REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABOU-BEKR BELKAID – TLEMCEN

THÈSE LMD

Présentée à :

FACULTE DES SCIENCES – DEPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

DOCTORAT

Spécialité : *Informatique distribuée* et *réseaux (IDR)*

Par :

Mme ZERGA Hideyat

Sur le thème

Optimisation des systèmes de e-santé à base de l'internet des objets

---

Soutenue publiquement le 20/05/2023 à Tlemcen devant le jury composé de :

Pr BENAMAR Abdelkrim	Professeur	Université de Tlemcen	Président
Pr BENMAMMAR Badr	Professeur	Université de Tlemcen	Directeur de thèse
Dr AMRAOUI Asma	Maître de Conférences A	Université de Tlemcen	Co-Directrice de thèse
Dr DEBBAL Mohammed	Maître de Conférences A	Université de Ain Temouchent	Examinateur
Dr MERAD BOUDIA Omar	Maître de Conférences A	Université d'Oran 1	Examinateur
Pr LABRAOUI Nabila	Professeur	Université de Tlemcen	Examinatrice

Laboratoire de Télécommunication de Tlemcen (LTT)
BP 119, 13000 Tlemcen - Algérie

A la mémoire de ma fille

Ibtissem Addou

Je dédie cette thèse à

Mes chers parents,

Mon mari Adel,

Ma fille Ines Sofia,

Mes soeurs Norhène et Kamila,

Mon frère Abdrezzak,

Mes beaux-parents,

Mes belles-soeurs

Et à toute ma famille et mes amis

Remerciements

Je souhaite remercier en premier lieu mon directeur de thèse Professeur BENMAMMAR Badr pour m'avoir accueilli au sein de son équipe. Je lui suis également reconnaissante pour le sérieux, la capacité d'analyse dont il a fait preuve durant mon encadrement en master et en doctorat. Il a toujours été disponible et à l'écoute. Enfin, ses nombreuses relectures et corrections de cette thèse ont été très enrichissantes. Cette thèse lui doit beaucoup. Pour tout cela merci.

Je remercie tout particulièrement ma co-directrice de thèse Docteur AMRAOUI Asma qui m'a dirigé tout au long de cette thèse. Elle a toujours été disponible, à l'écoute de mes nombreuses questions. Les nombreuses discussions que nous avons eues ainsi que ses conseils sont pour beaucoup dans le résultat final de ce travail. Sa capacité d'analyse et son enthousiasme m'ont montré que le monde de la recherche pouvait être un univers passionnant. Sa rigueur, ses qualités pédagogiques et scientifiques m'ont permis de progresser et ont répondu à plusieurs de mes préoccupations.

Je suis reconnaissante aux membres du Laboratoire LTT pour leur soutien. Ma gratitude va également aux membres du comité, professeur BENAMAR Abdelkrim en tant que président du comité, Docteur DEBBAL Mohammed de l'université de Ain Temouchent, Docteur MERAD BOUDIA Omar Rafik de l'université d'Oran 1 et Professeur LABRAOUI Nabila comme examinateurs, qui ont accepté d'examiner et d'évaluer mon travail. Je remercie aussi le Docteur MAATALLAH Houcine le chef du département informatique pour sa disponibilité, sa compréhension et son encouragement. Je remercie les relecteurs anonymes pour leur précieux commentaires, pour leur objectivité et pour leurs efforts.

A titre plus personnel, Je remercie chaleureusement mes parents sans qui je ne serai jamais là où je suis aujourd'hui. Leur soutien, leurs encouragements et leurs douaas ont fait de moi qui je suis. Je remercie aussi mon mari Adel pour la grande patience, l'encouragement et la confiance qu'il m'a témoignée. Je tiens à le remercier surtout pour son soutien moral ininterrompu et ses nombreux conseils tout le long de ma thèse. En fin je remercie mes beaux-parents pour leurs soutiens qui m'a été bien utile durant ma thèse.

Je suis reconnaissant à toutes personne qui m'a aidé de près ou de loin durant ma thèse.

Résumé

Avec sa capacité à "minimiser l'intervention humaine lors de la génération, de l'échange et de la consommation de données", l'Internet des objets (IoT) se déploie de plus en plus dans tous les secteurs, en particulier dans le secteur de la santé. L'IoT dans les soins de santé permet de garder les patients connectés avec des appareils portables et d'autres outils de surveillance des patients à distance afin d'aider les praticiens à travailler plus efficacement. Cependant, cette innovation implique que les patients partagent à distance leurs données personnelles et physiologiques avec le personnel hospitalier, ce qui peut mettre en danger la vie privée du patient. Ainsi, la mise en place d'un contrôle d'accès est obligatoire. Par conséquent, l'objectif de cette thèse est de parvenir à un contrôle d'accès distribué et fiable pour les systèmes de soins de santé en utilisant la technologie de la Blockchain. Pour ce faire, nous avons proposé trois approches différentes de contrôle d'accès basées sur des contrats intelligents et une approche basée sur les jetons non fongibles. Nos propositions ont été comparées avec des travaux connexes en termes de latence de réponse à la demande d'accès et de consommation de gaz liée au déploiement du contrat, à l'exécution des fonctions et aux différentes réponses. Les résultats obtenus sont très satisfaisants.

Mots clés : IoT, soins de santé, contrôle d'accès, blockchain, contrat intelligent, jetons non fongibles.

Abstract

With its ability to "minimize human intervention when generating, exchanging and consuming data", the Internet of Things (IoT) is increasingly being deployed in all sectors, particularly in the health sector. IoT in healthcare keeps patients connected with wearable devices and other remote patient monitoring tools to help practitioners work more efficiently. However, this innovation involves patients sharing their personal and physiological data remotely with the hospital staff, which may endanger patient privacy. Thus, the implementation of an access control is mandatory. Therefore, the objective of this thesis is to achieve distributed and reliable access control for healthcare systems using Blockchain technology. To do so, we proposed three different approaches to access control based on smart contracts and one approach based on non-fungible tokens (NFT). Our proposals were compared with related works in terms of access request response latency and gas consumption related to contract deployment, function execution and different responses. The obtained results are very satisfactory.

Keywords: IoT, healthcare, access control, blockchain, smart contract, NFT.

ملخص

بفضل قدرتها على "تقليل التدخل البشري عند إنشاء البيانات وتبادلها واستهلاكها"، يتم استخدام إنترنت الأشياء بشكل متزايد في جميع القطاعات، لا سيما في قطاع الصحة. تحافظ إنترنت الأشياء في مجال الرعاية الصحية على اتصال المرضى بالأجهزة القابلة للارتداء وغيرها من أدوات مراقبة المريض عن بُعد لمساعدة ممارسين المهنة على العمل بكفاءة أكبر. ومع ذلك، فإن هذا الابتكار ينطوي على مشاركة المرضى عن بعد لبياناتهم الشخصية والفسيولوجية مع موظفي المستشفى، مما قد يعرض خصوصية المريض للخطر. وبالتالي، فإن تنفيذ التحكم في الوصول إلزامي. لذلك، فإن الهدف من هذه الأطروحة هو تحقيق التحكم في الوصول الموزع والموثوق لأنظمة الرعاية الصحية باستخدام تقنية سلسلة الكتل. للقيام بذلك، اقترحنا ثلاث طرق مختلفة للتحكم في الوصول استنادًا على العقود الذكية وطريقة واحدة تعتمد على الرموز الغير القابلة للاستبدال. تمت مقارنة مقترحاتنا بالأعمال ذات الصلة من حيث زمن استجابة طلب الوصول واستهلاك الغاز المرتبط بنشر العقود وتنفيذ الوظيفة والاستجابات المختلفة. النتائج التي تم الحصول عليها مرضية للغاية.

الكلمات الرئيسية: إنترنت الأشياء، الرعاية الصحية، التحكم في الوصول، سلسلة الكتل، العقد الذكي، الرموز غير القابلة للاستبدال.

Table des matières

Liste des figures

Liste des tableaux

Liste des algorithmes

Liste des acronymes

ACC: Access Control Contract

AcToken: Access Token

AES: Advanced Encryption Standard

BLE: Bluetooth Low Energy

BC : Blockchain

CABA : Contrôle d'Accès Basé sur les Attributs

CABCap : Contrôle d'Accès Basé sur la Capacité

CABR : Contrôle d'Accès Basé sur les Roles

CABU : Contrôle d'Accès Basé sur l'Utilisation

CC : Couche Cloud

CAD : Contrôle d'Accès Discrétionnaire

CAO : Contrôle d'Accès Obligatoire

CASR : Contrôle d'Accès basé sur la Surveillance des Références

CDM : Couche de Dispositifs Médicaux

CF: Couche Fog

Cisco: Computer Information system Company

DM : Dispositifs Médicaux

DOS : Deni Of Service

DPoS : Distributed PoS

DES : Dossier de Santé Electronique

ECG: ElectroCarDiogram

EEG: ElectroEncephaloGram

EPC: Electronic Product Code

ERC: Ethereum Request for Comments

ETSI: European Telecom Standards Institute

FC: Fog computing

GSM: Global System for Mobile Communication

HIPPA: Health Insurance Portability and Accountability Act

IA: Intelligence Artificielle

IaaS: Infrastucture as a Service

IEEE: Institute of Electrical and Electronics Engineers

IETF: Internet Engineering Task Force

IoT: Internet Of Things

JVM: Java Virtual Machine

LAN: Local Area Network

LTE: Long Term Evolution

MCA : Matrice de Contrôle d'Accès

MCC: Mobile Cloud Computing

MITM: Man In The Middle

MJC: Misconduct Judge Contract

NAF : Nombre d'Accès Fréquents

NF: Noeud Fog

NFC: Near Field Communication

NFT : Non-Fungible Token

NIST: National Institute of Standards and Technology

ORIS: Organisation Régional d'Information sur la Santé

PaaS: Platform as a Service

PAN: Personal Area Network

PoA : Proof of Activity

PO: Propriétaire de l'Objet

PoB : Proof of Burn

PoD : Proof of Deposit

PoI : Proof of Importance

PoS : Proof of Stake

PoW : Proof of Work

RC: Register Contract

RFID: Radio Frequency IDentification

RGPD : Règlement Général sur la Protection des Données

RSA: Rivest, Shamir, and Adleman

SaaS: Software as a Service

SAP: Systems, Applications and Products for data processing

SC : Smart contract

SSL: Secure Socket Layer

TDC : Tiers De Confiance

TLS: Transport Layer Security

Ts : Right transfert Token

UMTS: Universal Mobile Telecom System

UWB: Ultra-Wide Band

VDC: Virtual Data Center

VoIP: Voix sur Internet Protocol

WiFi: Wireless Fidelity

WSN: Wireless Sensor Network

W3C: World Wide Web Consortium

XACML: eXtensible Access Control Markup Language

3G: Third Generation

Introduction générale

Avec sa capacité à "minimiser l'intervention humaine lors de la génération, de l'échange et de la consommation de données", l'Internet des objets ou Internet of Things (IoT) se déploie de plus en plus dans tous les secteurs, en particulier dans le secteur de la santé. L'intégration de l'IoT dans les systèmes de soins de santé a permis d'une part de garder les patients connectés avec des appareils portables et d'autres outils de surveillance à distance afin d'aider les praticiens à travailler plus efficacement, et de l'autre part, le suivi des besoins médicaux, comme les rappels de rendez-vous, la surveillance du nombre de calories, la surveillance de la pression artérielle, et bien d'autres fonctionnalités. Les systèmes de soins de santé basés sur l'IoT peuvent être définis comme la « prestation à distance de services de santé, y compris le suivi et la consultation au moyen d'outils de télécommunication, tels que les smartphones, les capteurs et actionneurs médicaux » [1].

