en permettant des échanges de données décentralisés et sécurisés. La BC est une base de données distribuée qui stocke en toute sécurité les données et les transactions. Contrairement aux bases de données traditionnelles, elle manque un point de contrôle central, ce qui le rend décentralisé et résistant aux attaques malveillantes. La BC offre également de nombreux avantages, notamment la transparence, la sécurité, l'immuabilité, l'efficacité et la décentralisation, elle peut être utilisé dans divers domaines tels que la finance, l'assurance, la logistique, la santé, l'énergie, l'immobilier, etc.

Dans ce travail, nous nous concentrons sur la convergence de deux technologies, à savoir la BC et l'Internet des objets (IoT). Nous étudierons en profondeur les différents aspects de la BC et de l'IoT, en examinant les défis actuels et futurs en matière de sécurité et de confidentialité. De plus, nous explorerons diverses architectures décentralisées basées sur la BC pour les environnements IoT.

Problématique

Notre problématique consiste à trouver des solutions pour garantir la confidentialité des données dans les environnements IoT, en prenant en compte les particularités de la technologie IoT ainsi que les limites de ressources des dispositifs IoT, tout en permettant une utilisation efficace de la technologie du BC. Plus spécifiquement, nous nous posons les questions suivantes:

- Comment assurer la sécurité et la confidentialité des données dans les environnements IoT en tenant compte des caractéristiques propres de cette technologie et des contraintes de ressources des dispositifs IoT ?

- Comment utiliser la technologie BC de manière optimale pour sécuriser les échanges de données dans les environnements IoT, tout en minimisant les impacts sur les performances réseau et les coûts ?
- Comment concevoir un modèle décentralisé basé sur la BC pour assurer la sécurité des environnements IoT, tout en prenant en considération les contraintes de ressources des dispositifs IoT ?

Objectif

Notre travail vise à concevoir et mettre en place un modèle décentralisé basé sur la technologie BC pour assurer la sécurité et la confidentialité des données échangées au sein des environnements IoT. Notre objectif est de proposer une solution innovante qui soit adaptée aux contraintes des environnements IoT sécurisés, tout en minimisant les coûts et les impacts sur les performances. Ainsi, notre démarche consiste à garantir la sécurité et la confidentialité des données échangées entre les dispositifs IoT, tout en optimisant les ressources disponibles et en offrant une solution fiable pour les environnements IoT sécurisés.

Plan de mémoire

Notre mémoire est structuré en quatre chapitres qui abordent différents aspects de notre recherche sur le modèle décentralisé basé sur la technologie BC pour assurer la sécurité des environnements IoT.

- **Chapitre 1:** Introduction aux environnements IoT et la technologie de Blockchain.
- **Chapitre 2:** État de l'art des mécanismes de sécurité décentralisés pour l'IoT.
- **Chapitre 3:** Architecture de sécurité basée sur la Blockchain pour les politiques des services dans un Smart Home.
- **Chapitre 4:** Implémentation et évaluation.
- **Conclusion et perspective.**
- **Références.**

Chapitre 1: Introduction aux environnements IoT et la technologie de Blockchain

1. Introduction

Ces dernières années, l'Internet des objets est devenu la nouvelle génération d'Internet la plus populaire et l'une des tendances technologiques les plus répandues du XXIe siècle. Où il est devenu possible de créer un réseau d'objets physiques qui contiennent des programmes, des capteurs et de multiples technologies numériques qui leur permettent de communiquer et d'échanger des données, aux dispositifs et équipements médicaux les plus avancés et même des outils industriels au monde.

Les environnements IoT (Internet des objets) et la technologie de Blockchain sont deux domaines technologique ont le potentiel de révolutionner la façon dont nous interagissons avec les appareils de quotidien. La relation entre la Blockchain et l'Internet des objets se développe rapidement, en particulier dans les réseaux contenant des milliards d'appareils intelligents à l'avenir. La combinaison de la technologie IoT et de la Blockchain promet une nouvelle génération de systèmes de gestion de données IoT plus sécurisés et transparents.

Dans le premier chapitre, nous allons introduire les concepts de base d'un environnement IoT en présentant leur définition, ses architectures, ses caractéristiques, ses applications et les défis de sécurité de ces environnements. Ensuite, nous allons présenter également la technologie de Blockchain, ses structures et leur principe de fonctionnement.

2. Les environnements IoT

L'Internet des objets (IoT) est une technologie émergente qui a le potentiel de collecter des données en temps réel. Il s'agit donc d'une révolution technologique majeure venant dans de nombreux domaines. Considérés par de nombreux institutions de recherche comme le futur de la technologie, car ils sont déjà partout autour de nous, les appareils en ligne sont désormais essentiels dans une gamme d'industries et de secteurs, et presque tout dans nos vies aujourd'hui - agriculture, industrie et santé - dépend de ces technologies [**1**].

2.1. Historique

Le concept de l'Internet des objets (IoT) n'est pas nouveau. Des chercheurs de l'université Carnegie Mellon ont utilisé des capteurs pour surveiller la consommation de boissons dans un bâtiment universitaire dans les années 1980. Depuis lors, la technologie a évolué pour inclure des objets de plus en plus complexes, tels que des capteurs de température, des caméras de sécurité, des compteurs d'eau et des thermostats connectés, et il est prévu que d'ici 2025, 41,6 milliards d'appareils collecteront des données sur la façon dont nous vivons, nous travaillons et nous déplaçons dans les villes, le fonctionnement et l'entretien des machines dont nous dépendons [**2**].

2.2. Définition

Les environnements Internet des objets (IoT) visent à utiliser de nombreux objets physiques, tels que des capteurs, des caméras et des appareils électroniques connectés à Internet qui peuvent communiquer entre eux pour collecter et échanger des données. L'objectif de l'Internet des objets est de créer un réseau d'objets connectés qui peuvent communiquer ensemble de manière indépendante pour résoudre des problèmes spécifiques. L'Internet des objets est devenu un réseau mondial omniprésent où tout le monde sera connecté à Internet [**3**].

2.3. Motivation

L'IoT présente de nombreux motivation, tant pour les entreprises que pour les consommateurs, de sorte que l'intérêt de recherche en ce domaine a augmenté de façon exponentielle et devenu l'une des premières priorités des recherches industriel et académique [**4**]. Dans ce qui suit, nous présenterons de nombreuses motivations, allant de l'amélioration de l'efficacité et de l'expérience client à la réduction des coûts et à l'innovation de produits.

- **La collecte de données en temps réel:** est le processus d'utilisation des dispositifs IoT en temps réel pour suivre les décisions d'un système intelligent. Les données capturé par ce processus sont transmises, stockées et peuvent être récupérées à tout moment pour prendre des décisions plus utiles.
- **L'automatisation des processus:** Avec les appareils IoT, de nombreux processus quotidien peuvent être automatisés. Par exemple, certains travaux quotidiens sont un ensemble de tâches simples, répétitives et ennuyeuses. Pourtant, ils sont indispensables au fonctionnement de l'entreprise et au respect de la réglementation. À l'aide d'appareils IoT, ces tâches peuvent être automatisées dans le but est de réduire les coûts et d'améliorer l'efficacité des tâches. Cela permet aux employés de se concentrer sur les tâches les plus importantes.
- **L'amélioration de l'expérience des utilisateurs:** de nombreuses applications IoT utilisent des services IoT accessible sur internet avec différentes critères de qualité de service (*QoS*). Fournir des services IoT dans un marché ou des services plus personnalisés en utilisant uniquement des métriques QoS sans tenir compte la qualité d'expérience des utilisateurs (*QoE*) n'est pas suffisant pour adaptés aux besoins individuels. Un modèle de qualité d'expérience (*QoE*) est devenu crucial pour améliorer ces expériences.
- **La surveillance et la gestion à distance:** Les appareils IoT peuvent être utilisés pour surveiller et gérer à distance des systèmes, ce qui peut être particulièrement utile dans certains contextes tels que la surveillance d'équipements industriels ou la surveillance de la santé des patients.

- **Réduction des coûts:** un environnement IoT permet de réduire les coûts de fonctionnement d'une entreprise. Par exemple, réduire les coûts associe au processus de production ou de consommation d'énergie.
- **L'innovation et le développement de nouveaux produits:** L'Internet des objets peut être intégré dans le processus de développement de nouveaux produits et services, ainsi que l'amélioration des produits existants en intégrant des fonctions connectées.

2.4. Architecture de l'IoT

Une architecture IoT se compose de plusieurs systèmes connectés organisé en couches, chaque système remplit une fonction spécifique pour assurer un bon fonctionnement global. Un certain nombre de recherches qui ont été proposées confirment qu'il n'existe pas une architecture de référence pour un environnement IoT, parce que ces environnements sont constitués d'une variété de systèmes intelligents. Pour cela, nous citons un certain nombre d'architectures IoT qui peuvent répondre au besoin potentiel comme l'architecture de sécurité multicouches proposé dans [**5**], dont les auteurs proposent une architecture de sécurité multi couches à trois niveaux tandis que *Wan et al* [**6**] proposent une architecture à quatre niveaux. *Zhang et Fuquan* [**7**], ont proposé une architecture à cinq couches en utilisant les meilleures caractéristiques des architectures des réseaux de gestion Internet et des télécommunications basées respectivement sur les modèles TCP/IP et TMN. De même, une architecture à six couches a également ont été proposée sur la base de la structure hiérarchique du réseau [**8**]. En conséquence, nous résumons les principaux éléments qui peuvent composer un système IoT comme suit:

2.4.1. La couche des périphériques

Cette couche est constituée de capteurs, d'actionneurs et d'objets connectés. Ces objets collectent les données de l'environnement et interagissent avec lui. Les capteurs peuvent mesurer une variété de grandeurs physiques telles que la température, l'humidité, la pression, la luminosité, etc. Alors que les actionneurs peuvent être utilisés pour contrôler des dispositifs tels que des vannes, des moteurs, des lumières, etc.

2.4.2. La couche de connectivité

Cette couche permet la communication entre les appareils IoT et les systèmes de gestion des données. Elle se compose de diverses technologies de communication, telles que Wi-Fi [**9**], Bluetooth, Zigbee [**10**], NFC [**11**], etc. Ainsi que de passerelles qui collectent les données des appareils et le transfert de ces données vers des systèmes de gestion de données

2.4.3. La couche de gestion des données

Cette couche est responsable de la collecte, du stockage et de la gestion des données générées par les appareils IoT. Cela inclut les bases de données, les systèmes de gestion de

fichiers, les plates-formes Cloud, etc. Ces systèmes stockent et gèrent les données de manière sécurisée et efficace.

2.4.4. La couche d'analyse des données

Cette couche est utilisée pour traiter les données générées par les appareils IoT afin de les convertir en décisions utiles. Elle fournit un ensemble des algorithmes d'analyse de big data pour construire des modèles prédictive à partir des données.

2.4.5. La couche des applications IoT

Cette couche fournit des applications et des services qui tirent parti des données générées par les appareils IoT. Il comprend des applications mobiles, des interfaces utilisateur, des portails Web, etc. Ce qui permet aux utilisateurs de surveiller, de contrôler et d'interagir avec les appareils IoT.

2.4.6. La couche de sécurité

Cette couche est nécessaire pour assurer la sécurité des appareils IoT et des données qu'ils génèrent. Elle comprend des techniques de chiffrement, de gestion des identités et des accès, la surveillance des menaces, etc. En outre, ces techniques aident à protéger les appareils IoT et les systèmes de gestion des données contre les attaques malveillantes et les atteintes à la vie privée.

2.5. Les éléments de l'IoT

Comme nous avons discuté précédemment, l'Internet des objets est un domaine qui utilise un large éventail de technologies pour connecter des appareils et des systèmes, collecter et traiter des données, fournir des services et des applications intelligentes. Nous allons présenter dans ce qui suit, un aperçu de certaines des principales technologies utilisées dans l'Internet des objets:

2.5.1. Identification par radiofréquence (RFID)

L'identification par radiofréquence (*RFID*) [**12**] est une technologie qui permet d'identifier des objets à distance à l'aide d'ondes radio. Un système RFID est composé de deux composants: un tag (ou étiquette RFID) qui est attaché à l'objet à identifier et un lecteur RFID qui envoie des ondes radio pour communiquer avec l'étiquette et récupérer les informations qu'il contient.

2.5.2. Réseau de capteurs sans fil (WSN)

Un réseau de capteurs sans fil (*WSN*) est un réseau de capteurs interconnectés via des liaisons sans fil (*SF*), sont l'objective est de collecter des données de l'environnement. Les capteurs sont généralement équipés par un ensemble des composants nécessaires pour une tâche donnée à savoir, des batteries, d'émetteurs radio, des interfaces physiques permettent la communication avec d'autres capteurs ou une station de base.

2.5.3. Cloud Computing

Le Cloud Computing (*CC*) est une technologie de traitement de l'information permet d'accéder à des ressources informatiques (telles que des serveurs, du stockage, des applications, etc.) via une connexion internet. Au lieu d'héberger des données et des applications sur un ordinateur local, l'utilisateur peut les stocker et les exécuter dans des centres de données distants, qui sont gérés par des fournisseurs de services Cloud. Le CC entrelacé avec des objets intelligents utilisant potentiellement des millions de capteurs peut être d'énormes avantages et peut aider l'IoT à se développer en grande échelle. La recherche est donc lancée car l'IoT dépendra entièrement du CC [**13**].

2.5.4. Technologies de mise en réseau

Les technologies de mise en réseau sont utilisées pour connecter des appareils intelligents, des capteurs et des appareils à Internet et à d'autres dispositifs. Pour le réseau de transmission à grande portée, on utilise couramment les réseaux de 3G, 4G et 5G, etc. De même, pour un réseau de communication à courte portée, on utilise des technologies Bluetooth, Wi-Fi, etc.

2.5.5. Technologies Nano

La nanotechnologie peut jouer un rôle important dans l'IoT en fournissant des solutions innovantes pour les capteurs, les actionneurs, les systèmes de stockage de données et la communication entre les appareils. Cette technologie peut être utilisée pour produire des capteurs miniatures qui peuvent être intégrés dans des appareils portables, des vêtements intelligents, des implants médicaux, etc. Les nano-capteurs peuvent détecter des paramètres tels que la température, l'humidité, la pression, la lumière, les gaz, les molécules biologiques, etc. Les nano-capteurs peuvent également être utilisés pour détecter des signes de maladie et surveiller la santé des patients.

2.5.6. Technologies des systèmes micro-électro-mécaniques (MEMS)

La technologie MEMS peut jouer un rôle important dans l'Internet des objets en proposant des capteurs miniatures. La combinaison entre MEMS et la nanotechnologie offre une solution rentable pour optimiser le système de communication de l'IoT. Elle présente d'autres avantages tels que la réduction de la taille des capteurs et les actionneurs, des dispositifs informatiques embarqués, l'informatique ubiquitaire, une bande passante de fréquence plus élevée, etc. [**14**].

2.5.7. Technologies optiques

Les technologies optiques peuvent jouer un rôle important dans l'IoT en fournissant des solutions de communication sans fil à haut débit, large bande passante et à faible consommation d'énergie. Les technologies optiques utilisent des signaux lumineux pour

transmettre des données, permettant des vitesses de transfert beaucoup plus rapides que les technologies sans fil traditionnelles, telles que le Wi-Fi et le Bluetooth.

2.6. Les applications IoT

Les applications IoT intelligentes offrent plus de clarté, d'informations et d'efficacité en capturant les données des capteurs des appareils connectés à l'aide des technologies cité précédemment. Les applications IoT sont très diverses et existent dans de nombreux secteurs, dans ce qui suit nous fournirons quelques exemples:

- **Smart Home:** Une maison intelligente est un ensemble de systèmes intelligents multifonctionnels qui ont la capacité de gérer une maison grâce à un ensemble d'applications et de télécommandes connectées à Internet. Ces systèmes offrent également aux propriétaires plus de confort et de sécurité. En plus, rationaliser la consommation d'énergie et de faciliter l'exécution des tâches. Ils peuvent également contrôler à distance des appareils domestiques intelligents tels que des thermostats, des caméras de sécurité, des systèmes de sécurité, des lumières, des serrures et des télécommandes à partir d'un Smartphone ou d'une tablette.
- **Health Care:** L'industrie de la santé commence à adopter IoT pour prendre soin des patients et suivre leurs besoins. Nous pouvons citer quelques exemple d'applications IoT dans Health Care à savoir la surveillance à distance, bandes de fitness, intégrations de dispositifs médicaux, balances intelligentes, trackers de fitness, biocapteurs mobile, tensiomètres, glucomètres, distributeurs d'ordonnances, lits intelligents, la collecte de données de santé et aide au suivi des maladies chroniques.
- **Agriculture intelligente:** Les capteurs IoT peuvent être utilisés pour surveiller les cultures agricoles dans les champs, permettant également aux agriculteurs de prendre des décisions plus éclairées concernant l'irrigation, la fertilisation et le contrôle des maladies.
- **Véhicules intelligents:** Les véhicules intelligents peuvent utiliser des capteurs IoT pour collecter des données de conduite et du trafic, l'état du véhicule et des données environnementales, ce qui peut contribuer à améliorer la sécurité routière et l'efficacité de la conduite sur route.
- **Industrie 4.0:** Les usines et les entrepôts peuvent utiliser IoT pour surveiller les machines, les équipements et les stocks, ce qui peut contribuer à améliorer l'efficacité et la sécurité de la production.
- **Smart Cities:** Les villes peuvent utiliser IoT pour préserver les paramètres environnementale, contrôler la consommation énergétique, la gestion des déchets, la

sécurité publique et la circulation, ce qui peut contribuer à améliorer la qualité de vie des citoyens.

- **Logistique et Supply Chain:** Les capteurs IoT peuvent être utilisés pour suivre les marchandises tout au long de la chaîne d'approvisionnement, de la production à la livraison, ce qui peut contribuer à améliorer l'efficacité des systèmes de transports et de logistique.

2.7. La sécurité dans l'IoT

La sécurité IoT vise à protéger les appareils et les réseaux auxquels ils se connectent contre les menaces et les intrusions sur Internet. Ceci est réalisé en identifiant, surveillant et traitant les vulnérabilités de sécurité potentielles sur les appareils. Dans sa forme la plus simple, la sécurité IoT est la pratique qui assure la sécurité des systèmes IoT. Pour assurer la sécurité des appareils IoT et la sécurité du réseau, nous présentons en ce qui suit quelques techniques de sécurité doivent être mises en œuvre dans le réseau:

2.7.1. La confidentialité des données

Parmi les menaces auxquelles sont confrontées les données collectées à partir des capteurs IoT est la possibilité que ces données soient divulguées à des tiers non fiables. Cette divulgation peut avoir des problèmes de confidentialité pour les systèmes et des conséquences négatives pour les utilisateurs. Les entreprises ont besoin donc de protéger leurs données en utilisant des techniques de sécurité multiple de cryptage et de contrôle d'accès. En effet, la manipulation de la sécurité dans IoT est un grand enjeu ces dernières années. Il s'agit d'approches utiles et des méthodes pour limiter la divulgation d'informations, c'est-à-dire des politiques de confidentialité qui peuvent empêcher la lecture ou la modification d'informations confidentielles.