Les signes vitaux des patients dans les systèmes de soins de santé basés sur l'IoT sont mesurés par des dispositifs médicaux puis sont regroupés dans des dossiers de santé électroniques connus sous le nom de DSE propres aux patients aux coté de leurs informations personnelles. Ces DSE sont stockés par la suite dans le cloud afin que les professionnels de la santé puissent y accéder. Ces dossiers contiennent des informations sensibles faisant partie de la vie privée du patient qui doivent être protégée [2].

Le Fog computing (FC) est une plateforme virtualisée placée entre les appareils des utilisateurs et les centres de données Cloud. Certains traitements de données peuvent être effectués plus près de la source de génération de données, ce qui permet de répartir les demandes de ressources, de réduire la latence et la consommation d'énergie [3, 4]. Ainsi, le FC permet d'optimiser les systèmes de soins de santé en réduisant la latence d'accès des utilisateurs à leurs dossiers.

La Blockchain (BC) est un registre sécurisé de blocs qui est utilisé pour stocker et partager des données de manière distribuée [5]. Étant donné que tous les nœuds ont la même copie de blocs, le besoin d'une autorité centrale est éliminé d'une part, et une atmosphère de transparence et d'ouverture est créée, permettant aux acteurs de santé de savoir comment leurs données sont utilisées d'une autre part. Plus important encore, compromettre un nœud dans le réseau ne signifie pas affecter l'état du grand registre. De plus, la propriété d'immuabilité de la BC qui rend impossible d'altérer ou de modifier tout enregistrement qui a été ajouté au registre correspond très bien aux exigences du stockage des DSE. Ainsi, la BC permet d'optimiser les systèmes de soins de santé en garantissant aux utilisateurs un accès sécurisé à leurs dossiers [6].

Il est essentiel de contrôler l'accès aux DSE afin d'éviter l'accès illégitime aux données des patients par des entités non approuvées. Dans le contrôle d'accès centralisé traditionnel, toutes les demandes d'accès des utilisateurs sont traitées par un serveur centralisé. Si ce serveur tombe en panne, tout le système tombera en panne, donc le mécanisme de contrôle d'accès ne sera plus efficace [7, 8]. Ce serveur centralisé est appelé un point de défaillance unique en matière de sécurité. Par conséquent, le système de contrôle d'accès doit être distribué pour résoudre le problème de point de défaillance unique, évolutif pour héberger un grand nombre d'utilisateurs et situé plus près de l'utilisateur final pour fournir une réponse en temps réel [9]. Afin de répondre à toutes ces exigences, nous allons dans cette thèse, combiner les mécanismes de contrôle d'accès avec la technologie blockchain pour assurer un contrôle distribué et évolutif, en utilisant un réseau à base de fog computing pour assurer une réponse en temps-réel. Ainsi, nous optimiserons le contrôle d'accès et la latence de la réponse d'accès dans les systèmes de soins de santé.

Les contrats intelligents de la BC sont des fonctions qui peuvent être écrites dans la BC puis exécutées par tous les nœuds de ce registre, ils peuvent être définis comme un ensemble de conditions prédéfinies et convenues entre les différentes parties impliquées [10]. Lorsque ces conditions sont remplies, une ou plusieurs actions sont déclenchées. Ainsi, la technologie de contrat intelligent est proposée comme solution pour mettre en œuvre un système de contrôle d'accès en premier lieu. Les jetons non fongibles connus sous le nom de NFT sont des actifs cryptographiques numériques dans la blockchain avec des codes d'identification uniques et des métadonnées qui les distinguent les uns des autres. Chaque NFT ne peut être utilisé que pour enregistrer des informations uniques relatives à un seul actif numérique. Ainsi l'utilisation de ces actifs est proposée comme solution pour mettre en œuvre un système de contrôle d'accès en second lieu.

Dans cette thèse, nous proposons trois mécanismes de contrôle d'accès basés sur les contrats intelligents et un mécanisme basé sur les NFT. La première proposition de contrôle d'accès basé sur les contrats intelligents consiste en un seul type de contrats intelligents, chaque propriétaire d'objet déploie un seul contrat intelligent qui va gérer les demandes d'accès de tous les sujets du système. La deuxième proposition consiste aussi en un seul type de contrats intelligents, sauf que le propriétaire d'objet déploie un contrat intelligent pour chaque sujet du système, chaque contrat va gérer les demandes d'accès d'un seul sujet. La troisième proposition consiste en trois types de contrats, le propriétaire d'objet déploie un contrat intelligent pour gérer tous les sujets du système, et l'administrateur du système déploie deux autres contrats. Dans le contrôle d'accès basé sur les NFT, le propriétaire de l'objet déploie un seul contrat intelligent qui va gérer les demandes d'accès de tous les sujets du système comme dans la première proposition du contrôle d'accès basé sur les

contrats intelligents, la seule différence est que le droit d'accès est encapsulé dans un jeton unique propre au sujet au lieu d'être encapsulé dans une transaction.

Pour donner au patient le contrôle total de ses données, c'est lui qui déploiera son contrat intelligent et définira ses politiques d'accès. Les nœuds et serveurs Fog seront intégrés à la Blockchain mais pas les appareils IoT (en raison de leur nature contrainte). Les nœuds Fog seront responsables de la gestion des autorisations de contrôle d'accès pour les appareils IoT tandis que les serveurs Fog agiront en tant que mineurs.

Nos contributions dans le cadre de cette thèse sont les suivantes :

1. Pour réduire la latence et la consommation d'énergie dans les systèmes de soins de santé basés sur l'IoT, nous proposons d'utiliser une architecture basée sur le Fog computing.
2. Pour permettre le partage des données entre domaines et garantir aux utilisateurs un accès sécurisé à leurs dossiers, nous avons intégré la technologie Blockchain.
3. Pour garantir un accès sécurisé aux données, nous proposons trois mécanismes de contrôle d'accès basés sur les contrats intelligents et un mécanisme basé sur NFT.
4. Pour donner au patient le contrôle total de ses données, c'est lui qui déploiera son contrat intelligent et définira ses politiques d'accès.
5. Après l'implémentation de nos schémas sur une BC Ethereum privée, la consommation de gaz liée aux déploiements de contrats et à l'exécution des fonctions, la surcharge de communication entre contrat et la latence de la réponse à la demande d'accès obtenues sont satisfaisant par rapport aux travaux connexes, ce qui prouve l'efficacité des quatre contrôles d'accès proposés dans les systèmes de soins de santé basé sur l'IoT.

Le manuscrit de cette thèse est organisé comme suit :

Dans le chapitre 1, nous expliquons en détail l'IoT ainsi que les systèmes de soins de santé, ensuite nous détaillons les avantages de l'intégration de l'IoT dans les systèmes de soins de santé. Dans le chapitre 2, nous définissons la blockchain ainsi que tous ses composants, puis nous expliquons comment l'usage de cette technologie peut optimiser les systèmes de soins de santé. Dans le chapitre 3, nous présentons l'architecture générale utilisée ainsi que les quatre propositions de contrôle d'accès et dans le chapitre 4, nous exposons l'implémentation des algorithmes proposés, puis nous discutons les résultats obtenus.

Chapitre I : Systèmes de santé basés sur l'IoT

I.1 Introduction

De nos jours, environ 5 milliards d'internautes dans le monde utilisent Internet pour naviguer sur le Web [11], envoyer et recevoir des courriels, accéder à du contenu multimédia et services, jouer à des jeux, utiliser des applications de réseautage social et de nombreuses autres tâches. L'internet constitue une structure continue de réseaux classiques et d'objets en réseau qui permettent aux machines et aux objets intelligents de communiquer et de dialoguer, rendant ainsi le contenu et les services autour de nous toujours disponibles, et ouvrant la voie à de nouvelles applications, et à de nouveaux modes de vie.

Dans cette perspective, le concept conventionnel d'internet en tant que réseau d'infrastructure à la recherche des terminaux des utilisateurs finaux s'efface, laissant place à une notion d'objets "intelligents" interconnectés formant des environnements informatiques envahissants appelé "internet des objets". Cette innovation a été rendue possible par l'intégration de l'électronique en objets physiques quotidiens, les rendant "intelligents" pour leur permettre de s'intégrer de manière transparente dans l'infrastructure cyber-physique globale résultante. L'infrastructure internet ne va pas disparaître. Au contraire, elle conservera son rôle vital en tant que colonne vertébrale mondiale pour le partage d'informations à l'échelle mondiale, et l'interconnexion d'objets physiques.

L'intégration de l'IoT dans les systèmes de soins de santé a considérablement amélioré la qualité et la prestation des soins et a réduit les coûts. En plus de donner accès à un nombre croissant de paramètres biométriques, il a donné au patient la possibilité d'autogérer ses informations et a donné aux professionnels de la santé un accès plus rapide et sécurisé à toutes les informations dont ils ont besoin pour prendre soin de leurs patients à distance.

Le chapitre I est organisé comme suit : dans la section 2, nous donnons la définition de l'IoT, puis nous présentons en détail l'architecture, les éléments et les domaines d'application de cette dernière ainsi que les normes qui facilitent sa mise en place. Nous expliquons ensuite ce qu'est le cloud et les paradigmes de bord et quel est leur relation avec l'IoT. Dans la section 3, nous définissons les systèmes de soins de santé et nous parlons de la génération 4.0. Dans la section 4, nous citons les avantages de l'intégration de l'IoT dans les systèmes de soins de santé, de même que les scénarios de déploiement qui peuvent en résulter, nous présentons une architecture de soins de santé, ensuite nous expliquons comment l'introduction du fog computing a pu améliorer ces systèmes, puis nous présentons les menaces de sécurité auxquelles ils font face, ainsi que les technologies utilisées pour les éviter. Nous concluons ce premier chapitre dans la section 5.

I.2 Internet des objets (IoT)

I.2.1 Définition de l'IoT

L'IoT permet aux objets physiques de voir, d'entendre, de penser et d'exécuter des tâches en les faisant « parler » ensemble, pour partager des informations, et pour coordonner les décisions. L'IoT transforme ces objets de traditionnels à intelligents en exploitant des technologies sous-jacentes telles que l'informatique omniprésente, les dispositifs intégrés, les technologies de communication, les réseaux de capteurs, et les protocoles et applications d'internet. Les objets intelligents avec leurs tâches supposées constituent des applications spécifiques à un domaine, tandis que les services de l'informatique et l'analyse omniprésente forment des services indépendants du domaine d'application [12].