2.7.2. La sécurité des communications

Lors de l'adoption de l'IoT dans les entreprises, l'une des principales préoccupations de ces entreprises est de s'assurer que leurs systèmes sont bien sécurisés. Ainsi que, les communications entre les périphériques IoT et les serveurs de l'entreprise doivent être sécurisés pour empêcher les interceptions et les attaques de type « *man-in-the-middle* ». Aujourd'hui, les approches les plus adoptés pour la communication IoT doivent implémentées via le protocole http, ou à l'aide de protocoles de sécurité tels que TLS (*Transport Layer Security*) ou IPSec (*Internet Protocol Security*) [**15**]. Lorsqu'il s'agit de transfert de données, il y a un ensemble de critères d'efficacité doivent être vérifiés: la consommation d'énergie, la gigue, la latence de sécurité et l'utilisation de la bande passante. La mise en œuvre de ces éléments peut générer des compromis en fonction des exigences fonctionnelles à satisfaire [**16**].

2.7.3. Authentification et autorisation

Les appareils IoT doivent être authentifiés pour s'assurer qu'ils sont autorisés à accéder au réseau. Les utilisateurs doivent également être autorisés à accéder aux données collectées par les capteurs IoT.

2.7.4. Sécurité physique des périphériques

Les périphériques IoT doivent être protégés contre le vol ou la destruction physique. Les dispositifs doivent également être protégés contre les attaques par injection de code malveillant, laquelle peut exploiter les vulnérabilités du système.

2.7.5. Gestion de la sécurité

Les appareils IoT doivent être protégés contre le vol ou la destruction physique. Les appareils doivent également être protégés contre les attaques par injection de code malveillant, qui peuvent exploiter les vulnérabilités du système.

2.8. Les types d'attaques dans l'IoT

Il existe différents types d'attaques possibles dans l'IoT, nous présentons quelques exemples d'attaques:

- **Attaques par déni de service (DDoS):** Les attaques par déni de service [**17**] consistent à inonder un système de demandes de connexion ou de données, ce qui peut rendre le système indisponible pour les utilisateurs légitimes. Les attaques DDoS dans l'Internet des objets peuvent provenir d'appareils compromis connectés au réseau.
- **Attaques par injection de code malveillant:** Les attaquants peuvent exploiter les vulnérabilités des systèmes IoT pour injecter du code malveillant dans les appareils [**18**]. Un code malveillant peut être utilisé pour collecter des données sensibles ou prendre le contrôle d'appareils.
- **Attaques de réseau:** Les attaques réseau consistent à intercepter ou à modifier les données circulant entre les appareils IoT et les serveurs [**19**]. Ces attaques pourraient permettre aux attaquants d'accéder à des données sensibles ou de modifier les commandes envoyées aux appareils.
- **Attaques de l'homme du milieu (MITM):** Les attaques MITM impliquent l'interception des communications entre les appareils IoT et les serveurs pour espionner ou modifier les données [**20**].Les attaques MITM sont particulièrement préoccupantes dans les réseaux IoT non sécurisés.
- **Attaques par phishing:** Les attaques de phishing sont souvent utilisées pour cibler les utilisateurs d'appareils IoT [**18**]. Les attaquants peuvent envoyer des e-mails ou des messages de Phishing pour inciter les utilisateurs à révéler leurs identifiants de connexion ou à télécharger du code malveillant.

- **Attaques physiques:** Les appareils IoT sont souvent utilisés dans des environnements non sécurisés et peuvent faire l'objet d'attaques physiques [**19**]. Les attaquants peuvent voler ou endommager des appareils pour accéder à des données sensibles ou perturber les opérations.

2.9. Défis de sécurité pour les applications de l'IoT

La sécurité est l'un des principaux défis de l'Internet des objets. Les appareils IoT collectent, stockent et échangent des données sensibles, telles que des informations personnelles et des données de santé. Cela rend l'Internet des objets un environnement plus ciblé par les cybercriminels qui cherchent à accéder à ces informations pour leur propre bénéfice. Les appareils IoT sont souvent vulnérables aux attaques DDoS, aux attaques par injection de code malveillant, aux attaques d'ingénierie sociale, aux attaques de Phishing, etc. De plus, certains appareils IoT ne sont pas conçus pour être mis à jour régulièrement, ce qui les rend vulnérables aux attaques de logiciels malveillants. Les défis de sécurité les plus courants dans l'Internet des objets sont:

- **L'authentification et l'autorisation:** La plupart des appareils IoT sont conçus pour se connecter automatiquement aux réseaux sans fil, ce qui facilite les accès non autorisés. Les mécanismes d'authentification et d'autorisation doivent être renforcés pour assurer la sécurité des appareils IoT.
- **Cryptographie:** Les appareils IoT collectent et communiquent souvent des données sensibles et privées. Le cryptage est un moyen essentiel pour assurer la confidentialité et l'intégrité de ces données. Cependant, les algorithmes de chiffrement nécessitent des ressources de traitement et de stockage importantes, ce qui peut constituer un défi pour les appareils IoT.
- **La gestion des clés:** La gestion des clés est un aspect important de la cryptographie car il est important que les clés cryptographiques soient utilisées pour protéger les données à stocker en toute sécurité et à gérer efficacement. Cela représente un défi pour les appareils IoT, qui disposent souvent de ressources limitées et peuvent ne pas être en mesure de stocker les clés en toute sécurité.
- **La confidentialité:** Les données collectées par les appareils IoT peuvent contenir des informations sensibles, telles que des informations médicales ou des données de localisation. La confidentialité de ces données est donc importante. Cependant, le cryptage peut ne pas être suffisant pour protéger la confidentialité, car les métadonnées peuvent être utilisées pour identifier un utilisateur.
- **La mise à jour et la maintenance des appareils:** Les appareils IoT sont souvent conçus pour être autonomes et sans intervention humaine. Cela peut rendre les appareils difficiles

à mettre à jour et à entretenir, ce qui peut les rendre vulnérables aux attaques connues ou aux vulnérabilités de sécurité. Par conséquent, la gestion de la sécurité des appareils IoT nécessite une planification et une gestion appropriées pour assurer une protection adéquate.

3. Les Blockchains

3.1. Historique

La Blockchain est une technologie relativement nouvelle, datant des années 1990. La Blockchain a été rendue célèbre en 2008 par la publication de l'article « *Bitcoin: A Peer-to-Peer Electronic Cash System* » de *Satoshi Nakamoto*, pseudonyme du créateur ou du groupe de créateurs de bitcoins [**21**].

Le *Bitcoin* (BTC) [**22**]est la première utilisation pratique de la BC, qui sert à enregistrer toutes les transactions effectuées sur le réseau BTC.

Depuis la création de la BTC, de nombreuses autres BC ont été développées, chacune avec ses propres caractéristiques et applications. Par exemple, *Ethereum* a été lancé en 2015 et est utilisé pour exécuter des contrats intelligents et des applications décentralisées (*dApps*). D'autres BC ont été créées pour des applications spécifiques, telles que *Ripple* [**23**] pour les paiements internationaux et *Hyperledger* [**24**] pour les entreprises.

L'histoire de la BC est relativement courte, mais elle est riche en événements et en développements, et son avenir s'annonce prometteur.

3.2. Définition

La BC est une technologie de stockage et de transmission d'informations permet de stocker les données de manière transparente, sécurisée et décentralisée. Contrairement à un système centralisé où toutes les données sont stockées sur un serveur central, une BC est distribuée sur un réseau d'ordinateurs appelés « *nœuds* ». Chaque nœud contient une copie complète de la BC, qui est une série de blocs contenant des informations cryptées.

La technologie BC n'est pas seulement une technologie unique, mais elles utilisent les techniques de sécurité liée aux données à savoir les techniques de cryptages, les preuves mathématiques et les modèles commerciaux. La combinaison avec les réseaux *peer-to-peer* et l'utilisation d'un algorithme de consensus distribué pour résoudre le problème de la synchronisation de la base de données des données distribuées traditionnelles [**25**] [**26**]. En effet, la technologie BC est caractérisée parles éléments suivants:

- **Open source:** La plupart des systèmes de BC sont open source, les enregistrements peuvent être vérifiés publiquement et les utilisateurs peuvent également utiliser les BC pour créer l'application de leur choix.

- **Transparent:** L'enregistrement des données par le système de BC est transparent pour chaque nœud, il est également transparent lors de la mise à jour des données.
- **Décentralisé:** C'est la caractéristique de base de la BC, signifie que la BC n'a plus besoin de s'appuyer sur un nœud centralisé, les données peuvent être enregistrées, stockées et mises à jour de manière distribuée.
- **Autonomie:** En raison de la base de consensus, chaque nœud d'un système BC peut transférer ou mettre à jour des données en toute sécurité.
- **Immuable:** Tous les enregistrements seront réservés pour toujours et ne pourront être modifiés que si quelqu'un peut prendre le contrôle de plus de 51% de nœuds en même temps.
- **Anonymat:** La technologie BC a résolu le problème de confiance entre les nœuds, de sorte que le transfert de données ou même la transaction peut être anonyme, il suffit de connaître l'adresse BC de la personne concerné.

3.3. Les générations de BC

Chaque génération de BC a ajouté de nouvelles fonctionnalités et amélioré la technologie, permettant de nouvelles applications et de nouvelles opportunités, avec la possibilité que de nouvelles générations émergent à l'avenir. Cette section nous permettra de connaître les bases du développement continu de chaque génération de la BC, et de comprendre le but de sa création et quel est le problème de chaque génération. *Daniel Ishbiah* a défini ces générations comme suit [**27**]:

3.3.1. La première génération

La première génération de la BC est représentée par *BTC*, qui a été lancé en 2009. Cette génération comportait l'utilisation de la preuve de travail (*PoW*) pour sécuriser le réseau, traiter les transactions et créer de nouvelles unités de crypto-monnaie.

3.3.2. La deuxième génération

La deuxième génération de la BC a été introduite avec *Ethereum* en 2015. Cette génération a introduit l'utilisation de contrats intelligents, qui sont des programmes autonomes pouvant s'exécuter sur la BC. Les contrats intelligents ont permis de créer des applications décentralisées (*dApps*) et des protocoles complexes sur la BC.

3.3.3. La troisième génération

La troisième génération de *BC* est représentée par des projets tels que *Cardano* [**28**] et *Polkadot* [**29**]. Cette génération se concentre sur l'évolutivité et la sécurité de *BC*, en utilisant des mécanismes de consensus plus avancés et en améliorant les protocoles de sécurité.

3.3.4. La quatrième génération

La quatrième génération de *BC* se concentre sur l'interopérabilité et la connectivité entre les différentes BC. Les projets de cette génération, tels que *Cosmos* [**30**] et *ICON* [**31**], permettent aux BC de communiquer entre elles et d'échanger des informations, ce qui ouvre la voie à une utilisation plus pratique de la technologie de la BC.

3.4. Le fonctionnement de la BC

La BC fonctionne en utilisant un réseau de nœuds, où chaque nœud contient une copie complète de la chaîne de blocs.

- **Création de blocs:** Lorsqu'une transaction est effectuée sur la BC, elle est enregistrée dans un bloc. Ce bloc contient les informations de transaction, ainsi qu'un hachage unique qui relie ce bloc au bloc précédent dans la chaîne.
- **Validation du bloc:** Le bloc doit être validé avant d'être ajouté à la chaîne par les nœuds du réseau. Le contrat vérifie la validité de la transaction et garantit qu'il y a suffisamment de fonds pour effectuer la transaction.
- **Ajout du bloc:** Une fois le bloc validé, il est ajouté à la chaîne. Le hash unique d'un bloc est enregistré sur la BC, reliant ainsi ce bloc au bloc précédent.
- **Consensus:** La BC utilise un mécanisme de consensus pour s'assurer que tous les nœuds du réseau s'accordent sur l'état de la chaîne. Dans le cas de *Bitcoin*, le mécanisme de consensus utilisé est la preuve de travail (*PoW*), où les mineurs doivent résoudre des problèmes mathématiques complexes pour ajouter un bloc à la chaîne.
- **Mise à jour de la chaîne:** Lorsqu'un bloc est ajouté à la chaîne, tous les nœuds du réseau sont mis à jour avec la nouvelle version de la chaîne. Cela garantit que tous les nœuds ont une copie exacte de la BC.
- **Sécurité:** La sécurité de la BC est assurée par l'utilisation de la cryptographie. Chaque bloc contient un hachage unique qui relie le bloc au bloc précédent, garantissant que toute modification apportée à un bloc affecte de la même manière tous les blocs suivants, rendant la manipulation des données pratiquement impossible.

3.5. Les Types de BC

Les participants peuvent interagir avec la BC en tant qu'écrivains ou en tant que lecteurs. Le lecteur est impliqué passivement dans le processus de transaction et se concentre sur l'analyse du contenu des enregistrements ou la validation de la BC. D'autre part, les écrivains sont activement impliqués dans le processus de transaction et ont la capacité d'étendre la chaîne en utilisant des protocoles de consensus [**32**]. Il existe plusieurs types de BC, y compris:

3.5.1. Blockchain publique

Ce type de BC est appelé BC sans autorisation (*permissionless*), et cette BC est ouverte à tous. Tout le monde peut participer à la vérification des transactions et à l'ajout de blocs à la chaîne.

3.5.2. Blockchain privée

Ce type de BC s'appelle une BC autorisée (*permissioned*), et cette BC est réservée à un groupe d'utilisateurs autorisés. Le contrat est contrôlé par un organisme qui contribue à maintenir la confidentialité et la sécurité des transactions.

3.5.3. Blockchain hybride

Comme son nom l'indique, la BC est une combinaison de BC publique et privée. Certaines parties de la BC sont publiques, tandis que d'autres sont réservées à des groupes d'utilisateurs autorisés.

3.5.4. Blockchain de consortium

Ce type de BC s'appelle une BC semi-privée, et la BC est gérée par un groupe d'entreprises. Il permet à un groupe d'entreprises de travailler ensemble en partageant des données, tout en gardant le contrôle sur la gestion de la BC.

En plus de ces types de BC, il existe également des variantes de BC, telles que les *Sidechains*, qui permettent de transférer des actifs entre plusieurs BC, ou les BC en couches, qui permettent la création d'applications décentralisées.

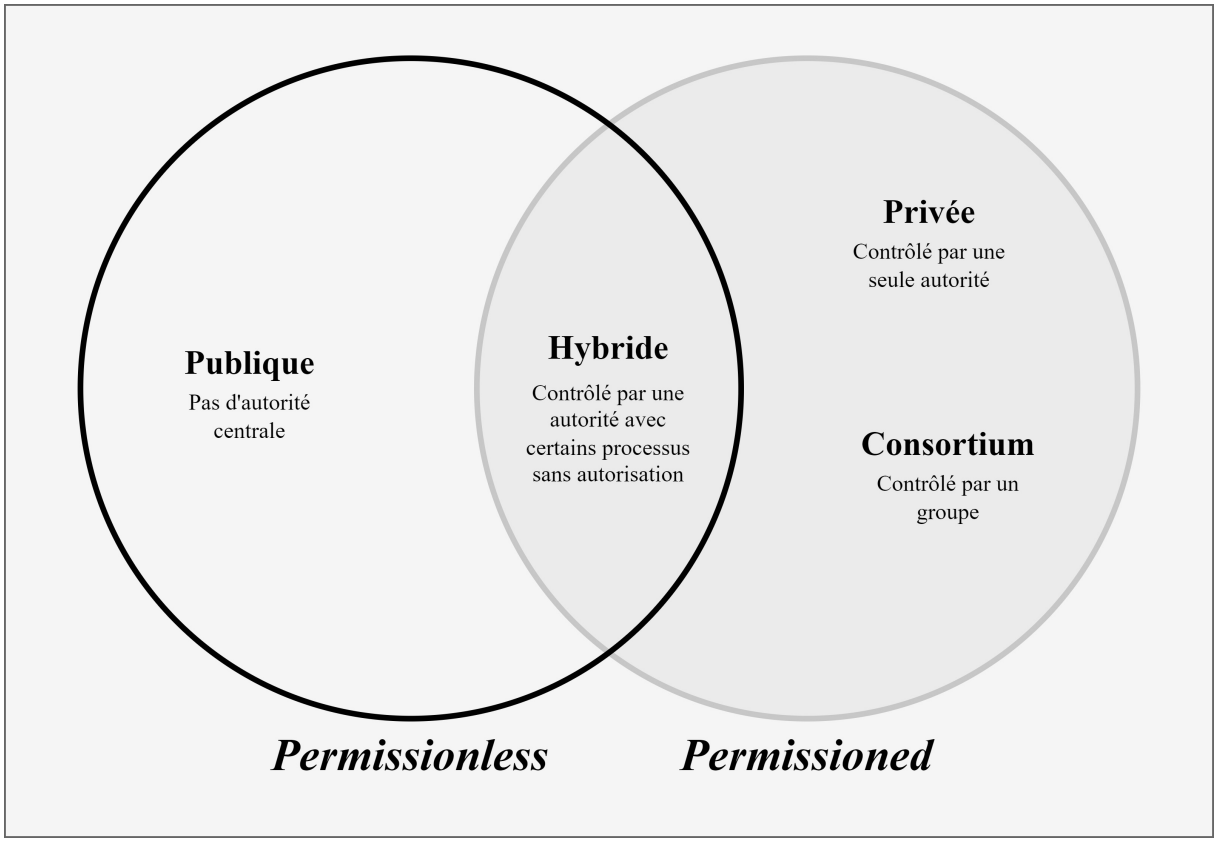


Figure 1: Les Types de Blockchains [**33**]

3.6. La structure de BC

En effet, chaque bloc de BC contient plusieurs éléments clés, à savoir les données de transactions, le hachage du bloc précédent, le hachage du bloc actuel, l'horodatage et d'autres informations [**34**]. Ces éléments sont présentés comme suit:

3.6.1. Les données de transactions

Selon le service auquel cette BC s'applique, une donnée de transaction est un enregistrement (compensation bancaire, IoT, etc.).