I.2.2 Architecture de l'IoT

L'IoT est capable d'interconnecter des milliards d'objets hétérogènes via internet. Il est donc indispensable de disposer d'une architecture en couches flexible. Comme illustré dans la Figure I.1, il existe différentes architectures proposées dans la littérature, parmi elles, on retrouve : l'architecture (a) à trois couches, (b) basée sur le middleware. (c) orientée service. (d) à cinq couches. Le nombre toujours croissant d'architectures proposées n'a pas encore convergé vers un modèle de référence. Dans ce qui suit, nous allons expliquer le modèle (d) à cinq couches [13,14].

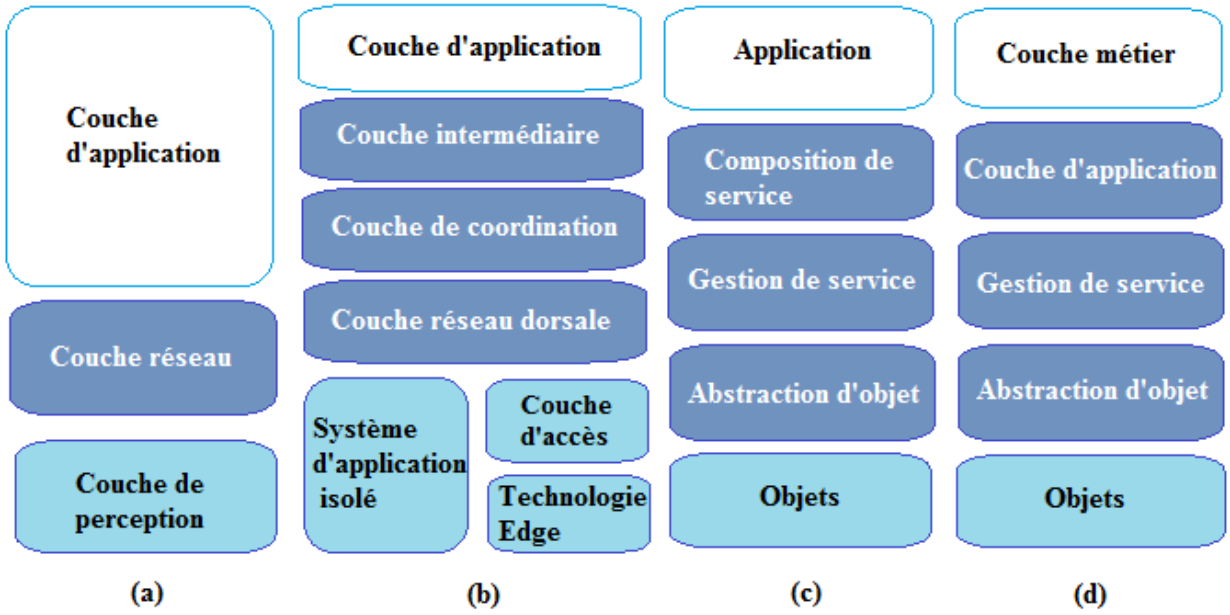

Figure I. 1. Architecture IoT : (a) à trois couches. (b) basée sur le middleware. (c) orientée service. (d) à cinq couches [13].

- **Couche d'objets :** la couche Objets appelée aussi couche perception de l'architecture à cinq couches, est responsable de la collecte de l'information grâce à des capteurs et des actionneurs qui peuvent effectuer différentes fonctionnalités telles que l'interrogation de l'emplacement, la température, le poids, le mouvement, l'humidité, etc. Les données volumineuses créées par l'IoT sont initiées à cette couche qui va les numériser pour ensuite les transférer à la couche d'abstraction d'objets.
- **Couche d'abstraction d'objet** : la couche d'abstraction d'objets est responsable du transfert des données produites par la première couche vers la troisième couche via des canaux sécurisés en utilisant diverses technologies telles que RFID (Radio Frequency Identification), 3G (third Generation), GSM (Global System for Mobile communication), UMTS (Universal Mobile Telecommunications System), WiFi (Wireless Fidelity), Bluetooth, infrarouge, ZigBee, ainsi que des processus de gestion des données.
- **Couche de gestion de service :** la couche gestion des services ou Middleware est responsable du traitement des données reçues, et de la prise de décision. En associant un service avec son demandeur en fonction des adresses et des noms, elle permet aux programmeurs d'applications IoT de travailler avec des objets hétérogènes sans tenir compte d'une plate-forme matérielle spécifique.
- **Couche d'application :** la couche d'application est responsable de la fourniture des services demandés par les clients, tel que la mesure de température ou celle de lézé'humidité de l'air. Cette couche est très importante pour l'IoT, car elle est capable de fournir des services intelligents de haute qualité pour répondre aux besoins des clients. Cette couche est hébergée sur des dispositifs puissants en raison de ses besoins informatiques complexes et énormes.
- **Couche métier :** en se basant sur les données reçues de la couche d'application, la couche métier est capable de construire un modèle d'entreprise, des graphiques, des organigrammes, etc. De plus, elle permet la conception, l'analyse, la mise en œuvre, l'évaluation, et la surveillance des éléments liés au système IoT. Elle permet aussi de prendre en charge les processus décisionnels basés sur l'analyse des Big data, et de comparer la sortie de chaque couche avec la sortie attendue pour améliorer les services et préserver la confidentialité des utilisateurs.

I.2.3 Eléments de l'IoT

Afin de mieux comprendre la signification réelle de l'IoT et sa fonctionnalité, nous allons définir les éléments requis pour l'IoT.

Figure I. 2. Eléments de l'IoT.

Comme illustrés dans la Figure I.2, il existe six éléments (composants) IoT qui permettent une informatique omniprésente transparente [13] :

- **Identification** : permet de correspondre les services avec leurs demandes. L'identifiant d'un objet désigne son nom, tel que "T1" pour un capteur de température particulier, à ne pas confondre avec l'adressage de l'objet qui se réfère à son adresse au sein du réseau de communication.
- **Détection** : signifie la collecte des données à partir de capteurs IoT pour les envoyer à un entrepôt de données. Les capteurs IoT peuvent être des capteurs intelligents, des actionneurs ou des dispositifs de détection portables.
- **Communication** : l'IoT utilise des protocoles de communication afin de connecter des objets hétérogènes pour fournir des services intelligents spécifiques, parmi ces protocoles, on peut citer : WiFi, Bluetooth, IEEE.802.15.4 (Institute of Electrical and Electronics Engineers) plus connu sous le nom de zigbee, Z-wave et LTE-Advanced (Long Term Evolution) qui sont utilisés pour les réseaux mobiles. Il existe d'autres protocoles de communications spécifiques qui sont utilisés par l'IoT tels que le RFID, le NFC (Near Field Communication) et le UWB (Ultra Wide Band).
- **Calcul** : l'unité de traitement (microcontrôleurs, microprocesseurs, circuits intégré) et les applications logicielles représentent le « cerveau » et la capacité de calcul de l'IoT. Il existe diverses plates-formes matérielles pour exécuter des applications IoT telles que : Arduino, UDOO, Raspberry PI, etc… et diverses plates-formes logicielles (systèmes d'exploitation et simulateurs) pour fournir les fonctionnalités IoT. RTOS Contiki, TinyOS, LiteOS et Riot OS sont des systèmes d'exploitation en temps réel pour le développement IoT.
- **Services** : les services IoT peuvent être classés en quatre classes : services liés à l'identité, services d'agrégation d'informations, services collaboratifs-conscients et services omniprésents. Les services liés à l'identité identifient chaque objet que chaque application apporte du monde réel au monde virtuel. Les services d'agrégation d'informations collectent et résument les mesures sensorielles brutes qui doivent être traitées et signalées à

l'application IoT. Les services collaboratifs-conscients s'ajoutent aux services d'agrégation d'informations et utilisent les données obtenues pour prendre des décisions et réagir en conséquence. Les services omniprésents visent toutefois à fournir les services collaboratifs-conscients à tout moment, à quiconque en a besoin.

- **Sémantique** : la sémantique est la capacité d'extraire des connaissances intelligemment par différentes machines pour fournir les services requis. L'extraction de connaissances comprend la découverte et l'utilisation de l'ensemble des ressources d'informations et de modélisation. En outre, elle comprend la reconnaissance et l'analyse de données pour fournir le service exact.

A ce stade, nous pouvons remarquer que :

- L'étiquette RFID représente une simple puce ou une étiquette attachée pour fournir l'identité de l'objet. Le lecteur RFID transmet un signal d'interrogation à l'étiquette et reçoit le signal reflété de la balise, qui à son tour est transmis à la base de données. Cette dernière se connecte à un centre de traitement pour identifier des objets sur la base des signaux reflétés dans une plage allant de (10 cm à 200 m) [13].
- La technologie de communication UWB est conçue pour prendre en charge les communications dans une zone de couverture basse gamme utilisant une faible énergie et une large bande passante.
- Le LTE est à l'origine un système sans fil standard de communication pour le transfert de données à grande vitesse entre téléphones mobiles basés sur les technologies de réseau GSM / UMTS. Il peut couvrir les déplacements rapides d'appareils et assurer la multidiffusion et les services de radiodiffusion.
- LTE-A (LTE Advanced) est une version améliorée du LTE comprenant une extension de bande passante qui prend en charge jusqu'à 100 MHz, une couverture étendue, un débit supérieur et des latences plus faibles.

I.2.4 Domaines d'application

Plusieurs domaines d'application sont touchés par l'IoT, parmi ces domaines nous pouvons citer : le transport, la santé, l'industrie, le secteur public, la domotique, et l'agriculture [15, 16].

- **Le transport :** depuis l'arrivée de l'IoT, le nombre de véhicules intelligents ne cesse d'augmenter. Presque tous les véhicules vendus aujourd'hui comportent des capteurs et des moyens de communication qui leur permettent de traiter la congestion du trafic, la sécurité, la pollution et le transport efficace des marchandises, etc. L'objectif du transport intelligent

et de la logistique intelligente est de donner à une voiture la capacité de communiquer avec un autre véhicule ou avec un centre de surveillance pour qu'elle puisse éviter les accidents et ainsi éviter tout frais relatifs telle que l'assurance. L'IoT peut permettre par exemple à une voiture qui vient de subir un accident, d'appeler automatiquement les secours, il peut même fournir la localisation du véhicule et établir une communication entre les usagers.

- **La santé :** l'utilisation de l'IoT dans le secteur de la santé a permis au patient ainsi qu'à son docteur d'échanger des informations en temps réels. L'IoT fournit une plate-forme parfaite pour réaliser la vision des soins de santé omniprésents en utilisant des capteurs de surface corporelle et un arrière-plan IoT pour télécharger les données sur des serveurs. Un exemple d'application de ce domaine est l'utilisation des capteurs corporels afin de surveiller les personnes âgées à domicile, ce qui permet aux médecins de réduire les coûts d'hospitalisation grâce à une intervention et un traitement précoce.
- **L'industrie :** dans un environnement de travail, l'IoT permet de collecter les informations à partir de différents objets connectés pour ensuite les diffuser de manière sélective. Dans la surveillance des chaînes de production, des capteurs peuvent être placés sur les équipements afin de détecter les défaillances imminentes pour que le matériel puisse être réparé ou retiré de la production jusqu'à sa réparation, réduisant ainsi les coûts d'exploitation, tout en améliorant les temps de bon fonctionnement et la gestion des performances des actifs.
- **Le secteur public :** les services publics utilisent des applications IoT pour la gestion des ressources afin d'optimiser les coûts par rapport aux bénéfices. Le réseau intelligent et le comptage intelligent dans le secteur public peuvent garantir une consommation d'énergie efficace qui peut être obtenue en surveillant en permanence chaque point d'électricité dans une maison et en utilisant ces informations pour modifier la façon dont l'électricité est consommée, ils peuvent aussi avertir les utilisateurs de pannes à grande échelle, voir les coupures d'eau, d'électricité, etc…
- **La domotique :** la domotique se définit comme un système qui permet l'automatisation de gestion de certaines fonctions d'un habitat. Elle vise à offrir un confort de vie en offrant le contrôle à distance de tout élément de la maison, à assurer la protection des personnes âgées et des biens en prévenant des risques d'accident (incendie, fuite de gaz, etc.).
- **L'agriculture :** l'objectif principal de l'IoT dans les systèmes agricoles est de renforcer leur capacité, et d'assurer la sécurité alimentaire. Les capteurs, l'imagerie satellitaire, et les systèmes de géolocalisation peuvent être utilisés par le fermier pour récolter des informations sur l'état du sol, le taux d'humidité, etc. pour qu'il puisse garantir une production optimale.

I.2.5 Normes communes à l'IoT

De nombreuses normes IoT sont proposées pour faciliter et simplifier le travail des programmeurs d'applications et des fournisseurs de services. Différents groupes ont été créés pour fournir des protocoles à l'appui de l'IoT, y compris les efforts menés par le W3C (World Wide Web Consortium), l'IETF (Internet Engineering Task Force), l'EPCglobal (Electronic Product Code), l'IEEE et l'ETSI (European Telecommunications Standards Institute). Les protocoles de L'IoT sont classés en quatre grandes catégories, à savoir : protocoles d'application, protocoles de découverte de service, protocoles d'infrastructure et autres protocoles influents comme le montre le Tableau I.1. Cependant, tous ces protocoles doivent être regroupés pour fournir une application IoT donnée. De plus, en fonction de la nature de l'application IoT, certains standards ne sont pas forcément nécessaires dans une application [13,17].

<table>
<tr><td colspan="2">Protocole d'application</td><td>DDS</td><td>CoAP</td><td>AMQP</td><td>MQTT</td><td>MQTT-SN</td><td>XMPP</td><td>HTTP REST</td></tr>
<tr><td colspan="2">Découverte de service</td><td colspan="3">mDNS</td><td colspan="4">DNS-SD</td></tr>
<tr><td rowspan="4">Protocole d'infrastructure</td><td>Protocole de routage</td><td colspan="7">RPL</td></tr>
<tr><td>Couche réseau</td><td colspan="5">6LoWPAN</td><td colspan="2">IPv4/IPv6</td></tr>
<tr><td>Couche de lien</td><td colspan="7">IEEE 802.15.4</td></tr>
<tr><td>Couche physique</td><td colspan="2">LTE-A</td><td colspan="2">EPCglobal</td><td>IEEE802.15.4</td><td>Z-WARE</td></tr>
<tr><td colspan="2">Protocole influent</td><td colspan="5">IEEE 1888.3. IPSec</td><td colspan="2">IEEE 1905.1</td></tr>
</table>

Tableau I.1 Effort de standardisation en faveur de l'IoT [13].

I.2.6 Cloud computing

L'IoT a rapidement trouvé son chemin dans notre vie moderne, dans le but d'améliorer la qualité de vie en connectant de nombreux appareils intelligents, technologies, et applications. Cependant, ces objets connectés produisent de grandes quantités de données dont le stockage et le traitement entrent dans le cadre du Big data. Afin de prendre en charge ce dernier, la notion de cloud computing a fait surface.

I.2.6.1 Définition

Selon la définition officielle du NIST (National Institute of Standards and Technology), le cloud computing est un système qui offre un accès réseau en temps réel à des ressources informatiques qui peuvent être facilement mis en service et libérés à la demande (par exemple, réseaux, serveurs, stockage, applications et services [18,19].

I.2.6.2 Modèles de déploiement

Le cloud computing peut être classé en fonction du modèle de déploiement et du modèle de prestation de services [18, 19, 20].

- **Cloud privé :** il fournit des services à un seul client ou à une seule organisation, et il est géré uniquement dans ce but. Le service peut être administré par l'organisation elle-même ou par un fournisseur externe.
- **Cloud communautaire :** l'infrastructure est partagée par plusieurs entreprises qui ont des intérêts communs. Il peut être géré par les entreprises elles-mêmes ou par un tiers.
- **Cloud public :** un modèle de cloud géré par divers fournisseurs de services de cloud computing. Un fournisseur de cloud public propose ses services à tous ses clients depuis Internet. Dans ce modèle de déploiement, c'est les fournisseurs qui s'occupent de la gestion des ressources, telles que les applications et le stockage.
- **Cloud hybride :** un modèle de cloud consistant à combiner plusieurs modèles. Comme son nom l'indique, il s'agit tout simplement d'un mélange de cloud privé et public. Pour une entreprise aux besoins complexes, cela revient à emprunter le meilleur des deux mondes.

I.2.6.3 Modèles de service

Les trois modèles de services de base proposés par les fournisseurs d'informatique du cloud sont IaaS (Infrastructure as a Service), PaaS (plateform as a Service) et SaaS (Software as a Service) [19, 20].

- **IaaS** : est un modèle de services cloud qui fournit une infrastructure complète (serveurs, stockage, bande passante, connexion réseau, software, hardware) à distance pour une entreprise. Le fournisseur de l'infrastructure est responsable de sa maintenance et de sa sécurité. Les entreprises, selon leurs croissances, peuvent augmenter leurs infrastructures au besoin. Amazon EC2, Windows Azure sont les meilleurs exemples de fournisseurs qui proposent un service IaaS.
- **Paas :** est un modèle de services cloud qui fournit la plateforme et l'environnement informatique nécessaire aux développeurs pour mettre en place leurs différents services tels

que les systèmes d'exploitation, environnement de script serveur, système de gestion de bases de données, logiciel serveur, support, stockage, accès réseau, outils de design et de développement, hébergement, sur internet. Toutefois, les entreprises qui développent leur propre logiciel interne peuvent aussi utiliser les services PaaS, notamment pour créer des environnements de développement et de test distincts et délimités. La plateforme office AZURE ou SAP (Systems, Applications and Products for data processing) sont les meilleurs exemples de fournisseurs qui proposent un service PaaS.

- **SaaS :** ce modèle de service se réfère à tout service cloud permettant aux clients d'avoir accès à des applications logicielles sur internet, il évite le support et la maintenance du logiciel. L'infrastructure du cloud est installée avec divers logiciels et les utilisateurs du cloud peuvent être en mesure d'utiliser les logiciels déjà installés dedans. Du coup, l'installation de logiciels sur leurs machines peut être éliminée. Les logiciels office sont le meilleur exemple d'entreprises utilisant un service SaaS, ainsi que Google Apps, Twitter, Facebook, Instagram etc...

I.2.6.4 Limitations

Bien que le cloud computing offre plusieurs avantages aux utilisateurs et aux fournisseurs de services par rapport aux paradigmes informatiques traditionnels, il présente également certaines limitations telles que [21] :

- Étant donné que le cloud computing est essentiellement basé sur Internet, il est indispensable de disposer d'une connectivité internet fiable avec une bande passante suffisante pour accéder aux services. Si la panne de liaison survient pour une raison quelconque, le système complet serait inaccessible.
- Etant situé sur internet qui est un vaste réseau hétérogène comportant de nombreuses topologies, vitesses et technologies sans contrôle central, de nombreux problèmes restent à résoudre, en particulier ceux liés à la qualité de services. La latence du réseau est un problème qui affecte gravement la qualité de service. Les applications en temps réel avec lesquelles les utilisateurs interagissent directement sont fortement affectées par les retards et la gigue de retard causés par la latence dans les réseaux.
- L'autre problème majeur auquel le cloud computing est confronté est la sécurité et la confidentialité. Les demandes des utilisateurs, la transmission des données et les réponses du système doivent traverser un grand nombre de réseaux intermédiaires en fonction de la distance qui sépare les utilisateurs des systèmes. Lorsque les données des clients se trouvent dans un cloud public, leur intégrité et leur confidentialité risquent d'être compromises.

I.2.7 Paradigmes de bord

Pour diverses raisons, le paradigme cloud computing n'est pas en mesure de répondre à certaines exigences (par exemple, faible temps de latence, aide à la mobilité) indispensables pour plusieurs applications comme la santé en ligne. Pour répondre à ces exigences, divers paradigmes, tels que le fog computing, le edge computing, ou encore le mobile cloud computing ont fait leur apparition ces dernières années [22].

I.2.7.1 Fog computing

Le fog computing est une solution prometteuse pour l'IoT, qui place les dispositifs informatiques plus proches de la source de données. Ces dispositifs informatiques de bord traditionnels tels que les commutateurs, routeurs, points d'accès, etc… sont équipés d'une infrastructure informatique, et de modèles de gestion. En conséquence, le traitement de certaines données peut être exécuté plus près de la source de données, tout en distribuant les ressources, en réduisant le besoin de communication de données à sauts multiples, en réduisant la latence et en promouvant la flexibilité des services. Bien que les ressources du FC soient limitées en termes d'énergie et de calcul par rapport au cloud, ils sont suffisamment souples pour être personnalisées en fonction du contexte de l'application [22].

A. Définition

Le terme "Fog Computing" ou "brouillard informatique" a été introduit par Cisco (Computer Information System Company*)* en tant que nouveau modèle pour faciliter le transfert de données sans fil vers des périphériques distribués dans le paradigme du réseau de l'IoT. Selon Firdhous et al. [20], la raison pour laquelle ce terme a été inventé pour identifier ce modèle est que le FC n'est rien d'autre qu'un cloud plus proche du sol. Par conséquent, le cloud computing réalisé au plus près des réseaux des utilisateurs finaux est donc considéré comme un fog computing. Le FC est une plate-forme virtualisée située généralement entre les machines des utilisateurs finaux et les centres de données cloud hébergés sur internet comme le montre la Figure I.3. Le FC peut donc offrir une meilleure qualité de service en termes de délai, de consommation électrique, de réduction du trafic de données sur internet, etc. [23].