3.6.2. Hash

Le hachage est une fonction cryptographique qui prend une entrée de données (telle qu'un bloc dans une BC) et qui génère une sortie unique et de taille fixe appelée «*hash*». Cette sortie est essentiellement une empreinte digitale de l'entrée de données.

3.6.3. Horodatage (Timestamp)

Un élément important dans la structure de chaque bloc. Il indique la date et l'heure exactes de création et d'ajout du bloc à la chaîne. L'horodatage est essentiel pour maintenir l'intégrité de la BC, car il permet de garantir que les blocs sont générés selon un ordre chronologique

3.6.4. Des autres informations

Comme la signature du bloc, la valeur Nonce ou d'autres données définies par l'utilisateur.

3.7. Mécanismes de validation

Dans la BC, les mécanismes de vérification sont utilisés pour assurer l'intégrité des données et la sécurité du réseau. Les deux principaux mécanismes de vérification utilisés dans les BC sont la preuve de travail (*PoW*) et la preuve de participation (*PoS*). La plupart des plateformes BC actuelles, soit plus de 90% de la capitalisation boursière totale des crypto-monnaies, utilisent le mécanisme de consensus [**35**].

3.7.1. La preuve de travail

Le principe de base de la preuve de travail est d'exiger des mineurs qu'ils résolvent un problème mathématique complexe pour ajouter un nouveau bloc à la chaîne. Les mineurs doivent résoudre une équation cryptographique difficile en utilisant une grande puissance de calcul. Le premier mineur à résoudre le problème reçoit une récompense sous forme de crypto-monnaie.

3.7.2. La preuve d'enjeu

Un mécanisme alternatif de vérification de la preuve de travail est utilisé dans certaines chaînes de blocs. Au lieu de résoudre des problèmes mathématiques complexes, la preuve

d'enjeu repose sur le fait d'avoir un certain nombre de jetons pour déterminer le validateur du bloc.

Il existe également d'autres mécanismes de validation moins courants, tels que la preuve d'autorité (*PoA*) et la preuve de capacité (*PoC*) [**36**], qui impliquent respectivement l'utilisation d'entités de confiance pour valider les transactions et l'allocation d'espace de stockage pour valider les transactions.

3.8. Les types d'attaques de BC

En effet, il existe plusieurs types d'attaques qui peuvent être menées contre les réseaux de BC, à savoir:

- **L'attaque des 51%:** elle se produit lorsque plus de la moitié de la puissance de calcul de la BC est contrôlée par une seule entité. Cela permet à cette entité de manipuler les transactions et d'empêcher les autres participants de participer au réseau.
- **L'attaque par déni de service (DoS):** elle consiste à submerger le réseau BC de trafic malveillant afin de le rendre indisponible pour les utilisateurs légitimes.
- **L'attaque par réorganisation de la chaîne (Chain reorganization attack):** elle se produit lorsque les mineurs d'une BC organisent les blocs de manière différente, créant ainsi des Forks et des blocs invalides [**37**].
- **L'attaque de Sybil:** elle consiste à créer de multiples identités fictives pour envahir la BC et la rendre vulnérable à des manipulations [**38**].
- **L'attaque de double dépense (Double spending attack):** elle se produit lorsque l'utilisateur tente d'utiliser les mêmes fonds pour effectuer plusieurs transactions différentes [**39**].

Ces attaques ne sont pas exhaustives, mais elles représentent quelques exemples des menaces potentielles auxquelles les réseaux BC sont confrontés.

4. Conclusion

En conclusion, l'IoT et la BC sont deux domaines technologiques en évolution rapide qui offrent de nombreux avantages pour améliorer la vie quotidienne. L'Internet des objets permet à divers objets d'être connectés à Internet pour échanger des données et rendre la vie plus sûre et plus efficace. Cependant, cette large connectivité expose également les appareils IoT à des vulnérabilités et à des risques de sécurité. C'est là qu'intervient la BC, qui offre des solutions de sécurité avancées pour protéger les données et prévenir les attaques malveillantes.

Chapitre 2: État de l'art des mécanismes de sécurité décentralisés pour l'IoT

1. Introduction

Les appareils IoT sont de plus en plus omniprésents actuellement dans la vie quotidienne. Cependant, les systèmes d'information de ces appareils sont vulnérables aux attaques malveillantes et aux failles de sécurité, qui peuvent avoir de graves conséquences, telles que la divulgation de données sensibles et le contrôle à distance de ces appareils.

La sécurité de ces systèmes distribués est primordiale au niveau architectural et doit être adaptée aux besoins spécifiques des applications IoT. La sécurité de ces systèmes consiste à utiliser un ensemble des techniques et des outils de protection les appareils IoT et ces données contre les attaques malveillantes. En particulier, ces aspects de sécurité sont particulièrement importants pour un environnement IoT qui ne dépend pas des autorités centrales pour leur fonctionnement, ce qui les rend plus résistants aux attaques.

Dans ce chapitre, nous décrirons les systèmes de sécurité décentralisés pour les environnements IoT. Ensuite nous aborderons les différentes solutions et techniques qui ont été proposées pour la sécurisation de ces environnements.

2. Les systèmes décentralisée IoT

La technologie Blockchain est l'un des systèmes décentralisés qui permet de stocker les données de manière sécurisée et transparente. Dans lequel ses architectures sont adoptées pour résoudre des problèmes critiques dans les environnements IoT.

2.1. La Blockchain dans l'IoT

La BC peut avoir plusieurs applications dans l'IoT, nous pouvons citer quelques applications de la BC dans l'IoT:

- **Les applications de sécurité:** La BC peut garantir la sécurité des communications entre les appareils IoT en appliquant des méthodes de vérification et d'authentification des transactions. La BC permet de stocker les données chiffrées et distribuées, ce qui rend ces données moins vulnérables aux attaques malveillantes [**40**].
- **Les applications de gestion de l'identité:** Le BC peut être utilisé pour assurer l'authenticité de chaque appareil IoT, ce qui est important pour prévenir les intrusions et l'usurpation d'identité en garantissant l'authenticité de chaque objet connecté [**41**].
- **Les contrats intelligents:** Les contrats intelligents sont des programmes informatiques qui s'exécutent automatiquement en fonction de certaines conditions prédéfinies, peuvent être utilisés pour régir les relations entre différents appareils IoT. Les contrats intelligents peuvent aider à automatiser les processus et à réduire les coûts [**42**].

- **Les applications de paiements:** La BC est utilisée pour construire des systèmes crypto-monnaie entre les appareils IoT, ce qui peut aider à éliminer les intermédiaires coûteux et à réduire les délais de transaction [**43**].
- **Les applications de stockage sécurisé des données:** La BC est utilisée pour faciliter l'échange et le stockage de données entre les différents appareils IoT de manière sécurisée et fiable. Les données peuvent être stockées de manière distribuée sur la BC, ce qui garantit leur intégrité et leur confidentialité [**44**].

2.2. Avantages et limites

L'utilisation de la BC avec l'Internet des objets (IoT) présente plusieurs avantages et des limites, à savoir:

2.2.1. Avantages

- **Sécurité renforcée:** La BC est une technologie sécurisée et transparente qui peut aider à protéger les données des utilisateurs de l'IoT contre les cyber-attaques, la falsification de données et le piratage.
- **Réduction des coûts:** Les transactions de l'IoT peuvent être effectuées directement entre les utilisateurs sans l'intermédiaire d'une tierce partie, ce qui peut réduire les coûts de transaction.
- **Interopérabilité:** La BC permettre aux différents objets connectés de communiquer entre eux de manière transparente, quel que soit leur système ou leur plateforme.
- **Traçabilité:** La BC peut fournir une traçabilité complète de toutes les transactions, ce qui est particulièrement utile dans les industries qui nécessitent une chaîne d'approvisionnement transparente.

2.2.2. Limites

- **Scalabilité:** La BC est une technologie relativement lente et coûteuse, ce qui peut la rendre difficile à mettre à l'échelle pour l'IoT.
- **Consommation d'énergie:** Les protocoles de consensus nécessaires à la BC peuvent consommer beaucoup d'énergie, ce qui peut être problématique pour les objets connectés qui ont des ressources limitées.
- **Complexité:** La mise en place d'une architecture BC pour l'IoT peut être complexe et nécessiter des compétences techniques avancées.

3. Cryptographie à clé publique pour la sécurité de l'IoT

La cryptographie à clé publique, également connue sous le nom de cryptographie asymétrique, est une technique de cryptage utilisée pour sécuriser les communications entre

deux parties. Elle est largement utilisée dans les systèmes d'Internet des objets (IoT) pour protéger les données sensibles [**45**].

3.1. Concepts de la cryptographie à clé publique

La cryptographie à clé publique est un ensemble de techniques de chiffrement qui utilise deux clés mathématiquement liées pour sécuriser les communications entre deux parties, sans qu'elles aient besoin de se connaître au préalable, comme le montre la figure ci-dessous .En utilisant ces concepts, la cryptographie à clé publique permet de protéger la confidentialité des données, d'assurer l'authenticité des messages et de garantir que les messages ne peuvent pas être modifiés lors de leur transmission.

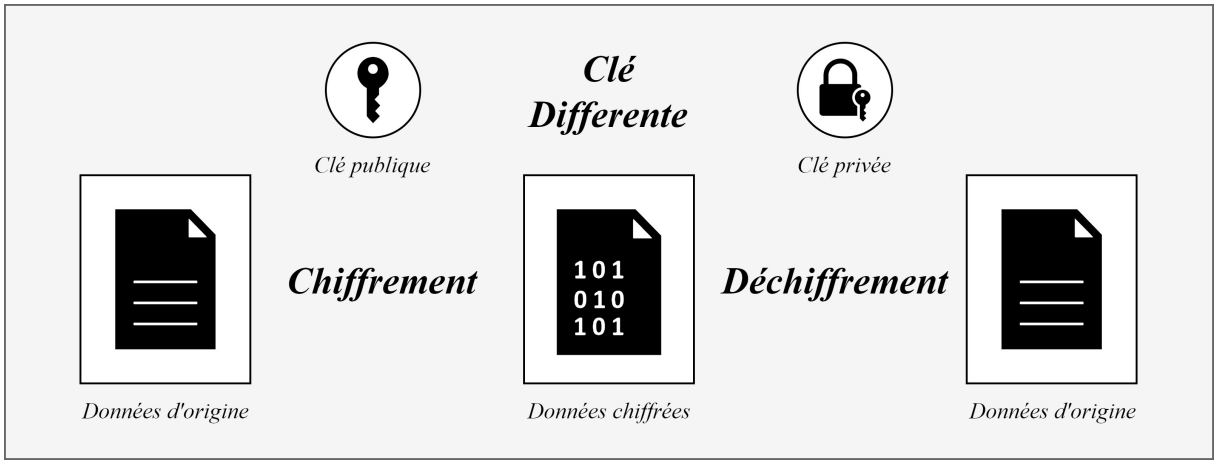


Figure 2: La cryptographie à clé publique

3.1.1. Clé publique

Une clé publique est une clé de chiffrement qui est disponible publiquement pour tout le monde. Elle est utilisée pour chiffrer les données avant leur envoi à un destinataire.

3.1.2. Clé privée

Une clé privée est une clé de décryptage gardée secrète par l'utilisateur et utilisée pour décrypter les données qui ont été cryptées avec la clé publique correspondante.

3.1.3. Chiffrement

Le cryptage est le processus de transformation des données dans un format illisible appelé « *texte chiffré* », à l'aide d'une clé publique.

3.1.4. Déchiffrement

Le déchiffrement est le processus de transformation du texte chiffré en données originales en utilisant la clé privée correspondante.

3.1.5. Signature numérique

Une signature numérique est une technique de cryptographie à clé publique qui permet à un expéditeur de signer numériquement un document ou un message. La signature numérique

utilise une clé privée pour générer une signature qui peut être vérifiée par toute personne ayant accès à la clé publique correspondante.

3.1.6. Certificat numérique

Un certificat numérique est un document électronique qui associe une clé publique à une entité (par exemple, une personne, une entreprise ou un site web) en utilisant la signature numérique d'une autorité de certification (*AC*). Les certificats numériques sont utilisés pour vérifier l'authenticité des clés publiques et des entités qui les possèdent.

3.2. L'efficacité de la cryptographie à clé publique dans IOT

Comme nous avons discuté précédemment, la cryptographie à clé publique peut être utilisée par les objets connecté pour sécuriser les communications entre ces objets, ainsi que la sécurité des passerelles IoT et les serveurs Cloud [**46**]. Cependant, l'évaluation de l'efficacité de la cryptographie à clé publique dans l'IoT dépend de plusieurs facteurs spécifiques à savoir:

3.2.1. Capacités des périphériques IoT

Certains appareils IoT ont une puissance de traitement, une mémoire et des capacités de stockage limitées. Ces limitations peuvent rendre difficile l'utilisation de clés plus longues et d'algorithmes de chiffrement plus complexes, ce qui peut réduire la sécurité de la cryptographie à clé publique [**47**].

3.2.2. Environnement de déploiement

Les périphériques IoT peuvent être déployés dans des environnements potentiellement hostiles, tels que des environnements industriels, urbains ou ruraux. Les interférences électromagnétiques, les attaques physiques et les environnements extrêmes peuvent affecter la qualité des communications et la sécurité de la cryptographie à clé publique.

3.2.3. Gestion des clés

La gestion des clés est une préoccupation majeure dans l'IoT, car un grand nombre de périphériques doivent être gérés. Les protocoles de gestion des clés doivent être robustes et fiables pour éviter tout compromis de sécurité [**48**].

3.2.4. Évolutivité

L'IoT est un domaine en constante évolution, avec de nouveaux périphériques, de nouveaux protocoles et de nouveaux cas d'utilisation qui émergent régulièrement. La cryptographie à clé publique doit être suffisamment évolutive pour répondre aux besoins de sécurité changeants de l'IoT [**49**].

L'évaluation de l'efficacité de la cryptographie à clé publique dans l'IoT dépend de facteurs tels que les capacités des périphériques IoT, l'environnement de déploiement, la gestion des clés et l'évolutivité. Donc, la cryptographie à clé publique est efficace pour les transactions à faible volume et à faible fréquence.

4. Les architectures IoT décentralisées

La décentralisation est un concept liée à la distribution du contrôle et l'autorité aux périphéries d'une organisation au lieu d'une seule autorité centrale contrôlant entièrement l'organisation. La décentralisation des systèmes IoT est une approche vise à établir des réseaux décentralisés d'appareils IoT interconnectés. Cette approche offre plusieurs avantages, notamment une plus grande résilience, une plus grande évolutivité, une meilleure confidentialité et une plus grande sécurité.

4.1. Décentralisation à l'aide de BC

Dans le contexte de l'IoT, la BC est utilisée pour créer des systèmes décentralisés qui permettent aux périphériques IoT de communiquer et d'interagir directement entre eux, sans passer par une infrastructure centrale telle qu'un serveur Cloud. Les transactions entre les périphériques IoT peuvent être vérifiées et sécurisées à l'aide de la BC, ce qui permet de garantir leur intégrité et leur confidentialité.

La figure ci-dessous montre différents types d'architectures proposées par Paul Baran

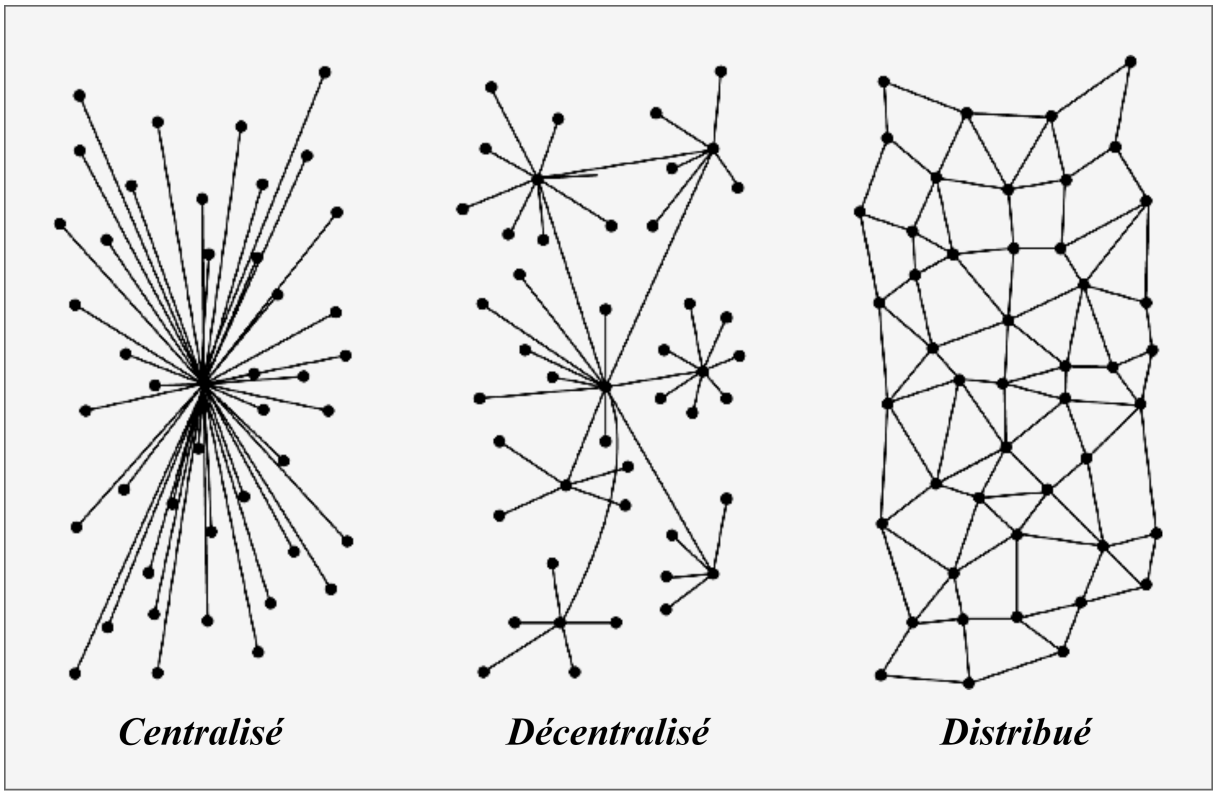


Figure 3: Différents types de réseaux/systèmes [**50**]

Une comparaison entre les systèmes centralisés et décentralisés (réseaux/applications) est présentée dans le tableau suivant:

Caractéristique	**Centralisé**	**Décentralisé**
Propriétaire	Fournisseur de services	Tous les utilisateurs
Architecture	Client/serveur	Distribué, différentes topologies
Sécurité	Basique	Plus sécurisé
La haute disponibilité	Non	Oui
Fault tolerance	Basic, single point of failure	Très tolérant, car le service est répliqué
Résistance à la collusion	Basique, parce qu'il est sous le contrôle d'un groupe ou même d'un seul individu	Très résistant, car les algorithmes de consensus assurent la défense contre les adversaires
Architecture applicative	Application unique	Application répliquée sur tous les nœuds du réseau
Confiance	Les consommateurs doivent faire confiance au fournisseur de services	Aucune confiance mutuelle requise
Coût pour le consommateur	Plus élevé	Plus bas

Tableau 1: La comparaison entre les systèmes centralisés et décentralisés [**51**]

4.2. Techniques de décentralisation

Deux méthodes peuvent être utilisées pour atteindre la décentralisation: la ***désintermédiation*** et la ***concurrence***.