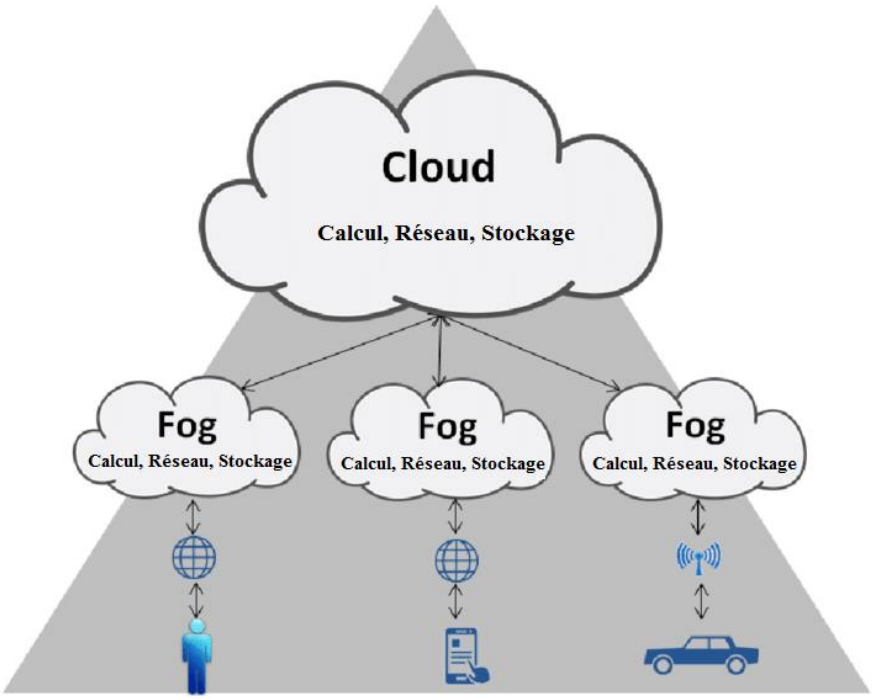

Figure I. 3. Architecture cloud/ fog computing [24].

B. Fog vs cloud

D'après les définitions, nous constatons que techniquement, le fog computing ressemble beaucoup au cloud computing, dans le sens où ils sont tous deux constitués de systèmes virtuels offrant la flexibilité de la fourniture à la demande de ressources de calcul, de stockage et de réseau. Mais par rapport au cloud, le fog est mis en œuvre de manière très proche des utilisateurs finaux. Le Tableau I.2 résume les résultats de la comparaison. Il montre que les caractéristiques du cloud computing présentent de très sérieuses limitations en ce qui concerne la qualité de service requise par les applications en temps réel nécessitant une action presque immédiate du serveur [21, 23].

Exigences	**Cloud computing**	**Fog computing**
Latence	Elevée	Faible
Emplacement des nœuds des serveurs	Sur internet	Au bord du réseau local
Retard de gigue	Elevé	Très faible
Distance entre le client et le serveur	Plusieurs sauts	Un saut
Sécurité	Indéfini	Peut être défini
Attaque sur les données en route	Probabilité élevée	Très faible probabilité
Connaissance de l'emplacement	Non	Oui

Orchestration géographique	Centralisée	Distribuée
Nombre de nœuds de serveur	Peu	Beaucoup
Soutien à la mobilité	Limité	Supporté
Interactions en temps réel	Limitée	Supportée
Ajout/ suppression de nœuds	Difficile	Flexible
Usage d'énergie des nœuds/serveurs	Elevé	Faible
Lien de communication	Fixes	Variables
Tolérance aux pannes du nœud / serveur	Faible	Elevée
Maintenance des machines virtuelles	Complexe	Simple
Prix	Elevée	Faible
Configuration des machines virtuelles	Rigide	Ajustable

Tableau I. 2 Cloud computing vs fog computing [19-24].

C. Avantages du fog

Les principaux avantages du FC sont [25, 26] :

- **Garder les données de l'utilisateur proches :** pour éliminer les retards dans le transfert de données, le FC permet de garder les données proches de l'utilisateur au lieu de les stocker dans des centres de données lointains.
- **Répartition géographique dense :** le FC crée un réseau de périphérie situé à divers endroits pour étendre les services du cloud, l'infrastructure géographiquement isolée aide à gérer et à analyser les données à grande échelle plus rapidement.
- **Grand soutien à la mobilité :** il y a une augmentation considérable dans la quantité de données et de périphériques. Le fog computing permet de gérer ces grandes données et informations et fournit un moyen plus rapide pour accéder aux données et les analyser.
- **Économiser de l'espace de stockage :** le fog serait une excellente option pour empêcher toute information inappropriée ou non pertinente de se rendre dans l'ensemble du réseau ; cela permet d'économiser de l'espace de stockage et réduit les délais.
- **Prise en charge du scénario IoT :** le fog computing peut être appliqué dans plusieurs domaines où l'IoT est présent. Cela pourrait jouer un rôle important dans diverses applications de l'IoT comme dans le fonctionnement des feux de circulation et des véhicules

intelligents, des réseaux intelligents, des villes intelligentes, des réseaux de capteurs sans fil et actionneur et des systèmes cyber-physiques.

- **Extension du cloud et intégration avec d'autres services :** le fog computing ne peut pas être considéré comme un remplacement du cloud, il s'agit d'une sorte d'extension pour fournir des entrées filtrées et plus rapides au cloud et aux utilisateurs.

I.2.7.2 Edge computing

A. Définition

Le "Edge computing" ou "informatique de bord" se réfère aux technologies habilitantes permettant d'effectuer le calcul en périphérie du réseau, le "bord" est définit comme toutes ressources informatiques se situant au point d'extrémité. La logique du edge computing est que l'informatique devrait se produire à proximité des sources de données [27, 28].

B. Différence entre le edge computing et le fog computing

Le fog Computing et le edge Computing offrent les mêmes fonctionnalités en termes de transfert de données et d'intelligence vers des plates-formes analytiques situées sur ou à proximité de l'origine des données, qu'il s'agisse d'écrans, de haut-parleurs, de moteurs, de pompes ou de capteurs. Les deux technologies peuvent aider les entreprises à réduire leur dépendance aux plates-formes basées sur le cloud pour analyser les données, ce qui entraîne souvent des problèmes de latence, et permettent à la place de prendre plus rapidement des décisions en fonction des données. La principale différence entre eux se résume à l'endroit où le traitement des données a lieu [28, 29].

Comme illustré dans la Figure I.4, le edge computing se produit généralement directement sur les périphériques auxquels les capteurs sont connectés ou sur un périphérique de passerelle physiquement « proche » des capteurs. Le fog computing déplace les activités de calcul de pointe vers les processeurs connectés au réseau local ou directement dans le matériel du réseau local, de sorte qu'ils puissent être physiquement plus éloignés des capteurs et des actionneurs. Ainsi, avec le fog computing, les données sont traitées dans un nœud fog ou une passerelle IoT située dans le réseau local, et avec le edge computing, les données sont traitées sur l'appareil ou le capteur lui-même sans être transférées nulle part. Le edge computing a donc une portée plus limitée, du coup il ne peut pas créer de connexions directes entre deux terminaux ou entre un terminal et une passerelle IoT à elle seule, pour cela il faut du fog computing. De plus, étant donné que le fog permet de collecter des données à partir de différents appareils, il a également une grande capacité à traiter plus

de données que le edge, donc le meilleur moment pour mettre en œuvre le fog computing est lorsque des millions d'appareils connectés partagent les données dans les deux sens.

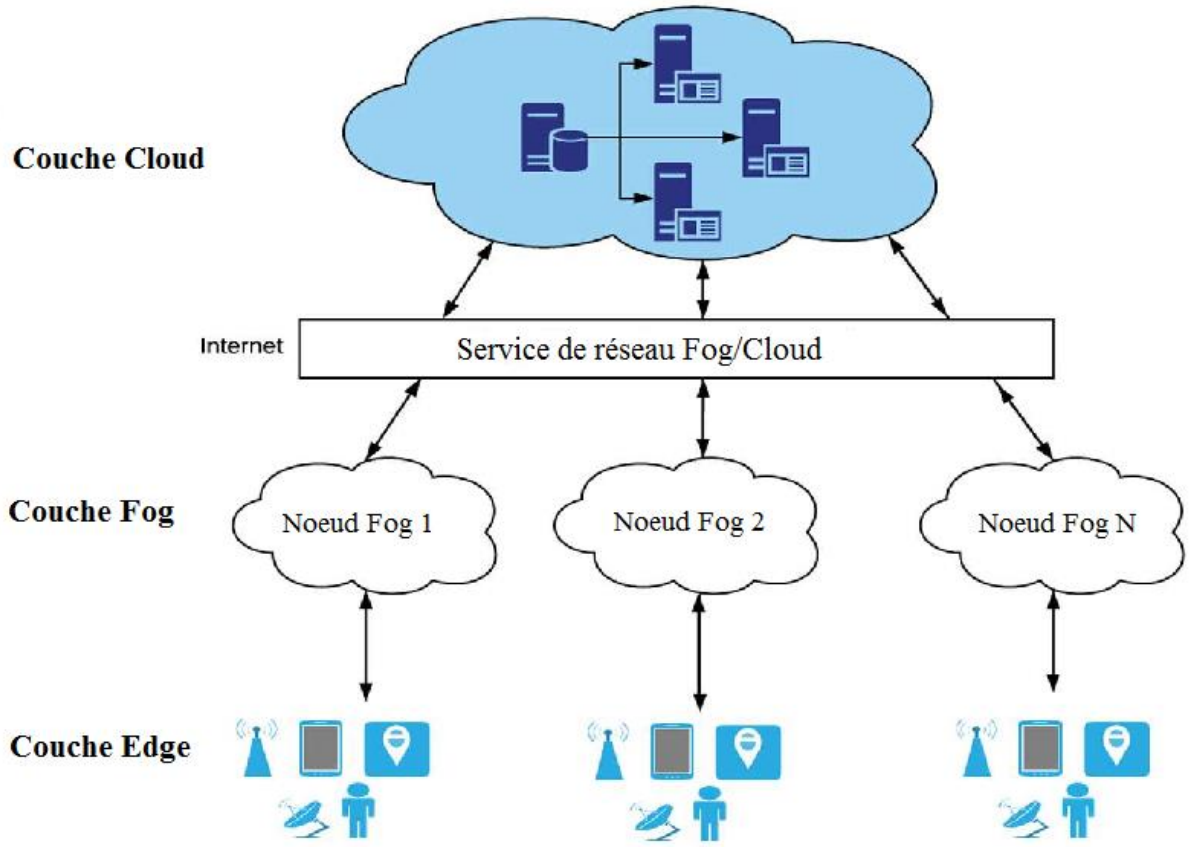

Figure I. 4. Architecture edge/fog computing [30].

I.2.7.3 Cloud computing mobile

A. Définition

Le mobile cloud computing (MCC) est l'association du cloud computing et de l'informatique mobile. Il se concentre principalement sur la notion de « Délégation mobile » : en raison des ressources limitées disponibles pour les appareils mobiles, il doit déléguer le stockage de données en vrac, et l'exécution de tâches de calcul intensives à une autre entité. Dans le concept original du MCC, introduit en 2009, seules les plates-formes cloud centralisées ont été considérées comme les solutions les plus viables pour mettre en œuvre l'exécution à distance de tâches. Plus tard, d'autres chercheurs ont élargi la portée du MCC. Dans cette nouvelle vision, les tâches pourraient également être déléguées à des dispositifs situés au bord du réseau [22].

B. Domaines de recherche

L'un des domaines de recherche les plus actifs du MCC est la délégation de tâches aux services externes. Il existe différentes solutions qui permettent aux applications de migrer soit une partie de leur code des appareils mobiles au calcul basé sur le cloud des ressources situées au bord. Les applications sont généralement mises en œuvre à l'aide de Framework tels que .NET. Ensuite, une partie de l'application mobile (y compris image mémoire, état de l'unité central de traitement, et autres) est chargé dans le clone. Enfin, certaines approches utilisent des infrastructures d'agents mobiles, où le périphérique mobile crée un agent mobile qui traitera des informations en son nom.

Un autre domaine de recherche important est la mise en œuvre des ressources informatiques cloud situées au bord. Il y a deux grandes stratégies : entités informatiques immobiles proches (ex : serveurs de virtualisation) et entités informatiques mobiles.

L'élément central de la première stratégie est le cloudlet (petit cloud). Ce concept fait référence à une petite infrastructure cloud située à proximité des mobiles utilisateurs. Cette petite infrastructure peut être déployée dans des locaux d'affaires (par exemple, des magasins spécialisés, des bâtiments). Elle permet aux appareils de charger une petite superposition de machine virtuelle sur une version d'images de machines virtuelles préexistantes. Plusieurs tests ont montré que les cloudlets améliorent le temps de réponse et la consommation d'énergie des dispositifs mobiles (respectivement 51% et 42%) par rapport au cloud centralisé [22].

Il existe plusieurs exemples de la deuxième stratégie qui spécifient tous une plate-forme informatique distribuée sur un cluster composé de périphériques proches jouant le rôle de serveurs basés sur les principes du cloud computing. Les éléments du cluster peuvent être des dispositifs mobiles, des dispositifs et entités IoT, ou une combinaison de plusieurs types de dispositifs. En raison des ressources limitées pour les appareils qui forment le cluster distribué, cette stratégie n'utilise pas de techniques de virtualisation. Au lieu de cela, certaines implémentations font appel à des algorithmes parallèles spécifiques tels que MapReduce, tandis que d'autres prennent une approche plus générale et permettent divers types de calcul de tâches intensives. Dans presque tous les cas, un contrôleur est chargé de recevoir les tâches et de découvrir quel appareil pourrait les exécuter de manière optimale [22].

I.3 Systèmes de soins de santé

I.3.1 Définition

Le système de soin est un sous-système du système de santé. Il comprend tous les services qui visent à fournir des prestations sanitaires à la population afin d'améliorer sa santé. Cependant, le système de santé ou le système de soins de santé est un terme qui est utilisé pour définir l'ensemble des moyens organisationnels, institutionnels, et stratégiques qui œuvrent à produire des interventions sanitaires de qualité [31].

I.3.2 Soins de santé 4.0

Comme toutes les autres industries (mécanique, électrique, agricole, ou civil), le secteur de la santé est passé de la génération 1.0 à la génération 4.0. L'industrie de la santé débuta en 1970 avec l'émergence des systèmes informatiques modulaires, mais les efforts étaient préliminaires et les ressources limitées. Cette période pourrait être appelée soins de santé 1.0. Au cours des quinze années qui ont suivi, les systèmes de soins de santé ont commencé à être mis en réseau avec l'utilisation des DSE, une version alternative du tableau de données des patients, cette période pourrait être appelée soins de santé 2.0. Au début des années 2000, les dispositifs portables et implantables ont émergé et quelques années plus tard, le concept de l'IoT a fait son apparition. L'intégration de toutes ces données avec les systèmes de DSE en réseau a permis l'émersion des soins de santé 3.0. Nous vivons actuellement l'émergence des soins de santé 4.0. C'est le rapprochement de toutes ces technologies, associées à la collecte de données en temps réel et en grande quantité et à l'utilisation accrue de l'intelligence artificielle. Les médecins disposent d'une immense quantité de données, mais les véritables facteurs essentiels sont la capacité d'extraire des informations de ces données et la portabilité de ces données. Des analyses améliorées des informations extraites permettent un diagnostic différentiel et des réponses médicales prédictives, rapides et novatrices et la portabilité des données permet aux patients et à leurs médecins d'y accéder à tout moment. Les soins de santé 4.0 permettent de valoriser les données de manière plus cohérente et efficace [32].

I.4 Systèmes de soins de santé basés sur l'IoT

I.4.1 Avantages de l'IoT dans les soins de santé

La prolifération et l'adoption généralisée des nouvelles technologies, en particulier de l'IoT et des appareils intelligents, ont créé de nouveaux modes de prestation des soins de santé, améliorant ainsi la santé et le bien-être humains. Les systèmes de soins de santé dotés de la vision IoT offrent de

nombreux avantages, tels que la disponibilité et l'accessibilité, la possibilité de fournir un système plus « personnalisé » et des soins de santé rentables et de haute qualité. Par conséquent, l'IoT est considéré comme une solution prometteuse pour le secteur de la santé, car il place le patient au centre du processus de traitement, permet l'autogestion de sa propre maladie, donne aux professionnels de la santé un accès plus rapide et sécurisé à toutes les informations dont ils ont besoin pour prendre soin de leurs patients à distance grâce à un équipement de surveillance en réseau associé [33].

En d'autres termes, les soins de santé pilotés par l'IoT permettent de [33] :

- Détecter et collecter des données relatives à la santé des patients à partir de divers capteurs de manière distante et sécurisée.
- Appliquer une variété de techniques et d'algorithmes d'exploration de données afin de découvrir des modèles cachés et de détecter toute anomalie, et sur la base des connaissances précieuses acquises, permet de prédire et de prendre des décisions.
- Partager les données via la connectivité sans fil avec ceux qui peuvent faire des commentaires adéquats et opportuns.

I.4.2 Architecture des soins de santé basée sur l'IoT

Les éléments suivants sont essentiels à la réalisation du système de santé basé sur l'IoT : une variété de capteurs, microcontrôleurs, microprocesseurs, passerelles spécifiques au secteur de la santé et le cloud. En d'autres termes, la majorité des systèmes de santé utilisant l'IoT se basent sur une architecture à trois couches comme illustré dans la Figure I.5, qui consiste en [33] :

- Couche de détection / perception dont les fonctions principales sont la détection et la collecte de données, ainsi que certaines actions de communication et de contrôle.
- Couche réseau responsable de la communication, de la connectivité, du routage, etc.
- Couche d'application constituée de modules fonctionnels pour les systèmes d'application et les utilisateurs. À ce niveau, les données détectées et collectées sont utilisées pour des analyses, des calculs, des visualisations, etc.

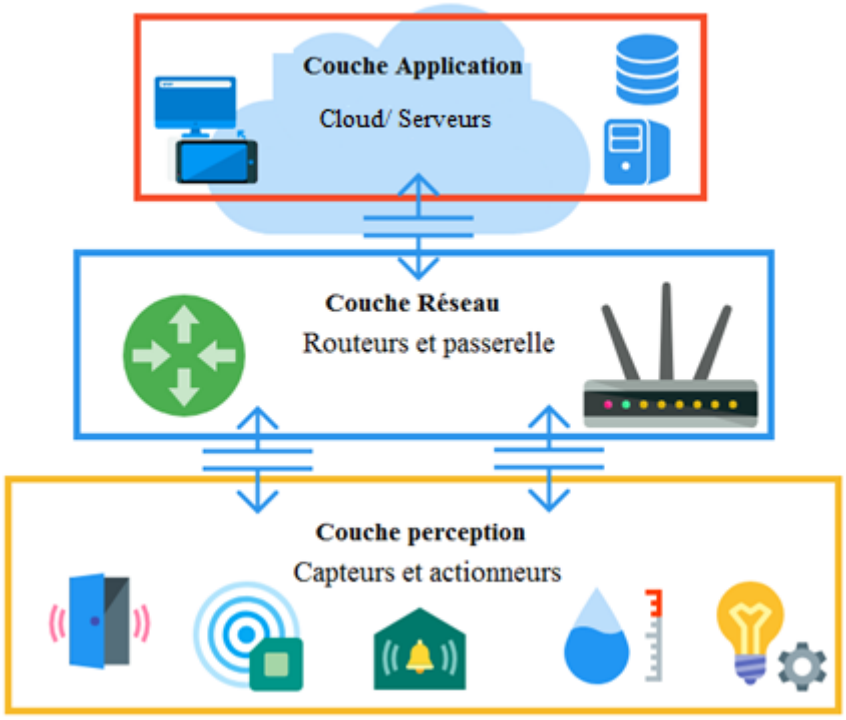

Figure I. 5. Architecture à trois couches d'un système de santé utilisant l'IoT [34].

I.4.3 Scénarios de déploiement des systèmes de soins de santé

Il existe cinq scénarios de déploiements des systèmes de soins de santé basés sur l'IoT, les scénarios diffèrent en termes d'utilisateurs et d'intervenants impliqués, d'appareils et de connectivité [33] :

- **Mobile :** dans ce scénario, les téléphones mobiles des utilisateurs agissent comme un concentrateur (hub) entre les capteurs et le cloud.
- **Traitement à domicile :** à domicile, la connectivité est souvent via l'accès internet du patient. Cela a une influence sur la propriété de l'appareil, la convivialité requise, la maintenabilité et comment les perturbations peuvent être atténuées.
- **Hôpital :** dans un hôpital, les appareils sont souvent propriétaires, et sont généralement détenus et entretenus par l'hôpital lui-même. Les systèmes sont considérablement plus complexes, ce qui oblige que les utilisateurs des applications soient des professionnels qualifiés.
- **Locaux non hospitaliers :** comme les hôpitaux, ce scénario couvre les points de soins professionnels, mais avec moins de personnel et infrastructure. Exemples : cliniques, cabinets médicaux ou maisons de repos. Les principaux appareils sont détenus et entretenus par la clinique, mais les patients doivent parfois connecter l'équipement personnel au réseau.

- **Transport :** ce scénario couvre la connectivité dans une ambulance ou un hélicoptère. Il est similaire au scénario de déploiement non-hospitalier, mais avec la complexité supplémentaire que l'infrastructure doit être mobile, par exemple en utilisant une connexion cellulaire.

I.4.4 Fog computing dans les soins de santé basés sur l'IoT

Les services du cloud computing ont été largement accrédités pour prendre en charge les solutions de soins de santé compatibles IoT, fournissant des solutions pour l'évolutivité, l'analyse des données et la fiabilité. Toutefois, transférer dans le cloud des volumes croissants de données liées à la santé, qui sont en croissance rapide et la plupart du temps non structurées, et renvoyer les données de réponse nécessite une plus grande largeur de bande, un temps considérable et peut entraîner des problèmes de latence, qui ne sont pas tolérables compte tenu des contraintes de temps dans les applications d'intervention d'urgence, telles que les soins de santé. Donc les applications qui s'appuient entièrement sur les centres de données distants sont inacceptable en raison de la sécurité des données du patient en cas de défaillance du réseau et du centre de données [35].