4.2.1. Désintermédiation

Une méthode de décentralisation consiste à supprimer les intermédiaires entre les utilisateurs et les autorités locales ou régionales. Cette méthode réduit les coûts et augmente l'efficacité en éliminant les niveaux hiérarchiques intermédiaires. Il peut être mis en œuvre en utilisant les technologies de l'information et de la communication (*TIC*) pour permettre une communication directe entre les utilisateurs et les autorités locales ou régionales [**52**].

4.2.2. Décentralisation axée sur la concurrence

Ce type de décentralisation est méthode consiste à introduire une concurrence entre les différentes autorités locales ou régionales pour fournir des services aux utilisateurs. Cette méthode peut conduire à une amélioration de la QoS fournis, à une réduction des coûts et à

une plus grande responsabilisation des autorités locales ou régionales envers les utilisateurs. Elle peut être mise en œuvre en encourageant la concurrence entre les autorités locales ou régionales pour fournir des services locaux, ou en permettant aux utilisateurs de choisir entre différentes autorités locales ou régionales pour la fourniture de services [**53**].

4.3. Plateformes de décentralisation

Aujourd'hui, il existe de nombreuses plates-formes disponibles pour la décentralisation qui permettent aux utilisateurs de participer à des réseaux distribués, décentralisés et autonomes. Dans ce contexte, il existe plusieurs systèmes basés sur les crypto-monnaies, à savoir:

4.3.1. Bitcoin

Le *Bitcoin* (*BTC*) [**21**] est le premier système de crypto-monnaie décentralisée permet d'effectuer des transactions directes dans un réseau Peer-To-Peer (*P2P*). Le BTC utilise un grand livre public décentralisé pour stocker les transactions enregistrées sur le réseau. Les transactions sont vérifiées par les mineurs BTC en résolvant des problèmes mathématiques complexes pour ajouter de nouveaux blocs de transactions à la BC. La preuve de travail est utilisée pour valider les transactions et que les mineurs sont récompensés pour leur travail.

4.3.2. Ethereum

Le *Ethereum* (*ETH*) [**54**] est également l'un des premiers réseaux BC, ce réseau introduit un langage Turing complet et le concept de machine virtuelle. Avec la disponibilité de son langage Turing-complet [**55**], Solidity [**56**] et d'autres pistes des recherches sont ouvertes pour le développement d'applications décentralisées. Ce BC a été proposé pour la première fois en 2013 par Vitalik Buterin, il fournit un BC public pour développer des contrats intelligents et des applications décentralisées. Les jetons de récompense sur ETH sont appelés éthers.

4.3.3. MaidSafe

MaidSafe (*Massive Array of Internet Disks - Secure Access For Everyone*) [**57**] est un projet de système de stockage de données décentralisé vise à fournir une alternative sécurisée et privée aux systèmes de stockage de données centralisés. Le projet a été lancé en 2006 et est développé par la société écossaise *MaidSafe*. Le système de stockage de données *MaidSafe* utilise une technologie de réseau distribué qui permet aux utilisateurs de stocker des données de manière sécurisée et privée, sans avoir à faire confiance à un tiers centralisé pour stocker et protéger leurs données. Le réseau est conçu pour s'autogérer et s'auto-organiser, ce qui signifie que les utilisateurs qui fournissent de l'espace de stockage sur leurs ordinateurs sont également responsables de la maintenance du réseau.

4.3.4. Lisk

Lisk est une plateforme de BC décentralisée open-source a été fondée en 2016 par Max Kordek et Oliver Beddows [**58**]. Ce BC permet aux développeurs de créer et de déployer des applications décentralisées (*dApps*) sur un réseau de chaînes latérales (*sidechains*). La plateforme est basée sur un langage de programmation *JavaScript* familier aux développeurs, ce qui facilite le développement d'applications décentralisées pour les développeurs qui connaissent déjà JavaScript.

4.3.5. EOS

L'EOS [**59**] est une plate-forme BC décentralisée conçue pour permettre le développement et le déploiement de contrats intelligents et d'applications décentralisées (*dApps*). La plateforme a été lancée en juin 2018 par la société Block One. L'EOS utilise une approche BC déléguée (*DPoS*) pour valider les transactions. Le système DPoS permet à un petit groupe d'acteurs élus, appelés « *producteurs de blocs* », de valider les transactions et de maintenir les blocs, permettant de traiter un grand nombre de transactions en une seule fois.

4.4. Les niveaux de décentralisation

Il existe différents niveaux de décentralisation dans les systèmes basés sur la technologie BC. Le but de ces niveaux de décentralisation dans les technologies de BC est de créer des systèmes résilients, sécurisés et résistants à la censure, tout en garantissant que les parties prenantes peuvent participer aux processus décisionnels. Il existe plusieurs niveaux de décentralisation qui peuvent être appliqués aux technologies de BC.

4.4.1. Décentralisation complète

Ce niveau de décentralisation se caractérise par l'absence d'un point central de contrôle ou de gouvernance. La prise de décision est partagée entre différents nœuds BC, ce qui rend la plateforme très résiliente et résistante à la censure. L'Ethereum est un exemple de BC complètement décentralisé.

4.4.2. Décentralisation partielle

Ce niveau de décentralisation se caractérise par l'existence d'un point central pour des raisons pratiques, mais les tâches sont réparties entre plusieurs nœuds. Par exemple, dans un BC de preuve de travail, les mineurs sont les principaux acteurs qui valident les transactions et maintiennent le BC, mais il peut y avoir une centralisation dans la distribution des récompenses minières.

4.4.3. Décentralisation hybride

Avec ce niveau de décentralisation, certaines parties du BC sont décentralisées, tandis que d'autres sont centralisées. Par exemple, dans une BC avec preuve de participation, les détenteurs de jetons peuvent voter pour les validateurs de blocs, mais il peut y avoir une centralisation dans la distribution initiale des jetons.

4.4.4. Décentralisation faible

Avec ce niveau de décentralisation, le pouvoir est fortement centralisé, mais il existe un certain niveau de participation communautaire. Les BC privés ou les BC d'entreprise sont des exemples de faible décentralisation, car le pouvoir est généralement concentré entre les mains de l'entreprise ou de l'organisation qui les exploite, mais les parties prenantes peuvent participer aux processus de prise de décision.

5. Modèles de sécurité l'IoT

Les modèles de sécurité de l'IoT sont des outils qui aident à concevoir et à mettre en œuvre des mesures de sécurité efficaces pour les dispositifs IoT. Ils permettent de prendre en compte les différents aspects de la sécurité de l'IoT, tels que la sécurité des dispositifs, la sécurité des communications, la gestion des identités et des accès, la sécurité des données, etc. Parmi les modèles de sécurité les plus connus, deux modèles qui, ensemble, donnent une bonne image générale de ce qu'est la sécurité de l'IoT: le triangle DIC et le cadre de cyber sécurité du NIST [**60**]. Ces modèles offrent une vue d'ensemble complète de la sécurité IoT et aident les organisations à identifier les risques potentiels, mettre en place des mesures de sécurité adéquates pour protéger les dispositifs IoT et les données qu'ils traitent. Il est important de souligner que la sécurité de l'IoT est un enjeu crucial, car les dispositifs IoT sont de plus en plus présents dans notre vie quotidienne et qu'ils traitent des données personnelles sensibles.

5.1. Le triangle DIC

Le triangle DIC (*Device, Identity, Connectivity*) est un modèle qui décrit les trois aspects fondamentaux de la sécurité IoT.

- **Dispositif:** Il s'agit de la sécurité au niveau du dispositif, qui inclut l'authentification, le chiffrement, la protection contre les attaques physiques, la sécurité des données, etc.
- **Identité:** Il s'agit de la sécurité au niveau de l'identité, qui comprend l'authentification des utilisateurs et des dispositifs, la gestion des accès, la confidentialité des données, etc.
- **Connectivité:** Il s'agit de la sécurité au niveau de la connectivité, qui concerne la protection du réseau, la sécurité de la couche transport, la gestion des clés, la protection des communications, etc.

Le modèle du triangle DIC permet donc d'avoir une vision globale de la sécurité IoT, en prenant en compte les différents aspects qui doivent être protégés pour assurer une sécurité efficace.

5.2. Le cadre de cyber sécurité du NIST

Le cadre de cyber sécurité du NIST (*National Institute of Standards and Technology*) est un autre modèle important pour la sécurité IoT. Ce cadre est organisé en cinq domaines principaux:

- **Identification:** Identification des systèmes, des données, des personnes et des processus pour assurer une sécurité adéquate.
- **Protection:** Protection des systèmes et des données grâce à des mesures de sécurité telles que le chiffrement, la gestion des accès, la sécurité physique, etc.
- **Détection:** Détection des menaces et des incidents de sécurité pour permettre une réponse rapide et efficace.
- **Réponse:** réponse aux incidents de sécurité pour minimiser les dommages et restaurer les systèmes à leur état normal.
- **Récupération:** récupération après un incident de sécurité pour rétablir les systèmes et les processus à leur état normal.

6. Travaux connexes

Konstantinos Christidis et al., [**61**] ont décrit comment les contrats intelligents de la Blockchain peuvent faciliter et soutenir le flux de travail autonome et le partage de services entre les appareils IoT, aussi ils ont expliqué comment l'IoT peut bénéficier des réseaux de Blockchain dans les aspects liés à la facturation, au commerce électronique, à l'expédition et à la gestion de la chaîne d'approvisionnement.

Mayra Samaniego et al., [**62**] la capacité de la Blockchain à créer/stocker/transférer des actifs numériques de manière distribuée, décentralisée et inviolable était d'une grande valeur pratique pour les systèmes IoT. Ils ont considéré qu'un défi clé dans le déploiement de Blockchain en tant que service (BaaS) pour l'IoT était l'environnement d'hébergement. Ils ont évalué l'utilisation du nuage ou du brouillard comme plate-forme pour relever ce défi.

Arshdeep Bahga et al., [**63**] ont proposé un framework basé sur la Blockchain pour l'IoT industriel (ou IIoT). Le framework permet aux appareils IIoT de communiquer avec le cloud ainsi qu'avec le réseau Blockchain. Chaque appareil IIoT est équipé d'un ordinateur monocarte (SBC) doté de capacités d'interface de contrôle et de communication pour le cloud et la Blockchain Ethereum. Les appareils IIoT sont conçus pour envoyer des données vers le cloud à des fins de stockage et d'analyse, et envoyer/recevoir des transactions vers d'autres appareils sur le réseau Blockchain, ainsi que pour déclencher l'exécution de contrats intelligents.

MicheleRuta et al., [**64**] ont proposé une architecture orientée services qui utilise la BC pour les opérations d'enregistrement, de découverte, de sélection et de paiement. Ces opérations sont implémentées sous forme de smart contrats permettant une exécution distribuée et fiable.

Pelin Angin et al., [**65**] ont proposé un framework de sécurité des données décentralisé pour l'IoT, adaptable à la nature et au contexte même des applications et des appareils IoT. Le framework a pour objectif de permettre la création d'un écosystème IoT sécurisé grâce à la normalisation et à l'interopérabilité, ouvrant la voie à de nouvelles recherches interdisciplinaires impliquant des domaines allant des véhicules autonomes aux réseaux de distribution d'énergie intelligents, supprimant la barrière de sécurité d'une adoption plus large des appareils compatibles IoT.

Yongfeng Qian et al., [**66**] ont proposé un schéma basé sur la BC pour sécuriser et gérer plusieurs appareils IoT. Le schéma proposé permet d'analyser les problèmes de sécurité des applications, du réseau et des couches. De plus, ils ont adopté un dispositif d'identification basé sur un algorithme pour connecter la BC et les dispositifs IoT afin de garantir la fiabilité et la sécurité.

Rahul Agrawal et al., [**67**] ont présenté un mécanisme de BC pour la sécurité IoT basé sur la fonction de décentralisation de la BC pour permettre une sécurité continue du système IoT sans intervention de l'utilisateur.

Ana Reyna et al., [**68**] ont analysé les défis et les opportunités lors de l'intégration de la Blockchain et de l'IoT. Les opportunités offertes par la Blockchain comprenaient la confiance et la confiance dans les environnements distribués sans la troisième autorité. Les défis comprenaient la capacité de stockage, l'évolutivité, la sécurité et l'anonymat.

Ayesha Altaf et al., [**69**] ont expliqué le modèle de confiance pour le système IoT et les défis existent dans la gestion de la confiance. Comme les appareils IoT sont des appareils bas de gamme ayant une faible puissance de traitement et moins de stockage de mémoire, le traitement peut être effectué à l'aide du concept de Fog Computing, car il offre une capacité de traitement et de stockage.

Oscar Novo [**70**] a proposé une architecture de preuve de concept basée sur une approche BC pour développer un système de gestion des accès aux ressources IoT. Il a conçu une politique décentralisée pour le système qui stockait les données dans la technologie BC.

7. Conclusion

Nous avons présenté dans ce chapitre la sécurité distribuée dans les environnements IoT, il est important de souligner l'importance croissante de la sécurité dans les réseaux Internet des Objets. En effet, avec l'expansion rapide de l'IoT et l'augmentation du nombre d'appareils connectés, les risques de cyber attaques ont également augmenté, c'est pourquoi les systèmes de sécurité décentralisés sont des solutions prometteuses à ces problèmes. , car ils contribuent à décentraliser les capacités de sécurité et à réduire les risques de centralisation excessive. Ces systèmes peuvent également contribuer à renforcer la confidentialité et l'intégrité des données en utilisant des protocoles de cryptage et de vérification.

Chapitre 3: Architecture de sécurité basée sur la Blockchain pour les politiques des services dans un Smart Home

1. Introduction

Les maisons intelligentes, également connues sous le nom de Smart Homes, sont de plus en plus appréciées en raison de leur capacité à automatiser les tâches domestiques et à améliorer le confort et la sécurité des résidents. Cependant, avec la prolifération des objets connectés et des services intelligents, la question de la sécurité et de la protection des données est devenue une préoccupation majeure.

Comme nous avons discutés dans les chapitres précédents, la technologie du BC offre une solution prometteuse pour garantir la sécurité des transactions et des échanges de données dans un environnement décentralisé. Dans ce chapitre, nous proposons une architecture basée sur la BC pour renforcer la sécurité des politiques de services dans un Smart Home. L'avantage de la proposition permet au propriétaire d'appliqué les politiques de contrôle locale de leur maison et de gérer à distance leur accès, elle intègre également les politiques d'intervention via des API Cloud.

Dans un premier temps, nous décrivons une architecture qui modélise les stratégies de stockage et de traitement d'un smart home. Ensuite, nous allons présenter l'architecture que nous proposons pour la sécurité des politiques des services smart home. Ainsi que, le smart contrat qui permet de garantir l'intégrité des politiques pour un accès à distance des propriétaires et empêche toute manipulation non autorisée.

2. Modélisation des politiques des services dans un Smart Home

Dans le cadre de la modélisation des politiques d'un Smart Home, différents dispositifs connectés, tels que les caméras de sécurité, les capteurs de température, les systèmes de contrôle d'éclairage, les assistants vocaux et les thermostats intelligents, collectent de manière continue des données. Ces données sont ensuite stockées dans des dispositifs de stockage dédiés. Généralement, les appareils d'un smart home sont connectés à Internet, ce qui les expose à des vulnérabilités telles que les attaques par déni de service distribué (*DDoS*), les intrusions dans le réseau et les logiciels malveillants. Cela nécessite un modèle de sécurité renforcé pour protéger ces politiques contre ces vulnérabilités. À cet égard, nous proposons une architecture IoT intelligente et connectée pour garantir la sécurité des politiques d'un smart home. La Figure 4 ci-dessous illustre deux stratégies essentielles pour la modélisation des politiques locaux d'un smart home, à savoir la stratégie du stockage des données provenant de l'IoT et la stratégie du traitement de ces données.

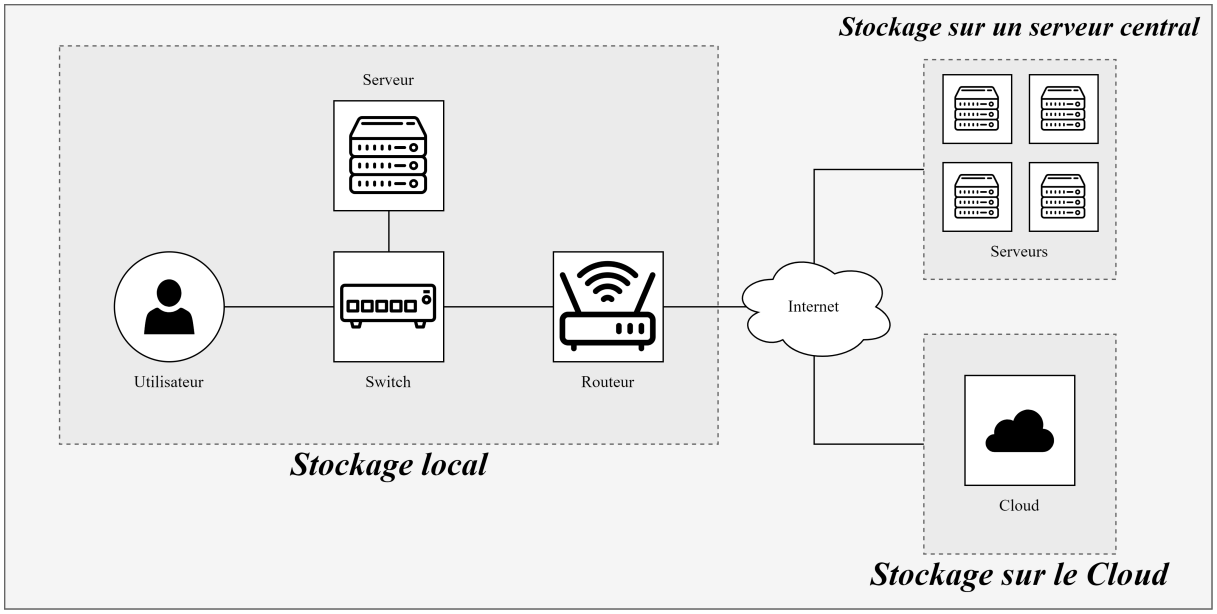


Figure 4: Les stratégies de stockage et de traitement des données collectées d'un Smart Home.