Le FC est une solution prometteuse car il rapproche les ressources informatiques de la source de données IoT. Dans cette solution, les dispositifs informatiques de périphérie traditionnels, tels que les commutateurs, les routeurs, les dispositifs informatiques compacts, etc., sont équipés d'une infrastructure informatique, de services et de modèles de gestion permettant de mettre en œuvre des applications locales maigres. Ainsi, certains traitements de données peuvent être exécutés plus près de la source de données, ce qui permet de répartir la demande en ressources, de réduire le besoin de communication de données à plusieurs sauts, de réduire le temps de latence et la consommation d'énergie et de promouvoir la flexibilité des services. Bien que les ressources du FC soient limitées en termes d'énergie et de puissance de calcul par rapport au cloud computing, elles sont suffisamment flexibles pour être personnalisées en fonction du contexte de l'application [35].

I.4.4.1 Architecture du système de santé basé sur le fog computing

Comme le montre la Figure I.6, l'architecture du système de santé basé sur le FC est composée de trois couches principales : Couche de Dispositifs Médicaux (CDM), Couche Fog (CF) et Couche Cloud (CC). Elle fournit une solution complète à partir de l'acquisition des données, du traitement de données à l'analyse de données volumineuses sur une plate-forme cloud [32].

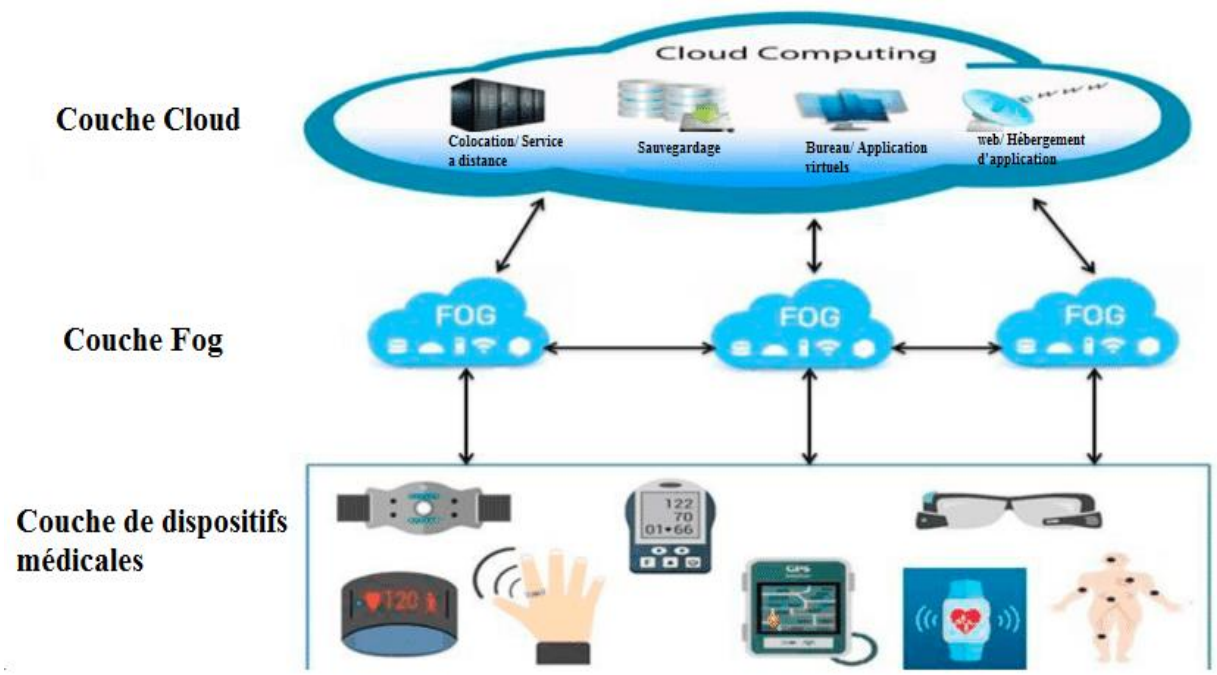

Figure I. 6. Architecture du système de santé basé sur le fog computing [36].

a. Couche de dispositifs médicaux

Un vaste ensemble de dispositifs médicaux (DM) basés sur l'IoT, tels que des dispositifs portables, des capteurs et des smartphones, permettent aux utilisateurs des soins de santé 4.0 de surveiller l'état de santé en temps réel des patients. L'état de santé peut être capturé en temps réel par n'importe quel ordinateur personnel ou téléphone portable et faire correspondre les informations en toute sécurité avec la plate-forme cloud. Dans la Figure I.6, la CDM est la première couche, où différents capteurs tels que les montres intelligentes, les lunettes intelligentes, le lecteur de glucose, le capteur ECG (electrocardiogram), etc… sont utilisées pour acquérir les données des patients. Ces capteurs répartis géographiquement génèrent d'énormes flux de données de détection qui doivent être traités avec soin. Dans ce scénario spécifique, une quantité énorme de protocoles de réseau personnel et de protocoles de réseau de capteurs sans fil est utilisée pour une connexion correcte, tel que le WiFi pour les réseaux distants et Bluetooth Low Energy (BLE) pour les connexions à courte distance.

b. La couche fog

Une fois que les données brutes sont générées à partir de la CDM, elles sont transmises à la couche suivante, à savoir CF, pour être traitées et transmises pour une analyse. Cette couche est équipée de divers nœuds de calcul à faible puissance et haute performance comparé aux DM appelés Nœud Fog (NF). Chaque NF est connecté à un groupe local de DM qui couvre généralement une petite communauté et effectue une analyse des données à temps. La criticité temporelle et la faible

latence sont l'un des critères clés des soins de santé 4.0. La CF couvre tous les défis des soins de santé 4.0 en accompagnement avec le cloud computing, le big data analytique, l'intelligence artificielle (IA) et les systèmes de DSE.

Le modèle cloud computing conventionnel qui collecte et analyse les données sensibles, les bio-signaux et les signes vitaux en temps réel des patients sur une large zone géographique prend beaucoup de temps et ne peut pas être compatible avec la santé 4.0. La méthode la plus efficace pour s'attaquer à ces problèmes consiste à utiliser le fog pour amener le cloud et ses services à un autre niveau. Dans la plate-forme de soins de santé 4.0, un système peut analyser les données sensibles au facteur temps et prendre la décision en fonction du placement des NF. Ces NF sont placés le plus près possible des dispositifs médicaux intelligents. Sinon, le reste des données est envoyé au cloud en tant que stockage principal. Un autre problème rencontré dans les soins de santé 4.0 est la bande passante du réseau en raison de la transmission importante de données, par exemple, le dispositif ECG génère plusieurs giga octets de données brutes de séries chronologiques au cours d'une journée. Cependant, il n'est pas faisable d'envoyer cette grande quantité de données de milliers de patients sur le cloud. Par conséquent, les NF sont suffisamment souples pour traiter ces données, les filtrer, les compresser et les transférer au Cloud pour stockage.

La CF a différentes caractéristiques pour diminuer la latence et augmenter la bande passante du réseau, telles que :

- Le NF est compétent pour maintenir une connectivité fiable et sécurisée entre les appareils médicaux et les réseaux internes et externes. Ceci comprend la traduction de protocole, la commutation, le routage, la sécurité et l'analyse de réseau.
- Le NF fournit un flux de données bidirectionnel. Les données ont été acheminées vers le cloud par les MD via la CF et seront disponibles pour le médecin ou patient sur demande. Le NF envoie périodiquement les données agrégées au cloud.
- Le nœud fog est doté d'une interface multi-standard compatible avec les protocoles de réseau de capteurs PAN (Personal Area Network) et sans fil, par exemple, WiFi, 3G / 4G, RFID, BLE, Zigbee et de protocoles câblés tel qu'Ethernet.
- Le NF effectue la compression, le filtrage, l'agrégation et le formatage des données médicales brutes recueillies auprès des médecins.
- La CF fournit des mesures de sécurité multicouches pour le cryptage, l'authentification et le contrôle d'accès.

c. La couche Cloud

La couche la plus élevée est la couche cloud qui contient un centre de données de calcul puissant, elle fournit un contrôle centralisé et une large surveillance. Dans cette couche, la capacité de stockage et le calcul distribué permettent une analyse à long terme, complexe et comportementale, une modélisation des relations et la reconnaissance de formes à long terme. Elle est composée de serveurs cloud qui effectuent des décisions dynamiques. De plus, le serveur cloud est responsable du stockage supplémentaire et de l'agrégation des données des patients envoyées par la CF. Les praticiens peuvent accéder à ces données à des fins de facturation ou de synthèse. Les patients sont en mesure d'accéder à leurs antécédents médicaux / factures actuels et historiques au moyen d'une application mobile ou d'une interface Web.

I.4.4.2 Avantages du fog computing dans les soins de santé

Le FC offre trois avantages majeurs pour les systèmes de soins de santé : une faible latence, la confidentialité et la résilience face au cloud computing [33, 37].

- **Faible latence :** le FC fonctionne à proximité des périphériques de stockage et des ressources informatiques de l'utilisateur. Cela aide à réduire les coûts de transmission des données car il traite les données avec le calcul local. Cela aide aussi à la mise en œuvre d'une latence ultra faible pour des applications en temps réel telles que les soins de santé.
- **Confidentialité :** le FC traite les données localement avant de les partager avec des serveurs tiers. Ainsi, la partie confidentielle des données peut être filtrée, cela améliore considérablement la confidentialité et la sécurité des données médicales.
- **Résilience contre les pannes du cloud / réseau :** en cas de défaillance du réseau ou du cloud, le FC permet la récupération en toute sécurité des applications et des données. De plus, il peut identifier les défaillances des liens et les signaler sans effort en utilisant d'autres ressources accessibles à l'utilisateur par exemple.
- **Flexibilité du lieu de calcul :** lorsque les problèmes d'évolutivité, de confidentialité et de fiabilité empêchent de mettre en place une solution cloud, le FC peut offrir les ressources de calcul nécessaires au sein du réseau pour répondre à la fois aux exigences réglementaires et techniques. Pour que de telles approches soient efficaces, il est important d'avoir des ressources de calcul entre les capteurs et le cloud, tout en assurant la transparence de l'exécution pour l'application, ainsi qu'une flexibilité quant à l'endroit où le calcul peut être exécuté. Avec le FC, l'emplacement peut être dynamique, il dépend du contexte, de l'environnement et des exigences d'application.

- **Intégration :** dans le paysage actuel, l'introduction de nouveaux capteurs nécessite souvent l'introduction simultanée d'une infrastructure de support. Un exemple est le système de surveillance de la fréquence cardiaque, qui nécessite une infrastructure dédiée. C'est un fardeau considérable lors de l'introduction de nouveaux dispositifs innovants. Dans l'architecture FC, de nouveaux capteurs peuvent être ajoutés à l'infrastructure existante. Le FC peut servir de couche de compatibilité pour faire la traduction entre différentes normes.
- **Mobilité des patients :** l'infrastructure spécifique à l'application limite également la zone où les patients peuvent être surveillés. C'est surtout pertinent lorsque les patients sont sur le point de quitter l'infrastructure hautement instrumentée d'un hôpital. Ces cas d'utilisation courants souvent ne couvrent pas cette transition, qui peut effectivement prolonger le séjour du patient à l'hôpital. Mais avec des ressources du FC, les transitions entre différents environnements peuvent être gérées plus progressivement.