2.1. Les stratégies de stockage

2.1.1. Le Stockage local

Les stratégies de stockage des données IoT dans un smart home sont des règles et des pratiques mises en place pour gérer et sécuriser les données collectées par les dispositifs connectés de la maison. En particulier, les politiques locales de stockage consistent à stocker les données collectées par ces dispositifs connectés directement sur un ensemble des appareils localisé dans la maison plutôt que un stockage distant. De type de stockage expose des défis de sécurité et de confidentialité des données.

2.1.2. Stockage sur un serveur central

Il est également possible de stocker les données collectées par les appareils intelligents sur un serveur central distant. Dans le cas des grandes maisons intelligentes ou des complexes d'appartements, les serveurs centraux sont couramment utilisés pour centraliser le stockage des données provenant de tous les appareils intelligents de l'ensemble immobilier.

2.1.3. Le Stockage dans le Cloud

Il est également possible de stocker les données collectées par les appareils intelligents sur le Cloud, une infrastructure distante accessible via Internet. Les services Cloud tels qu'Amazon Web Services, Google Cloud Platform et Microsoft Azure offrent des solutions pour stocker, gérer et traiter les données collectées, et des Cloud publique /privé. Cela permet un accès pratique et sécurisé aux données depuis n'importe où pour le propriétaire, sans dépendre de l'espace de stockage local des appareils.

2.2. Le traitement

2.2.1. Le traitement local

En d'autres termes, les informations sont traitées localement sur des dispositifs intelligents tels que les thermostats intelligents ou les caméras de sécurité. Cela présente l'avantage d'un traitement rapide et de l'autonomie des appareils, mais peut également restreindre la capacité de traitement en fonction des capacités des appareils.

2.2.2. Traitement en Cloud

Une autre approche pour le traitement des données IoT d'un smart home consiste à envoyer ces données vers le Cloud, où elles sont traitées à l'aide de services Cloud. Cette méthode offre une puissance de traitement accrue, car les ressources du Cloud sont généralement plus importantes que celles des appareils domestiques. Cela permet de renforcéla sécurité des politiqueset d'appliquer les structures de BC centralisé sur les données collectées.

2.2.3. Traitement distribué

Le traitement des données dans un smart home implique une approche de traitement distribué, où les données sont traitées à la fois localement sur les appareils intelligents et dans le Cloud en utilisant des systèmes distribués. Cette méthode combine les avantages des deux approches précédemment mentionnées.

En traitant les données localement, sur les appareils intelligents tels que les thermostats intelligents ou les caméras de sécurité, on bénéficie d'un traitement rapide des données en utilisant les ressources disponibles localement. Cela permet de minimiser la latence et de garantir une réponse en temps réel pour les actions qui nécessitent une prise de décision immédiate.

3. L'intégrité des politiques d'un Smart Home

Assurer l'intégrité des politiques des services dans un smart Home est crucial pour instaurer la confiance et la fiabilité des services. Cependant, cette tâche présente plusieurs défis qui nécessitent une attention particulière. Il est essentiel de protéger les politiques des services contre les altérations non autorisées, les attaques visant leur intégrité, les erreurs humaines et les vulnérabilités liées aux communications. De plus, la gestion des mises à jour des politiques doit être effectuée de manière sécurisée afin de préserver leur intégrité tout au long du processus. En surmontant ces défis, il est possible d'établir des mécanismes solides qui garantissent l'intégrité des politiques des services, assurant ainsi la confiance et la fiabilité des opérations dans un Smart Home.

3.1. Défis et exigences

Les défis et les exigences liés à l'intégrité des politiques des services dans un smart home représentent des aspects cruciaux à prendre en compte pour assurer la confiance et la fiabilité du système, le tableau suivant montre un ensemble des défis liés à l'intégrité des politiques et les exigences de leur sécurité:

Défi	Exigences
Protection contre les attaques d'intégrité	Mettre en place des mécanismes de sécurité robustes pour détecter et prévenir les attaques visant à altérer les politiques des services, tels que les attaques de manipulation de données ou les attaques par rejeu
Sécurité des canaux de communication	Assurer la sécurité des canaux de communication utilisés pour transmettre les politiques des services, en utilisant des protocoles de communication sécurisés pour éviter les interceptions ou les altérations indésirables
Vérification de l'intégrité	Disposer de mécanismes permettant de vérifier l'intégrité des politiques des services, tels que l'utilisation de techniques de hachage cryptographique, de signatures numériques ou de preuves de vérifiabilité
Protection contre les erreurs humaines	Mettre en place des mesures de sécurité pour réduire les risques d'altérations involontaires des politiques des services causées par des erreurs humaines, et détecter rapidement toute altération non autorisée

Tableau 2: Les défis et les exigences liés à l'intégrité des politiques des services dans le Smart Home

En tenant compte de ces défis et en respectant ces exigences, il est possible de garantir une intégrité solide des politiques des services dans un Smart Home, ce qui contribue à instaurer la confiance et la fiabilité du système.

3.2. Les types d'attaques

Il existe différents types d'attaques potentielles dans les Smart Homes, qui peuvent compromettre la sécurité des occupants de la maison et des données personnelles collectées par les dispositifs connectés.

3.2.1. Attaques par déni de service (DDoS)

Les attaques DDoS consistent à saturer un système ou un réseau avec un trafic de données excessif, ce qui peut entraîner une interruption de service pour les occupants de la maison.

3.2.2. Attaques de phishing

Les attaques de phishing impliquent l'envoi de messages frauduleux aux occupants de la maison, dans le but de les inciter à divulguer des informations personnelles ou à télécharger des logiciels malveillants.

3.2.3. Attaques de reconnaissance

Les attaques de reconnaissance consistent à collecter des informations sur les occupants de la maison, leurs habitudes et leurs dispositifs connectés, dans le but de planifier des attaques plus sophistiquées.

3.2.4. Attaques de logiciels malveillants

Les logiciels malveillants peuvent être introduits dans les dispositifs connectés des Smart Homes, ce qui peut permettre aux pirates informatiques de prendre le contrôle de ces dispositifs et d'accéder aux données personnelles des occupants de la maison.

3.2.5. Attaques d'usurpation d'identité

Les pirates informatiques peuvent usurper l'identité des occupants de la maison en utilisant leurs données personnelles pour accéder à leurs comptes en ligne et voler des informations sensibles.

4. Architecture IoT sécurisé basés BC pour les politiques des services

Pour prévenir les risques potentiels précédents de sécurité d'un smart home, il est crucial de mettre en œuvre une sécurité robuste, toute en assurant le chiffrement des données et les communications, la mise à jour régulière des appareils et des services, la vigilance quant à l'utilisation de services tiers sécurisés. Pour cela, nous proposons une architecture de sécurité des politiques intelligents et connecté, la figure ci-dessous montre l'architecture que nous proposons. En particulier, elle est basée sur l'intégration de la technologie de BC pour offrir plusieurs avantages, tels que la transparence, l'immuabilité et la décentralisation.

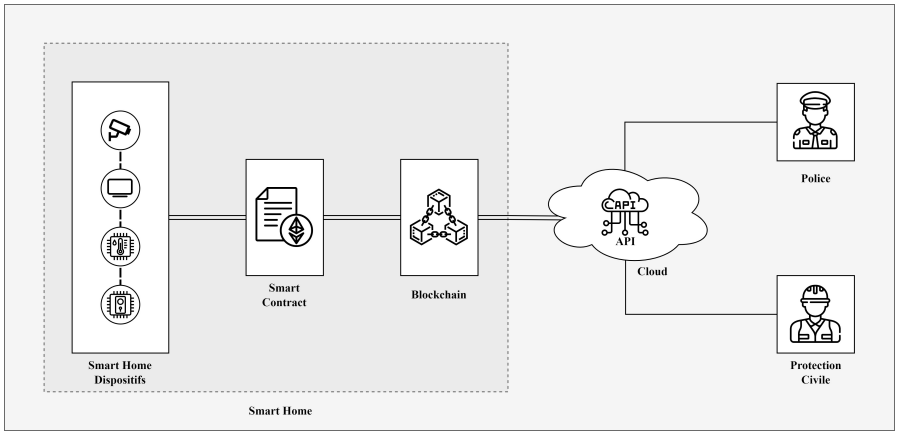


Figure 5: Architecture de sécurité basée BC pour la sécurité des politiques

Avec cette architecture, nous explorerons comment la technologie BC et l'intégration API peuvent être utilisée pour améliorer la sécurité d'une maison intelligente (Smart Home). Plus précisément, nous examinerons comment l'utilisation de la BC peut fournir un enregistrement sécurisé et inaltérable de tous les services de politique d'un Smart Home, tandis que l'intégration API peut être utilisée pour alerter automatiquement la police ou la protection civile en cas d'activité suspecte.

- Le système de Smart Home est équipé de capteurs et d'autres appareils intelligents qui peuvent surveiller l'activité telle que les mouvements, l'ouverture des portes et des fenêtres et les changements de température et d'humidité.
- Le système de maison intelligente est connecté à un réseau BC, qui sert de grand livre sécurisé et décentralisé pour enregistrer les politiques.
- Le système de Smart Home est intégré à une API qui peut transmettre automatiquement des données à la police ou à la protection civile si certaines conditions des politiques sont remplies.
- L'accès au réseau BC sera limitées aux seuls personnels autorisés, et toutes les activités sur le réseau seront surveillées et auditées pour détecter toute violation de sécurité potentielle.

4.1. Les éléments de l'architecture

L'architecture que nous proposons englobe les éléments suivants:

4.1.1. BC Locale

L'idée de base de notre solution réside dans la BC locale, qui offre la possibilité de stocker de manière immuable les politiques de service de notre Smart Home. Ce type deBCest conçu de manière décentralisée, garantissant ainsi une disponibilité élevée et une résistance aux attaques.

4.1.2. Smart Contracts (Contrats intelligents)

Les contrats intelligents sont utilisés pour automatiser les interactions entre les dispositifs connectés et les utilisateurs de notre Smart Home. Ils permettent de mettre en place des politiques de sécurité strictes qui sont exécutées de manière autonome par la BC.

4.1.3. Utilisation d'un registre distribué

La BC serve un registre distribué pour enregistrer et valider les politiques de sécurité du Smart Home. Chaque politique peut être enregistrée sous forme de transaction dans un bloc, et chaque bloc est lié de manière cryptographique pour former une chaîne de blocs immuable.

4.1.4. Stockage des politiques de sécurité

Les politiques de sécurité sont stockées dans les contrats intelligents de la BC. Les utilisateurs autorisés peuvent modifier ces politiques de sécurité à tout moment, mais ces modifications doivent être validées par la communauté des utilisateurs.

4.1.5. Autorisations des utilisateurs

Les autorisations des utilisateurs de Smart Home sont stockées dans la BC et vérifiées par les contrats intelligents avant chaque interaction avec un dispositif connecté. Les utilisateurs peuvent gérer leurs autorisations en temps réel via une interface dédiée ou une application.

4.2. Structure de BC

La structure de BC pour les politiques des services de Smart Home est conçue pour fournir un moyen sécurisé et décentralisé d'enregistrer, gérer et appliquer les politiques. La BC est composée de blocs contenant les politiques spécifiques, telles que les autorisations d'accès, les configurations des appareils et les règles de confidentialité. Chaque bloc est identifié de manière unique et horodatée pour assurer l'intégrité et la chronologie des politiques enregistrées. Grâce à la nature décentralisée de la BC, les politiques de smart home peuvent être vérifiées et appliquées de manière transparente, offrant ainsi un niveau supplémentaire de confiance et de sécurité aux propriétaires de maisons intelligentes.

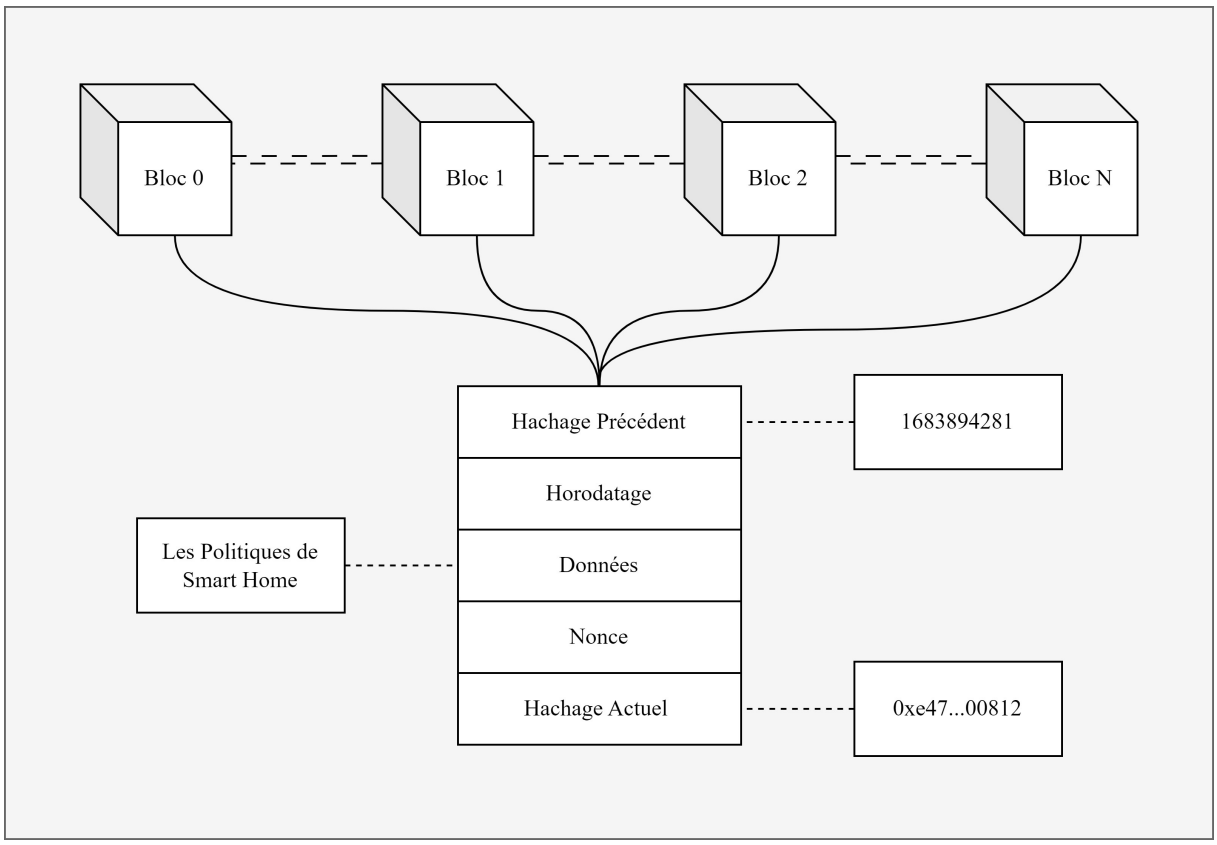


Figure 6: Structure de BC

4.3. Smart Contract

Le Smart Contract est un programme informatique autonome qui exécute automatiquement les termes d'un contrat prédéfini lorsque les conditions spécifiées sont remplies. Il utilise la technologie de la BC pour assurer la transparence, la sécurité et l'immutabilité des transactions. Les Smart Contracts éliminent le besoin d'intermédiaires et permettent l'exécution automatisée de diverses activités, telles que les transactions financières, la gestion des actifs numériques et les échanges décentralisés. Ils offrent des opportunités pour l'automatisation des processus commerciaux et la création de nouveaux modèles économiques. L'ensemble du cycle de vie des contrats intelligents se compose de quatre phases consécutives:

4.3.1. Création de contrats intelligents

Plusieurs parties impliquées négocient d'abord sur les obligations, les droits et les interdictions des contrats. Après plusieurs cycles de discussions et de négociations, un accord peut arriver. Après cela rédiger un accord contractuel initial. Ensuite cet accord écrit en langages naturels devient un contrat intelligent écrit en langages informatiques, y compris les langages déclaratifs et les langages de règles basés sur la logique. Semblable au développement de logiciels informatiques, la procédure de conversion de contrat intelligent est composée de conception, de mise en œuvre et de validation (c'est-à-dire de test). Il convient de mentionner que la création de contrats intelligents est un processus itératif impliquant de multiples cycles de négociations et d'itérations. Parallèlement, il est également impliqué avec plusieurs parties, telles que des parties prenantes, des avocats et des ingénieurs en logiciel.

4.3.2. Déploiement de contrats intelligents

Les contrats intelligents validés peuvent ensuite être déployés sur des plateformes de BC. Les contrats stockés sur les BC ne peuvent pas être modifiés en raison de l'immuabilité des BC. Toute modification nécessite la création d'un nouveau contrat. Une fois les contrats intelligents déployés sur les BC, toutes les parties peuvent accéder aux contrats via les BC. De plus, les actifs numériques des deux parties impliquées dans le contrat intelligent sont verrouillés via le gel des portefeuilles numériques correspondants. Pendant ce temps, les parties peuvent être identifiées par leur portefeuille numérique.

4.3.3. Exécution de contrats intelligents

Après le déploiement des Smart Contracts, les clauses contractuelles ont été suivies et évaluées. Une fois les conditions contractuelles atteintes, les procédures (ou fonctions) contractuelles seront automatiquement exécutées. Il convient de noter qu'un contrat intelligent consiste en un certain nombre d'instructions déclaratives avec des connexions logiques.

Lorsqu'une condition est déclenchée, la déclaration correspondante sera automatiquement exécutée, par conséquent une transaction sera exécutée et validée par les mineurs dans les BC. Les transactions validées et les états mis à jour ont été stockés sur les BC par la suite.

4.3.4. Achèvement de contrats intelligents

Après l'exécution d'un contrat intelligent, les nouveaux états de toutes les parties impliquées sont mis à jour. En conséquence, les transactions lors de l'exécution des contrats intelligents ainsi que les états mis à jour sont stockés dans des BC. Pendant ce temps, les actifs numériques ont été transférés d'une partie à une autre. Par conséquent, les actifs numériques des parties impliquées ont été déverrouillés. Le contrat intelligent a alors terminé tout le cycle de vie.

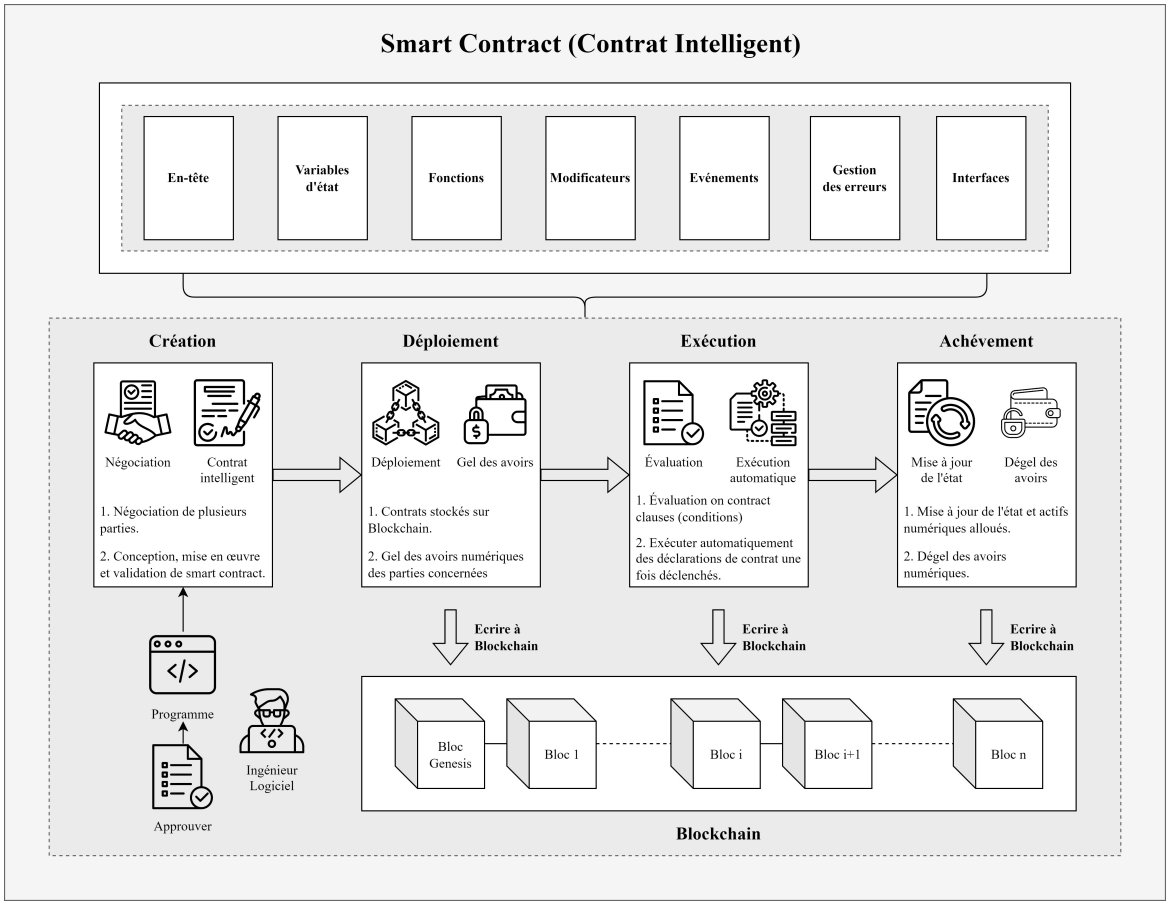


Figure 7: Le contrat intelligent du l'architecture proposé

Les contrats intelligents sont utilisés dans une maison intelligente (Smart Home) pour automatiser et sécuriser diverses fonctionnalités.

Dans notre scénario nous allons utiliser des Smart Contracts pour assurer la sécurité des politiques de service de smart home. Le 1er Smart Contract peut également être utilisé pour l'authentification des utilisateurs de notre maison intelligente, et un autre Smart Contract pour la réglementation des politiques de Smart Home.

5. Conclusion

Dans ce chapitre, nous avons présenté de manière détaillée notre architecture visant à sécuriser les politiques des services d'un Smart Home en utilisant une structure basée sur la BC et en intégrant des Smart Contracts. Les politiques enregistrées dans la BC sont immuables, ce qui signifie qu'elles ne peuvent pas être modifiées ou altérées une fois enregistrées. Le smart contrat proposé garantit l'intégrité des politiques pour un accès à distance des propriétaires et empêche toute manipulation non autorisée via des API Cloud et les services d'intervention (police/protection civile), renforçant ainsi la confiance et la fiabilité des services smart home.

Le chapitre suivant se concentrera sur l'implémentation et l'évaluation de notre scénario de Smart Home.

Chapitre 4: Implémentation et évaluation

1. Introduction

Dans le chapitre précédent, nous avons présenté en détail la proposition de notre architecture Blockchain visant à renforcer la sécurité et la confidentialité des politiques de service d'une Smart Home. Dans ce chapitre, nous nous concentrerons sur la mise en œuvre pratique de cette architecture, ainsi que l'évaluation de ses performances.

L'objectif principal de ce chapitre est de démontrer la faisabilité et l'efficacité de notre architecture. Nous décrirons les différentes étapes de mise en œuvre, en mettant l'accent sur le choix des outils, des technologies et des langages de programmation appropriés pour développer notre système.

En particulier, nous évaluerons les performances de notre architecture en utilisant divers logiciels et le projet Arduino. Ces évaluations se concentreront sur plusieurs aspects clés, notamment la vitesse de traitement des transactions, l'évolutivité du système, la résilience aux attaques et la consommation d'énergie. Cette évaluation approfondie nous permettra de mieux comprendre les avantages et les limites de notre proposition.

2. Outils de développement et langages

2.1. VS code

VS Code [**71**] est un éditeur de code source populaire développé par Microsoft. Il offre une interface utilisateur intuitive et épurée, avec des fonctionnalités puissantes pour la programmation. Son écosystème d'extensions est vaste, offrant aux développeurs la possibilité d'ajouter des fonctionnalités supplémentaires et de personnaliser leur environnement de développement. Avec des capacités de débogage intégrées, une intégration transparente avec Git et un large éventail de langages de programmation pris en charge, VS Code est un choix privilégié pour de nombreux développeurs cherchant un éditeur polyvalent et extensible.

2.2. Node.js

Node.js [**72**] est un environnement d'exécution côté serveur construit sur le moteur JavaScript V8 de Chrome. Il permet aux développeurs de créer des applications réseau rapides et évolutives en utilisant JavaScript tant pour le côté client que pour le côté serveur. Node.js bénéficie également d'une vaste bibliothèque de modules appelée npm (*Node Package Manager*), qui offre un écosystème riche et dynamique pour les développeurs.

2.3. Truffle

Truffle [**73**] est un framework de développement populaire utilisé pour créer et déployer des applications décentralisées (*DApps*) sur la Blockchain Ethereum. Il fournit un ensemble d'outils et de bibliothèques qui simplifient le processus de développement, de test et de déploiement de contrats intelligents. Truffle facilite la création de projets Ethereum à partir de zéro en fournissant une structure de répertoire organisée et des configurations prédéfinies.

2.4. Ganache

Ganache [**74**] est un environnement de développement personnel destiné à la Blockchain Ethereum. Il fournit une simulation locale d'un réseau Ethereum, permettant aux développeurs de tester et de déployer leurs contrats intelligents de manière rapide et sécurisée. Ganache crée un réseau privé Ethereum avec des comptes préchargés de jetons virtuels, permettant aux développeurs de simuler des interactions entre les contrats intelligents et les utilisateurs. Cela offre un environnement isolé et contrôlé pour tester les fonctionnalités, les performances et les scénarios d'utilisation de leurs applications décentralisées.

2.5. MetaMask

MetaMask [**75**] est une extension de navigateur populaire qui permet aux utilisateurs d'accéder facilement à des applications décentralisées (*DApps*) basées sur la Blockchain Ethereum. Il agit comme un portefeuille de cryptomonnaie et une interface utilisateur pour interagir avec les contrats intelligents et les tokens Ethereum. MetaMask simplifie le processus de gestion des clés privées et des transactions cryptographiques en offrant une interface conviviale et sécurisée. Il permet aux utilisateurs de créer des comptes Ethereum, d'importer des portefeuilles existants et de gérer leurs actifs numériques.

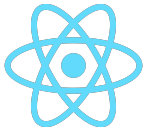
2.6. Solidity

Solidity [**76**] est un langage de programmation spécialement conçu pour écrire des contrats intelligents sur la Blockchain Ethereum. Il offre aux développeurs une syntaxe similaire à celle de JavaScript et C++. Solidity permet de définir la logique et le comportement des contrats intelligents, qui sont des programmes autonomes exécutés sur la Blockchain Ethereum.

2.7. Javascript

JavaScript (*JS*) [**77**] est un langage de programmation polyvalent et largement utilisé dans le développement web. Il permet d'ajouter de l'interactivité et de la dynamique aux pages web en manipulant le contenu HTML, en gérant des événements utilisateur, en effectuant des requêtes réseau, en manipulant des données et en créant des interfaces utilisateur réactives. JS est interprété par les navigateurs web, ce qui signifie qu'il s'exécute directement sur l'ordinateur de l'utilisateur sans nécessiter de compilation préalable.

2.8. React JS



React JS [**78**] est une bibliothèque JavaScript open source développée par Facebook. Elle est utilisée pour créer des interfaces utilisateur interactives et réactives pour les applications web. React JS est basé sur le concept de composants réutilisables, ce qui permet de diviser l'interface utilisateur en petites pièces autonomes. Ces composants peuvent être créés individuellement et combinés pour construire des interfaces complexes. React JS utilise également JSX, une syntaxe qui permet d'écrire du code HTML directement dans les fichiers JavaScript, facilitant ainsi la création d'interfaces utilisateur et la manipulation des données.

2.9. Arduino IDE

Arduino IDE (*Integrated Development Environment*) [**79**] est un environnement de développement open source spécialement conçu pour la programmation des cartes Arduino. Il fournit une interface conviviale et intuitive permettant aux utilisateurs de développer et de télécharger des programmes sur les cartes Arduino. Arduino IDE utilise le langage de programmation basé sur Wiring, qui est facile à apprendre et à utiliser, même pour les débutants en programmation.

2.10.Bootstrap

Bootstrap [**80**] est un framework front-end open source largement utilisé pour la création de sites web réactifs et esthétiquement attrayants. Il fournit une collection de composants prédéfinis, de styles CSS et de scripts JavaScript qui facilitent la conception et le développement de sites web modernes.

2.11.Font Awesome

Font Awesome [**81**] est une bibliothèque d'icônes open source largement utilisée dans le développement web. Elle offre une vaste collection d'icônes vectorielles, qui peuvent être facilement intégrées et stylisées dans les sites web et les applications. Les icônes de Font Awesome sont basées sur des polices de caractères, ce qui signifie qu'elles sont extensibles et personnalisables à l'aide de CSS. Elles sont également compatibles avec les principaux frameworks front-end tels que Bootstrap, ce qui facilite leur intégration dans les projets.

3. Description du système

Notre système vise à concevoir une application pour sécuriser les politiques de services dans un Smart Home en utilisant la technologie de la BC. Le but est de garantir l'intégrité, la confidentialité et la traçabilité des politiques de services dans un Smart Home. Le système présent une application décentralisée permet d'automatiser et de contrôler divers politiques de la maison. Nous intégrant un prototype Arduino pour créer des scénarios personnalisés pour l'automatisation de la sécurité des politiques quotidiennes (voir la figure ci-dessous), cette intégration offre les avantages suivants:

1. Connecter les capteurs et les actionneurs par ce prototype permet de collecté des données en temps réel à partir de capteurs et les transmettent à une application décentralisée IoT. Cela permet de surveiller des variables telles que la température, l'humidité, la luminosité, mouvement, etc. Les données collectées peuvent ensuite être protégé par le Blockchain pour sécuriser le déclenchement automatique des politiques.
2. Automatiser le contrôle de divers paramètres liés au déclenchement des politiques, ce qui permet le contrôle à distance des politiques telles que le contrôle d'accès contre le vol et la prévention des dangers.

L'architecture du prototype Arduino pour la sécurité d'une maison intelligente comprend plusieurs modules qui interagissent pour mettre en œuvre des fonctionnalités avancées d'automatisation, de surveillance, de sécurité et d'interaction conviviale dans le cadre d'une maison intelligente. Le rôle de chaque module est résumé comme suit :

Module Arduino: Ce module constitue le cœur du système et est responsable de la collecte des données provenant de différents capteurs et de l'interaction avec les autres modules. Il agit comme un microcontrôleur programmable qui exécute les tâches de contrôle et de communication.

Module de contrôle des variables IoT: Ce module est responsable de la communication entre le système Arduino et les périphériques intelligents de la maison. Il permet de contrôler à distance les dispositifs connectés et transmit les informations de contrôle au système de sécurité, etc. Il assure également la connectivité avec le réseau IoT pour l'échange de données.

Système de température et humidité: Ce module est responsable de la collecte précise des données de température et d'humidité, qui peuvent être utilisées pour prendre des décisions éclairées sur le contrôle de l'environnement.

Système d'alarme: Ce module est chargé de la surveillance de la sécurité de la maison. Il peut être composé de capteurs de mouvement, de détecteurs de fumée, de capteurs d'intrusion, etc. En cas d'événements anormaux, le système d'alarme peut déclencher des alertes pour informer les occupants ou les autorités compétentes.

Interface utilisateur: Ce module fournit une interface conviviale permettant aux utilisateurs d'interagir avec le système de la maison intelligente. Il peut s'agir d'un écran tactile, d'une application mobile ou d'une interface vocale utilisant des assistants virtuels. Les utilisateurs peuvent surveiller et contrôler les appareils connectés, configurer des scénarios d'automatisation, recevoir des notifications, etc.

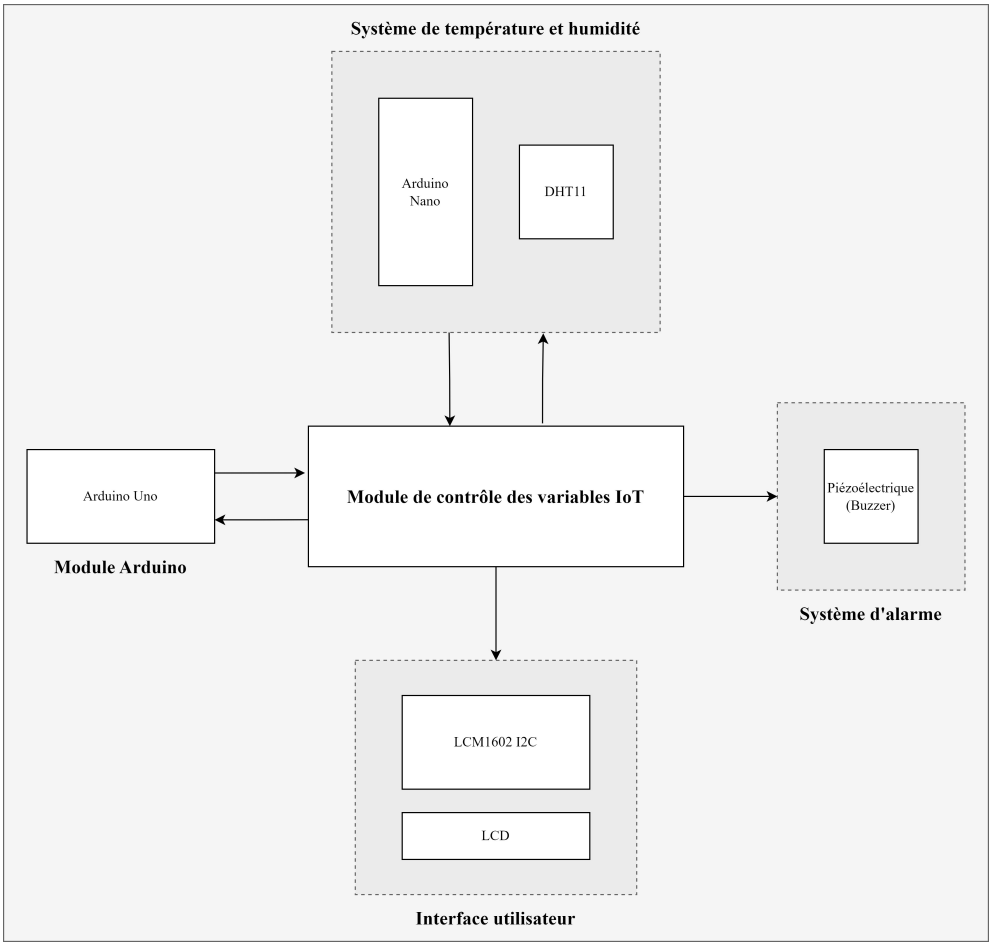


Figure 8: La représentation du prototype Arduino du système proposé

4. Configuration du système

Dans cette partie, nous allons présenter les étapes nécessaires pour développer notre système.

4.1. Les versions des composants

Les versions des composants qui ont été utilisées dans notre environnement de développement sont représentées dans le tableau suivant:

Composant	**Version**
Truffle	v5.8.4
Ganache	v7.8.0
Node	v18.16.0
Web3.js	v1.8.2

Tableau 3: Les versions des composants utilisées

4.2. Création de projet

La première étape est la création d'un nouveau répertoire pour notre projet à l'aide de l'invite de commande.

```
mkdir SmartHomeBC
```

Accédons-y à notre répertoire de projets « *SmartHomeBC* » à l'aide de l'invite de commande.

```
cd SmartHomeBC
```

Dans cette étape nous allons initialiser le projet Truffle dans le répertoire actuel.

```
truffle init
```

Après l'initialisation du projet Truffle nous allons configurer les paramètres de notre projet à partir du fichier « *truffle-config.js* », pour lier notre projet Truffle avec Ganache nous allons ajouter ce code dans le fichier de configuration.

```
networks: {
  development: {
    host: "127.0.0.1",
    port: 7545, // Port par défaut de Ganache
    network_id: "*", // Correspond à n'importe quel réseau ID
  },
},
```

4.3. Création des Smart Contracts

Dans notre système décentralisé nous allons utiliser deux Smart Contracts la 1er Smart Contract pour authentification des utilisateurs « *Auth.sol* » et la deuxiéme Smart Contract pour les capteurs IoT de notre système Smart Home « *SensorIoT.sol* ».

4.3.1. Contrat d'authentification

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.22 <0.9.0;

contract Auth {
  uint public userCount = 0;

  mapping(string => user) public usersList;

  struct user
  {
    string username;
    string email;
    string password;
  }

  event userCreated(string username, string email, string password);

  function createUser(string memory _username, string memory _email,
string memory _password) public
  {
      userCount++;
      usersList[_email] = user(_username, _email, _password);
      emit userCreated(_username, _email, _password);
    }
}
```

4.3.2. Contrat des capteurs IoT

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.22 <0.9.0;

contract SensorIOT {

  uint Temperature;
  uint Humidity;

  function setDATA(uint _Temperature, uint _Humidity) public {
        Temperature = _Temperature;
        Humidity = _Humidity;
  }

  function getDATA() public view returns (uint, uint) {
      return (Temperature, Humidity);
  }

  function Notification() public pure returns (string memory) {
    return "The temperature exceeds 60 degrees!!";
  }

}
```

4.3.3. Définition des migrations

Dans cette étape nous allons créer les fichiers de migrations de notre contrat.

```
truffle create migration Auth
```

```
 truffle create migration SensorIoT
```

Après l'exécution de deux commandes précédentes Deux fichiers Javascript seront créés dans le répertoire « *migrations* » qui définissent les étapes de déploiement de nos contrats.

```
const Auth = artifacts.require("./Auth.sol");

module.exports = function (deployer) {
  deployer.deploy(Auth);
};
```

```
const SensorIOT = artifacts.require("./SensorIOT.sol");

module.exports = function (deployer) {
  deployer.deploy(SensorIOT);
};
```

4.3.4. Déploiement les contrats

Pour le déploiement de nos contrats nous allons utiliser « migrate » cela exécutera les migrations et déploiera les contrats sur le réseau configuré.

```
truffle migrate --reset
```

4.4. Metamask et Ganache

Afin de connecter Metamask à Ganache, nous suivons les étapes suivantes:

- Nous ouvrons Metamask et cliquons sur l'icône du compte en haut à droite.
- Nous sélectionnons "Test réseau" dans la liste déroulante.
- Nous cliquons sur "Réseau personnalisé".
- Dans les paramètres réseau personnalisés, nous entrons un nom pour le réseau (par exemple "Ganache").
- Dans le champ "URL RPC", nous collons l'URL LAN Ethereum fournie par Ganache.
- Nous cliquons sur "Enregistrer" pour enregistrer les paramètres.
- Metamask se connectera automatiquement au réseau Ganache.

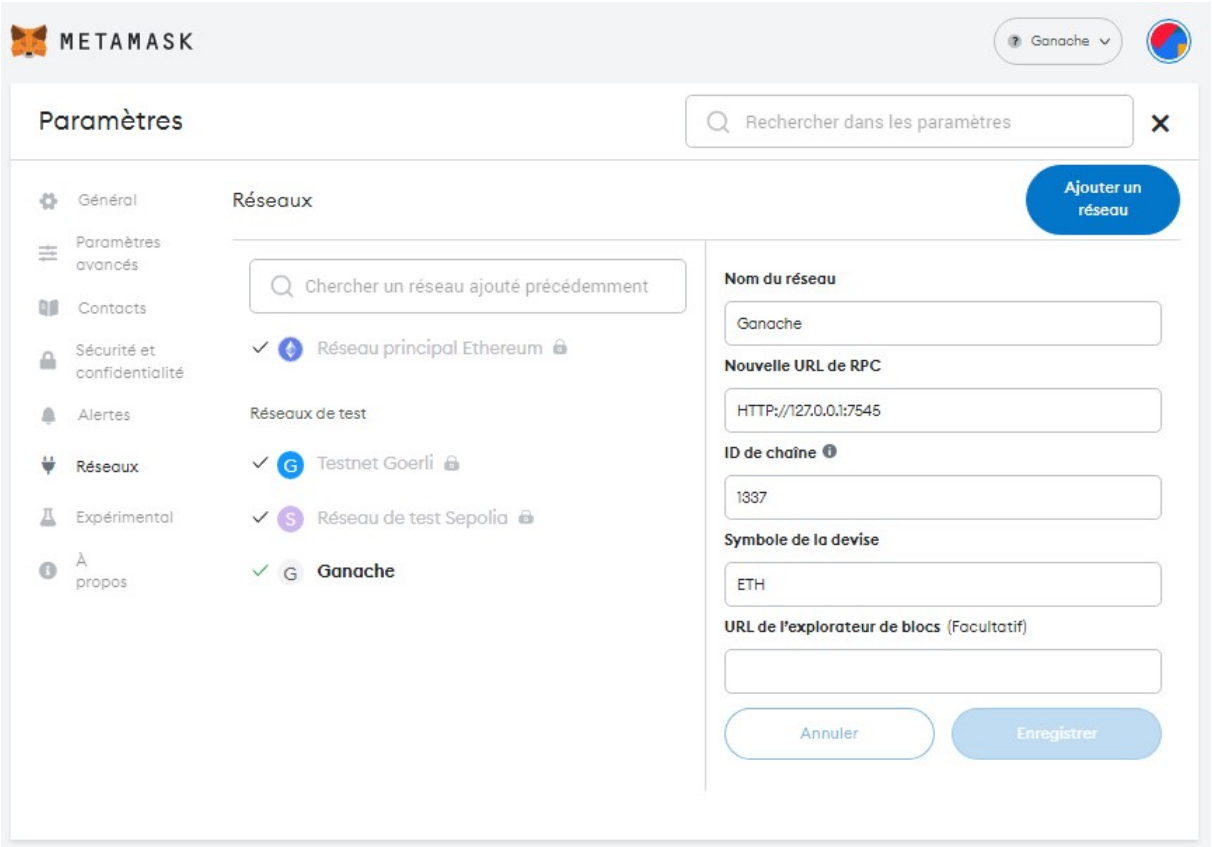


Figure 9: Détails du réseau de Metamask

4.5. Création de l'application

4.5.1. Application Serveur

Dans l'application serveur, notre appareil Arduino y sera directement connecté, car il mesurera les températures et collectera des informations sur notre maison intelligente. L'application sera également liée au système Blockchain et également liée à l'API.

La première étape est la création d'un nouveau répertoire pour notre serveur à l'aide de l'invite de commande.

```
mkdir Server
```

Accédons-y à notre projet « *Server* » à l'aide de l'invite de commande.

```
cd Server
```

Dans cette étape, nous allons créer d'un nouveau projet à l'aide de l'invite de commande.

```
npm init -y
```

Dans cette étape, nous allons installer les dépendances nécessaires pour interagir avec Web3 et Johnny-Five et Axios.

```
npm install web3
npm install johnny-five
npm install axios
```

Exemple 1: Le code ci-dessous pour importer les dépendances nécessaires.

```
const axios = require("axios");
const Web3 = require("web3");
const five = require("johnny-five");
```

Exemple 2: Le code ci-dessous pour connecter notre projet avec Ganache.

```
const SensorIOT = require("../app/src/build/contracts/SensorIOT.json");
const provider = new Web3.providers.HttpProvider("HTTP://127.0.0.1:7545");
const web3 = new Web3(provider);
const contract = new web3.eth.Contract(SensorIOT.abi,
SensorIOT.networks[5777].address);
```

Exemple 3: Le code ci-dessous pour récupérer les informations du capteur DHT11.

```
var five = require("johnny-five");
var board = new five.Board({ port: "COM3"});

board.on("ready", function() {
  var multi = new five.Multi({
    controller: "DHT11_I2C_NANO_BACKPACK"
  });

  multi.on("change", function() {
    console.log("Thermometer");
    console.log("Temperature Celsius: ", this.thermometer.celsius);
    console.log("--------------------------------------");

    console.log("Hygrometer");
    console.log("Humidity: ", this.hygrometer.relativeHumidity);
    console.log("--------------------------------------");
  });
});
```

Exemple 4: Le code ci-dessous pour envoyer les données au Blockchain.

```
async function setDATA(_Temperature, _Humidity) {
    const {contract} = state;
    await contract?.methods?.setDATA(_Temperature, _Humidity)?.send({from:
"0x614Dbc22E29Ecc16b1Cda9df53eB8a74C667FBe0"});
  }
```

Exemple 5: Le code ci-dessous pour envoyer les données au API.

```
const axios = require("axios");

let data = [{
    "Temperature": 21,
    "Humidity": 30
}];

axios.post('http://localhost:5000/api', data)
      .then(response => {
        console.log("added data to api");
      })
      .catch(error => {
        console.error(error);
      })
```

En fin nous allons starter le server à l'aide de l'invite de commande.

```
Node server.js
```

4.5.2. Application Client

Dans l'application client, notre application sera également liée au système Blockchain pour faire l'authentification des utilisateurs et également liée à l'API.

La première étape est la création d'un nouveau projet React à l'aide de l'invite de commande.

```
npx create-react-app Client
```

Accédons-y à notre React projet « *Client* » à l'aide de l'invite de commande.

```
cd Client
```

Dans cette étape, nous allons installer les dépendances nécessaires pour interagir avec Web3 et Axios.

```
npm install web3
npm install axios
```

Exemple 1: Le code ci-dessous c'est le code du fichier « *App.js* ».

```
import "./App.css";
import { BrowserRouter, Routes, Route, Navigate } from "react-router-dom";
import Login from "./views/Login";
import SignUp from "./views/SignUp";
import Home from "./views/Home";

function App() {
const email = localStorage.getItem("email");
return (
        <div className="App">
        <BrowserRouter>
                <Routes>
                <Route exact path="/" element={<Login />} />
                <Route path="/Signup" element={<SignUp />} />
                <Route
                        path="/Home"
                        element={email ? <Home /> : <Navigate to="/" />}
                />
                </Routes>
        </BrowserRouter>
        </div>
);
}

export default App;
```

Exemple 2: Le code ci-dessous c'est le code authentifie l'e-mail et le mot de passe saisis par l'utilisateur et accède à la page d'accueil.

```
const res = await auth.methods.usersList(email).call();

if (res.password === password) {
    localStorage.setItem("email", email);
    localStorage.setItem("account", accounts);
    navigate("/Home");
} else {
    alert("wrong user credentials or please signup");
}
}
```

Exemple 3: Le code ci-dessous c'est le code pour stocker les fonctions web3 utilisées pour charger l'adresse à partir de Metamask.

```
import Web3 from "web3/dist/web3.min.js";

import Auth from "../src/Auth.json";

export const loadWeb3 = async () => {
if (window.ethereum) {
        window.web3 = new Web3(window.ethereum);
        await window.ethereum.enable();
} else if (window.web3) {
        window.web3 = new Web3(window.web3.currentProvider);
} else {
        window.alert(
        "Non-Ethereum browser detected. You should consider trying
MetaMask!"
        );
}
};

export const loadBlockchainData = async () => {
const web3 = window.web3;
const accounts = await web3.eth.getAccounts();

const networkId = await web3.eth.net.getId();

if (networkId) {
        const auth = new web3.eth.Contract(
        Auth.abi,
        Auth.networks[networkId].address
        );
        return { auth, accounts: accounts[0] };
}
};
```

4.5.3. Application API

Dans l'application API, nous allons créer un simple API pour stocker les donnes sous forme un JSON.

La première étape est initializer d'un nouveau projet Node.js.

```
npm init -y
```

Dans cette étape nous allons installer Express.js.

```
npm install express
```

Le code ci-dessous c'est le code du fichier « *api.js* ».

```
const express = require('express');
const app = express();
app.use(express.json());

let data = [{
    "Temperature": 55,
    "Humidity": 30
}];

app.get('/api', (req, res) => {
  res.json(data);
});

app.post('/api', (req, res) => {
  const newData = req.body;
  data.push(newData);
  res.json(newData);
});

const port = 5000;
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

En fin nous allons starter le server API à l'aide de l'invite de commande.

```
Node api.js
```

5. Les interfaces

Dans cette partie, nous présentons quelques captures d'écran de quelques interfaces de notre projet.

L'une des interfaces les plus importantes se situe au niveau du projet « *Client* »:

- SignUp.js (Page d'inscription)
- Login.js (Page de connexion)
- Home.js (Page d'accueil)

Figure 10: Page d'inscription

Figure 11: Page de connexion

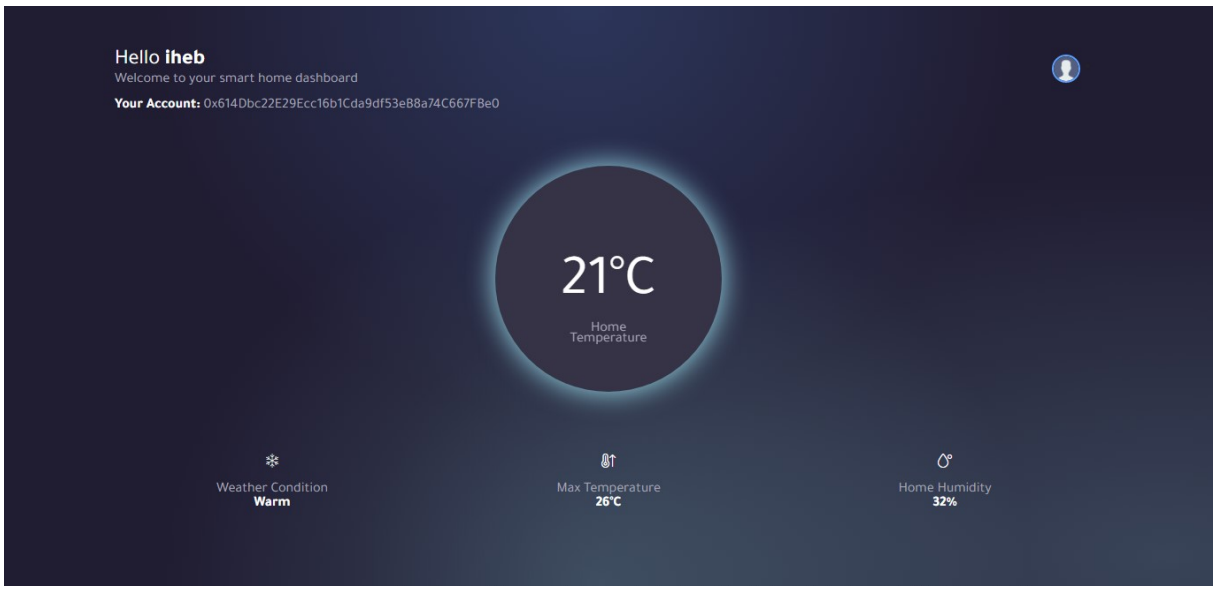


Figure 12: Page d'accueil

6. Évaluation

Dans cette partie, nous allons évaluer l'implémentation de notre système basé sur la Blockchain pour la confidentialité des politiques des services d'un Smart Home. Cette évaluation basée sur le gaz et le coût et la vitesse utilisés dans le déploiement des contrats et les appels des fonctions.

Les deux tableaux et les figures ci-dessous représentent l'évaluation en détails.

Déploiement	**Gaz utilisé**	**Coût total (ETH)**	**Vitesse**
Déploiement du contrat (SensorIoT.sol)	198499	0,000496247501389493	172ms
Déploiement du contrat (Auth.sol)	637373	0.001593432504461611	200ms

Tableau 4: Évaluation de déploiement des Smart Contracts

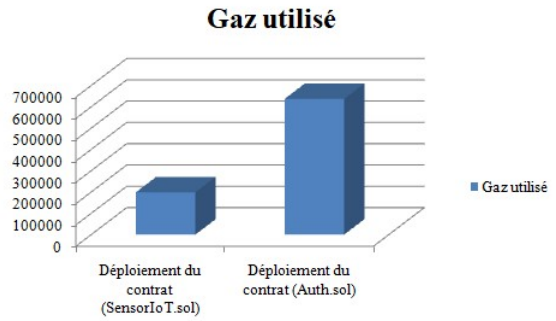


Figure 13: Le gaz utilisé dans le déploiement des Smart Contracts

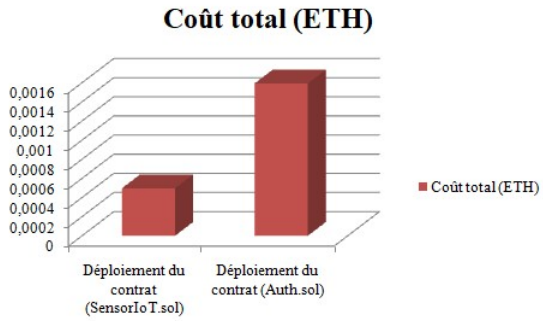


Figure 14: Le coût total (ETH) utilisé dans le déploiement des Smart Contracts

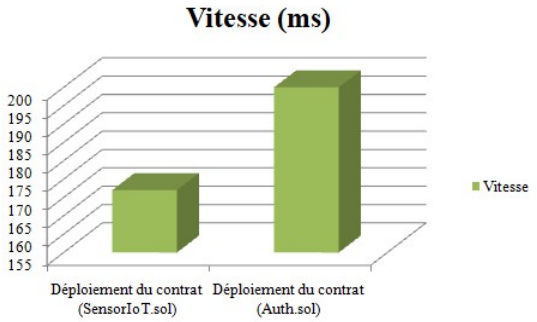


Figure 15: La vitesse en milliseconde pour le déploiement des Smart Contracts

Appel des fonctions	Gaz utilisé	Coût total (ETH)	Vitesse
SetDATA	29151	0,00023614	70ms
CreateUser	120501	0,00045228	122ms

Tableau 5: Évaluation d'appel des fonctions

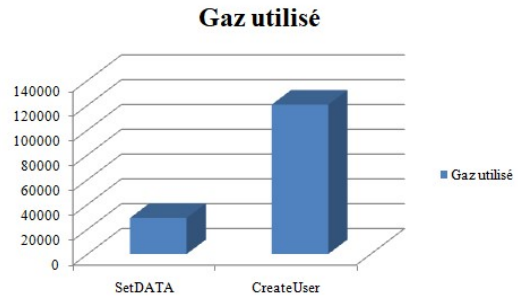


Figure 16: Le gaz utilisé dans l'appel des fonctions

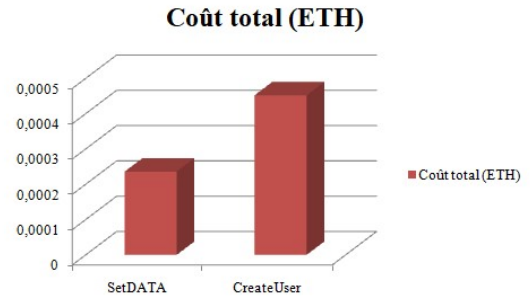


Figure 17: Le coût total (ETH) utilisé dans l'appel des fonctions

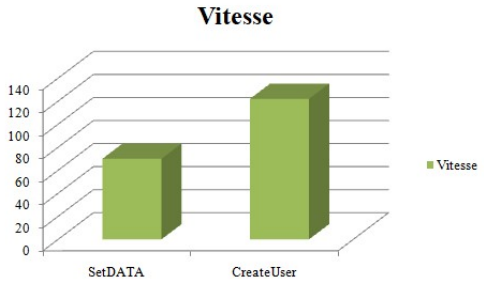


Figure 18: La vitesse en milliseconde pour l'appel des fonctions

7. Conclusion

En conclusion, nous avons exposé l'implémentation de notre scénario visant à sécuriser les politiques des services d'un Smart Home, ainsi que les outils et langages nécessaires au développement de notre système. Nous avons évalué notre système a la basée de gaz et de coût et de vitesse utilisés dans le déploiement des contrats et les appels des fonctions.

Conclusion Générale

Conclusion

L’objet de travail visait principalement à présenter un modèle décentralisé basé sur la technologie de la Blockchain dans le but est de renforcer la sécurité dans les environnements de l'Internet des objets (IoT). L'accent a été mis sur le développement d'une architecture sécurisée pour les Smart Homes afin de protéger les politiques des services. Notre objectif était de proposer une solution innovante et prometteuse pour relever les défis de sécurité propres aux environnements IoT

Dans le premier chapitre, nous avons exposé les fondements des environnements IoT et de la technologie de la Blockchain, en mettant en évidence les défis de sécurité auxquels ils sont confrontés. Nous avons également souligné les avantages potentiels de la décentralisation offerte par la Blockchain pour résoudre ces défis.

Le deuxième chapitre a examiné l'état de l'art des mécanismes de sécurité décentralisés pour l'IoT, en identifiant les approches existantes, leurs avantages et leurs limites. Cette revue de littérature nous a permis de comprendre les aspects actuels de la sécurité dans l'IoT et d'identifier les opportunités pour développer une solution plus robuste.

Dans le troisième chapitre, nous avons présenté une architecture de sécurité basée sur la Blockchain spécifiquement conçue pour les politiques des services dans un Smart Home. Cette architecture a exploité les caractéristiques clés de la Blockchain, telles que la décentralisation, l'immuabilité et la transparence, pour garantir l'intégrité, la confidentialité et l'authenticité des données et des transactions dans un environnement IoT.

Ensuite, nous avons décrit l'implémentation et l'évaluation de l'architecture proposée. Des tests et des expérimentations ont été réalisés pour évaluer les performances, la sécurité et l'efficacité de l'architecture dans des scénarios réels. Les résultats obtenus ont confirmé la faisabilité et l'efficacité de la solution proposée, démontrant son potentiel pour sécuriser les environnements IoT.

En conclusion, ce mémoire de recherche a apporté une contribution significative à l'exploration de solutions décentralisées basées sur la Blockchain pour sécuriser les environnements IoT. L'architecture de sécurité proposée a démontré son efficacité et son potentiel à résoudre les défis de sécurité spécifiques aux environnements IoT, tels que la confidentialité des données, l'intégrité des transactions et la résistance aux attaques malveillantes.

Perspective

L’objet de ce travail jouera un rôle important dans le domaine de la sécurité des environnements IoT en offrant une solution décentralisée et robuste. Le mémoire établit les bases d'un modèle décentralisé basé sur la Blockchain pour sécuriser les environnements IoT. Il ouvre de nouvelles perspectives de recherche et de développement visant à renforcer la sécurité des systèmes IoT et contribue à un avenir plus sécurisé et fiable pour l'IoT.

Les perspectives d'avenir dans le domaine de la sécurité des environnements IoT sont cruciales pour assurer la fiabilité et la confidentialité des données échangées. Cette recherche propose un modèle décentralisé basé sur la Blockchain qui offre une solution robuste pour renforcer la sécurité des systèmes IoT. Il est prévu que cette approche contribuera à un avenir plus sécurisé et fiable pour l'IoT en garantissant l'intégrité des données, la confidentialité et l'authentification dans les environnements IoT.

Il convient de noter que les résultats de recherche fournis ne sont pas spécifiques à la proposition de recherche mentionnée dans la question, mais ils offrent un contexte général sur les prévisions et les rôles importants joués par différentes initiatives et propositions.

Références

[1] R. Ramaswamy, Siddharth Tripathi Somayya Madakam, "Internet of Things (IoT): A Literature Review," *Journal of Computer and Communications*, pp. 164-173, 2015.

[2] World Economic Forum, "State of the Connected World," World Economic Forum In collaboration with the Global Internet of Things Council and PwC, 2020.

[3] Sarmad Ullah Khan, Rifaqat Zaheer and Shahid Khan Rafiullah Khan, "Future Internet: The Internet of Things Architecture Possible Applications and Key Challenges," *Proceedings of Frontiers of Information Technology (FIT)*, pp. 257-260, 2012.

[4] Venkata P. Yanambaka and Ahmed Abdelgawad Pintu Kumar Sadhu, "Internet of Things: Security and Solutions Survey," *Sensors*, 2022.

[5] Wang Chen, "AN IBE BASED SECURITY SCHEME OF INTERNET OF THINGS," *Cloud Computing and Intelligent Systems (CCIS)*, pp. 1046-1049, 2012.

[6] Jiafu Wan, Caifeng Zou and Jianqi Liu Hui Suo, "Security in the Internet of Things: A Review," *in Computer Science and Electronics Engineering (ICCSEE)*, pp. 648-651, 2012.

[7] Ting-lie Lu, Fei-Yang Ling, ling Sun and Hui-Ying Du MiaoWu, "Research on the architecture of Internet of things," *in Advanced Computer Theory and Engineering (ICACTE),* pp. 484-487, 2010.

[8] Minghui Zhang and Fuquan Sun Xu Cheng, "Architecture of internet of things and its key technology integration based-on RFID," *Fifth International Symposium on Computational Intelligence and Design*, pp. 294-297, 2012.

[9] Youssef Baddi, Abderrahim Hasbi Hamza Zemrane, "Comparison between IOT protocols: ZigBee and WiFi using the OPNET simulator," *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications*, pp. 1-6, October 2018.

[10] Chunxia Chen, Qijie Jiang Yongkang Wang, "Security algorithm of Internet of Things based on ZigBee protocol," *Cluster Computing*, vol. 22, pp. 14759-14766, 2019.

[11] Ramon Villarino, David Girbau Antonio Lazaro, "A Survey of NFC Sensors Based on Energy Harvesting for IoT Applications," *Sensors*, 2018.

[12] Lin Zhu Handong Zhang, "Internet of Things: Key technology, Architecture and Challenging Problems ," *Computer Scienceand Automation Engineering (CSAE)*, vol. 4, pp. 507-512, June 2011.

[13] Muhammad Waseem, Sadia Mazhar, Anjum Khairi and Talha Kamal M.U. Farooq, "A Review on Internet of Things (IoT)," *International Journal of Computer Applications*, vol. 113, no. 1, 2015.

[14] Jung-Chih Chiao V.M. Lubecke, "MEMS technologies for enabling high frequency communications circuits," *Telecommunications in Modern Satellite, Cable and Broadcasting Services*, vol. 2, pp. 382-389, 1999.

[15] Cristian Martín, Jaime Chen, Enrique Soler, Manuel Díaz Ana Reyna, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173-190, November 2018.

[16] Bogdan IANCU, Catalin BOJA, Tiberiu-Marian GEORGESCU, Cosmin CARTAS, Marius POPA, Cristian Valeriu TOMA Alin ZAMFIROIU, "IoT Communication Security Issues for Companies: Challenges, Protocols and The Web of Data," *International Conference on Business Excellence*, pp. 1109-1120, 2020.

[17] Derar Eleyan Rozan Khader, "Survey of DoS/DDoS attacks in IoT," *Sustainable Engineering and Innovation*, vol. 1, pp. 23-28, January 2021.

[18] Chrysostomos Chrysostomou, George Hadjichristofi Ioannis Andrea, "Internet of Things: Security Vulnerabilities and Challenges ," *2015 IEEE Symposium on Computers and Communication (ISCC)*, pp. 180-187, 2015.

[19] Amarsinh Vidhate Jyoti Deogirikar, "Security Attacks inIoT: A Survey," *International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pp. 32-37, February 2017.

[20] V Anantha Narayanan Sode Pallavi, "An Overview of Practical Attacks on BLE Based IOT Devices and Their Security," *5th international conference on advanced computing & communication systems (ICACCS) IEEE*, pp. 694-698, March 2019.

[21] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[22] Dr. Henry Selvaraj Suman Ghimire, "A Survey on Bitcoin Cryptocurrency and its Mining," *26th International Conference on Systems Engineering (ICSEng)*, pp. 1-6, December 2018.

[23] Ripple. Ripple. [Online]. https://ripple.com/

[24] Hyperledger. Hyperledger. [Online]. https://www.hyperledger.org/

[25] A. Kiayias, et N. Leonardos J. Garay, "The Bitcoin Backbone Protocol: Analysis and Applications," *Springer Berlin Heidelberg*, pp. 281-310, 2015.

[26] G. O. Karame, V. Capkun, et S. Capkun A. Gervais, "Is bitcoin a decentralized currency?," *IEEE Security Privacy*, vol. 12, pp. 54-60, 2014.

[27] Daniel Ichbiah. (2021, Octobre) Futura Sciences. [Online]. www.futura-sciences.com

[28] Pablo Lamela Seijas, Alexander Nemish, Simon Thompson Dmytro Kondratiuk, "Standardized crypto-loans on the Cardano blockchain," *Financial Cryptography and Data Security Springer*, pp. 579-594, March 2021.

[29] Alfonso Cevallos, Peter Czaban, Rob Habermeier, Syed Hosseini, Fabio Lama, Handan Alper, Ximin Luo, Fatemeh Shirazi, Alistair Stewart, Gavin Wood Jeff Burdges, "Overview of Polkadot and its Design Considerations," *arXiv preprint*, June 2020.

[30] Cosmos. cosmos Network. [Online]. https://cosmos.network/

[31] ICON. ICON. [Online]. https://icon.community/

[32] Bharat Bhushan, et Mohd Abdul Ahad Shivam Saxena, "Blockchain based solutions to secure IoT: Background, integration trends and a way forward," *Journal of Network and Computer Applications*, 2021.

[33] Kathleen E. Wegrzyn Eugenia Wang. (2021) Foley & Lardner LLP. [Online]. https://www.foley.com/en/insights/publications/2021/08/types-of-blockchain-public-private-between

[34] Iuon-Chang Lin and Tzu-Chun Liao, "A Survey of Blockchain Security Issues and Challenges," *International Journal of Network Security*, vol. 19, no. 5, pp. 653-659, 2017.

[35] Vanessa R. L. Chicarino, Célio V. N. de Albuquerque, and Antônio A. de A. Rocha Emanuel Ferreira Jesus, "A Survey of How to Use Blockchain to Secure Internet of Things and the Stalker Attack," *Security and Communication Networks*, 2018.

[36] Fahad Algarni and Mohammad Tabrez Quasim Mohammad Ayoub Khan, "Decentralised Internet of Things," *Springer*, vol. 71, pp. 3-20, 2020.

[37] James Peter Thomas Lovejoy, "An Empirical Analysis of Chain Reorganizations and Double-Spend Attacks on Proof-of-Work Cryptocurrencies," *S.B. Electrical Engineering and Computer Science*, MAY 2020.

[38] Ramani S, Marimuthu Karuppiah Nisanth Reddy Kasi, "Blockchain architecture, taxonomy, challenges, and applications," *Hybrid Computational Intelligence for Pattern Analysis*, pp. 1-31, 2022.

[39] Heung-No Lee Jehyuk Jang, "Profitable Double-Spending Attacks," *Applied Sciences*, vol. 10, 2020.

[40] Andreas Bogner, Dominik Bilgeri, Elgar Fleisch, Felix Wortmann Mathieu Chanson, "Blockchain for the IoT: Privacy-Preserving Protection of Sensor Data," *Journal of the Association for Information Systems*, vol. 20, no. 9, pp. 1274-1309, September 2019.

[41] Pradeep Kumar Mallick Nallapaneni Manoj Kumara, "Blockchain technology for security issues and challenges in IoT," *Procedia Computer Science*, vol. 132, pp. 1815-1823, 2018.

[42] Shoji Kasahara, Yulong Shen, Xiaohong Jiang, Jianxiong Wan Yuanyu Zhang, "Smart Contract-Based Access Control for the Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594-1605, 2018.

[43] Uurtsaikh Jamsrandor, Ralph Deters Mayra Samaniego, "Blockchain as a Service for IoT," *IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData)*, pp. 433-436, 2016.

[44] Mehdi Aminian, Bahman Javadi Nazanin Zahed Benisi, "Blockchain-based decentralized storage networks: A survey," *Journal of Network and Computer Applications*, vol. 162, p. 102656, 2020.

[45] Heliasadat Hosseinian, Leila Damghani Hamidreza Damghani, "Cryptography review in IoT," *Conference on Technology In Electrical and Computer Engineering*, 2019.

[46] Ondrej Gallo, Roderik Ploszek, Peter Špaček, Pavol Zajac Stefan Balogh, "IoT Security Challenges: Cloud and Blockchain, Postquantum Cryptography, and Evolutionary Techniques," *Electronics*, vol. 10, no. 21, p. 2647, October 2021.

[47] Michael Milliken, Pushpinder Kaur Chouhan, Bryan Scotney, Zhiwei Lin, , Ali Sajjad, Mark Shackleton Jorge Martınez Carracedo, "Cryptography for Security in IoT," *Fifth International Conference on Internet of Things: Systems, Management and Security*, pp. 23-30, October 2018.

[48] Maaz Bin Ahmad, Muhammad Khalid Khan Mohammad Ayub Latif, "A Review on Key Management and Lightweight Cryptography for IoT," *Global Conference on Wireless and Optical Technologies (GCWOT)*, pp. 1-7, 2020.

[49] Amer Al-Rahayfeh, Muder Almiani, Salman Yussof, Omar Alfandi, Ahed Abugabah, Yaser Jararweh Saleh Atiewi, "Scalable and Secure Big Data IoT System Based on Multifactor Authentication and Lightweight Cryptography," *IEEE Access*, vol. 8, pp. 113498-113511, 2020.

[50] Paul Baran, "Introduction to distributed communications networks," *RAND CORP SANTA MONICA CALIFORNIA*, August 1964.

[51] Imran Bashir, *Mastering Blockchain A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more*, Third Edition ed., Packt, Ed.: Packt, 2020.

[52] Benedikt Notheisen, Carolin Beer, David Dauer Esther Mengelkamp, "A blockchain-based smart grid: towards sustainable local energy markets," *Computer Science-Research and Development*, vol. 33, pp. 207-214, 2018.

[53] T. M Stoenescu, *Decentralized resource allocation mechanisms in networks*., 2004.

[54] Buterin V, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, pp. 2-1.

[55] Sergei Tikhomirov, "Ethereum: State of Knowledge and Research Perspectives," *Springer International Publishing*, pp. 206-221, 2018.

[56] Chris Dannen, *Introducing Ethereum and Solidity*., 2017.

[57] David Irvine, "MaidSafe Distributed File System," p. 2010.

[58] Beddows Kordek M, "White paper: Lisk," *Technical report*, 2016.

[59] EOSIO. EOS. [Online]. https://eos.io/

[60] KPN IoT, *La sécurité de l'Internet des Objets (ou IoT) C'est parti pour un Internet des objets sûr*, 10th ed., kpn, Ed.

[61] Michael Devetsikiotis Konstantinos Christidis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292-2303, 2016.

[62] Ralph Deters Mayra Samaniego, "Blockchain as a Service for IoT ," *IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data*, pp. 433-436, 2016.

[63] Vijay K. Madisetti Arshdeep Bahga, "Blockchain Platform for Industrial Internet of Things," *Software Engineering and Applications*, pp. 533-546, 2016.

[64] Floriano Scioscia, Saverio Ieva, Giovanna Capurso, Eugenio Di Sciascio Michele Ruta, "Semantic Blockchain to Improve Scalability in the Internet of Things," *Open Journal of Internet of Things (OJIOT)*, vol. III, pp. 46-61, 2017.

[65] Melih Burak Mert, Okan Mete, Azer Ramazanli, Kaan Sarica and Bora Gungoren Pelin Angin, "A blockchain-based decentralized security architecture for IoT," *Springer International Publishing*, pp. 3-18, June 2018.

[66] Yingying Jiang, Jing Chen, Yu Zhang, Jeungeun Song, Ming Zhou et Matevž Pustišek Yongfeng Qian, "Towards decentralized IoT security enhancement: A blockchain approach," *Computers and Electrical Engineering*, pp. 266-273, 2018.

[67] Pratik Verma, Rahul Sonanis, Umang Goel, Dr. Aloknath De, Sai Anirudh Kondaveeti et Suman Shekhar Rahul Agrawal, "CONTINUOUS SECURITY IN IOT USING BLOCKCHAIN," *IEEE*, pp. 6423-6427, 2018.

[68] Cristian Martín, Jaime Chen, Enrique Soler, Manuel Díaz Ana Reyna, "On blockchain and its integration with IoT. Challenges and opportunities," *Future generation computer systems*, vol. 88, pp. 173-190, 2018.

[69] Haider Abbas, Faiza Iqbal, Abdelouahid Derhab Ayesha Altaf, "Trust models of internet of smart things: A survey, open issues, and future directions," *Journal of Network and Computer Applications*, vol. 137, pp. 93-111, 2019.

[70] Oscar Novo, "Scalable Access Management in IoT Using Blockchain: A Performance Evaluation," *IEEE INTERNET OF THINGS JOURNAL*, vol. VI, no. 3, pp. 4694-4701, June 2019.

[71] Microsoft Visual Studio Code. Visual Studio Code. [Online]. https://code.visualstudio.com/

[72] Node JS. Node.js. [Online]. https://nodejs.org/

[73] Truffle Suite. Truffle. [Online]. https://trufflesuite.com/docs/truffle/

[74] Truffle Suite. Ganache. [Online]. https://trufflesuite.com/docs/ganache/

[75] MetaMask. The crypto wallet for Defi, Web3 Dapps and NFTs | MetaMask. [Online]. https://metamask.io/

[76] Solidity. Solidity Programming Language. [Online]. https://soliditylang.org/

[77] JavaScript. JavaScript Mozilla Developer Network. [Online]. https://developer.mozilla.org/fr/docs/Web/JavaScript

[78] Facebook React. React - A JavaScript library for building user interfaces. [Online]. https://legacy.reactjs.org/

[79] Arduino. Arduino CC. [Online]. https://www.arduino.cc/en/software

[80] Bootstrap. Bootstrap The most popular HTML, CSS, and JS library. [Online]. https://getbootstrap.com/

[81] Font Awesome. Font Awesome. [Online]. https://fontawesome.com/