I.4.5 Menaces de sécurité dans les soins de santé

Les systèmes de soins de santé sont vulnérables à un grand nombre d'attaques et de menaces. Ils sont fréquemment ouverts à plusieurs menaces externes et intrusions, qui pourraient pirater le réseau. Ainsi, les problèmes de sécurité et de confidentialité doivent être traités avec rigueur. L'attaquant peut cibler la disponibilité du système en capturant ou neutralisant un nœud particulier, ce qui entraîne parfois une perte de la vie d'un patient. Par exemple, l'attaquant peut capturer ou désactiver un capteur EEG (ElectroEncephaloGram) et envoyer de fausses informations au médecin, cela pourrait entraîner une situation dangereuse mettant la vie en danger ou même causant la mort du patient. Un attaquant peut également utiliser le brouillage et l'altération pour bloquer l'ensemble du réseau. Cette attaque ne peut pas bloquer les grands réseaux, mais comme les systèmes de santé sont généralement de petits réseaux, les chances de blocage du réseau sont assez élevées, et peuvent causer la perte de paquets. Il est aussi possible qu'un attaquant interfère électroniquement, endommage ou supplante le système pour obtenir les renseignements personnels sur la santé d'un patient. Il peut également utiliser la technique d'inondation pour épuiser la mémoire en envoyant à plusieurs reprises des paquets inutiles supplémentaires, que le système est incapable de gérer. Cela empêche les utilisateurs légitimes du réseau d'accéder aux services ou les ressources. Cela peut être aussi réalisé par déni de service (DoS), une attaque qui vise non seulement à perturber, subvertir et détruire le réseau, mais aussi à diminuer sa capacité à fournir les services d'urgence nécessaires [39, 40].

I.4.6 Mesures de sécurité dans les soins de santé

Afin de garantir la sécurité d'un système de santé, différentes exigences doivent être remplies, telles que [41, 42] :

- La normalisation est nécessaire pour avoir un système IoT efficace et efficient.
- La réduction de la distance entre les utilisateurs et les fournisseurs de services.
- Un cadre analytique et des algorithmes de Big data sont nécessaires pour traiter la quantité de données générées par le système de santé IoT.
- Le système de santé IoT moderne doit utiliser le stockage distribué pour stocker les informations médicales plutôt qu'un stockage de données centralisé conventionnel.
- Afin de protéger et de maintenir les ressources, la confidentialité et l'intégrité des informations médicales, les techniques avancées d'authentification et d'autorisation doivent être identifiées.
- La présence des capteurs mobiles disponibles doit être notifiée par les passerelles locales dans la couche fog. Cela permet les mises à jour nécessaires sur le réseau.
- Les capteurs médicaux doivent être détectés et adressables à tout moment.
- La mobilité est l'une des principales préoccupations du système IoT de soins de santé, le défi consiste à éviter la gigue, les retards et les interruptions du transfert de données pendant le processus de transfert.
- Le système doit être sécurisé et confidentiel et les informations de santé du patient doivent être sécurisées et correctement formatées.
- Le système doit fournir une protection contre toute rareté de sécurité, de confidentialité et d'intégrité quand ils se produisent.
- Le système doit fournir une protection contre l'accès ou l'utilisation non autorisée des informations de santé du patient [35].

Aux états unis par exemple, la HIPPA (Health Insurance Portability and Accountability Act) réglemente en outre d'autres domaines critiques comme :

- Sécuriser les dossiers de santé des patients, notamment de ceux qui n'ont pas besoin des informations qui se trouvent dans ces dossiers.
- Établir des systèmes qui nécessitent une identification de l'utilisateur à la fois les patients et le personnel médical.
- Seule la personne autorisée a le droit d'accéder aux données sensibles des applications.

- Assurer l'intégrité des informations sur la santé des patients tout au long du cycle de vie au sein du système [35].

I.4.7 Techniques utilisées pour garantir la sécurité

Diverses techniques sont utilisées pour garantir la sécurité et la confidentialité des systèmes de soins de santé. Les techniques les plus utilisées sont les suivantes [43] :

- **Authentification :** l'authentification est l'acte de confirmer l'authenticité des allégations faites par ou sur le sujet. Il remplit des fonctions vitales au sein de toutes organisations : sécuriser l'accès aux réseaux d'entreprise, protéger l'identité des utilisateurs, et s'assurer que l'utilisateur est vraiment ce qu'il prétend être. L'authentification des informations peut poser des problèmes particuliers, notamment dans l'attaque man in the middle (MITM). La plupart des protocoles cryptographiques incluent une certaine forme d'authentification du point de terminaison spécifiquement pour empêcher les attaques MITM. Par exemple, TLS (Transport Layer Security) et son prédécesseur, SSL (Secure Sockets Layer), sont des protocoles cryptographiques qui assurent la sécurité des communications sur des réseaux comme internet. TLS et SSL chiffrent les segments des connexions réseau au niveau de la couche transport. Plusieurs versions des protocoles sont largement utilisées dans des applications comme la navigation web, le courrier électronique, la télécopie Internet, la messagerie instantanée et la voix sur IP (VoIP). On peut utiliser SSL ou TLS pour authentifier le serveur en utilisant une certification de confiance mutuelle.

  Des techniques de hachage comme SHA-256 [40] et le mécanisme Kerberos basé sur le ticket d'octroi ou le ticket de service peuvent également être implémenté pour réaliser l'authentification.
- **Cryptage :** le cryptage des données est un moyen efficace d'empêcher tout accès non autorisé des données sensibles. Ses solutions protègent et conservent la propriété des données tout au long du cycle de vie, du centre de données au point final (y compris les appareils mobiles utilisés par les médecins, et les administrateurs) et dans le cloud. Le cryptage est utile pour éviter l'exposition à des violations telles que le reniflage de paquets (sniffing) et le vol des périphériques de stockage. Les organisations ou prestataires de soins de santé doivent s'assurer que le schéma de cryptage soit efficace, facile à utiliser par les patients et les professionnels de la santé, et facilement extensible pour inclure de nouveaux dossiers de santé électroniques. Bien que divers algorithmes de chiffrement aient été développés et déployés efficacement (RSA (Rivest, Shamir, and Adleman) et AES

(Advanced Encryption Standard), etc…), la sélection appropriée d'algorithmes de chiffrement appropriés pour appliquer un stockage sécurisé reste un problème difficile.

- **Masquage des données** : le masquage remplace les éléments de données sensibles par une valeur non identifiable. Ce n'est pas vraiment une technique de cryptage donc la valeur d'origine ne peut pas être retournée à partir de la valeur masquée. Il utilise une stratégie de dépersonnalisation des ensembles de données ou de masquage des identifiants des données personnelles tels que le nom, le numéro de sécurité sociale et de suppression ou généralisation des quasi-identifiants comme la date de naissance et les codes postaux. Ainsi, le masquage des données est l'une des approches les plus populaires de l'anonymisation des données en direct.
- **Contrôle d'accès** : une fois authentifiés, les utilisateurs peuvent entrer dans un système d'information mais leur accès sera toujours régi par une politique de contrôle d'accès qui est généralement basée sur des privilèges et des droits de chaque praticien autorisé par le patient ou un tiers de confiance. Il s'agit alors d'un mécanisme puissant et flexible pour accorder des autorisations aux utilisateurs. Il offre des contrôles d'autorisation sophistiqués pour garantir que les utilisateurs ne peuvent effectuer que les activités pour lesquels ils disposent d'autorisations. Un certain nombre de solutions ont été proposées pour aborder la sécurité et les préoccupations du contrôle d'accès : contrôle d'accès basé sur les rôles (RBAC) et contrôle d'accès basé sur les attributs (ABAC), ce sont les modèles les plus populaires pour les DSE.
- **Surveillance et audit** : la surveillance de la sécurité rassemble et étudie les événements du réseau pour attraper les intrusions. L'audit signifie l'enregistrement des activités des utilisateurs des systèmes de soins de santé par ordre chronologique, comme la tenue d'un journal de chaque accès et modification des données. Pour mesurer et assurer la sécurité dans un système de santé, il y a deux choses à prendre en considération : les procédures de détection et les procédures de prévention des intrusions sur l'ensemble du trafic réseau.

I.5 Conclusion

Dans ce premier chapitre nous avons passé en revue l'internet des objets ainsi que les systèmes de santé, nous avons vu les opportunités potentielles que l'IoT peut offrir dans le domaine de la santé (ex. stimuler la recherche en santé, la découverte de connaissances, l'accès à distance aux données biométrique du patient, l'autogestion de la santé). Malheureusement, plusieurs obstacles entravent son véritable potentiel, notamment les problèmes de confidentialité et de sécurité. La sécurité des données médicales du patient ainsi que sa vie privée sont considérées comme un énorme obstacle

pour les chercheurs dans ce domaine. Dans le deuxième chapitre, nous allons voir l'intégration d'une nouvelle technologie appelé blockchain dans le domaine de la santé et comment cette technologie peut participer à améliorer la protection de la vie privée du patient.

Chapitre II : Usage de la Blockchain dans les systèmes de soins de santé

II.1 Introduction

Depuis l'introduction de la blockchain (BC) via bitcoin, les recherches se sont multipliées pour étendre son application à des cas d'utilisation non financier. La technologie BC permet un environnement décentralisé et distribué sans besoin d'une autorité centrale. Les transactions sont à la fois sécurisées et fiables en raison de l'utilisation de principes cryptographiques. Ces dernières années, cette technologie est devenue très tendance et a pénétré différents domaines notamment celui de la santé où son potentiel est énorme.

Peut-être, l'avantage le plus évident et le plus remarquable de la blockchain est le fait qu'elle supprime le besoin d'un tiers de confiance centralisé dans les applications distribuées. En effet, la BC permet à deux ou d'avantage de parties d'effectuer des transactions dans un environnement distribué sans autorité centralisée. Elle surmonte le problème du point de défaillance unique qu'une autorité centrale aurait autrement présenté. Elle améliore également la vitesse de transaction, en supprimant le retard introduit par l'autorité centrale et, en même temps, elle rend les transactions moins chères, car les frais de transaction facturés par l'autorité centrale sont supprimés. Au lieu d'une autorité centrale, la blockchain utilise un mécanisme de consensus pour réconcilier les écarts entre les nœuds dans une application distribuée. L'intégration de la BC dans les soins de santé favoriserait alors l'automatisation des processus tout en réduisant les intermédiaires, permettrait aux acteurs de santé de savoir comment leurs données sont utilisées, protégerait les données des soins de santé contre la perte potentielle, la corruption ou les attaques de sécurité et améliorait la sécurité et la confidentialité des données des patients.

Ce chapitre est organisé comme suit : dans la section 2, nous définissons la blockchain, et nous présentons la technologie du registre distribué, les protocoles de consensus, le processus de chaînage des blocs dans la BC, les types de blockchains qui existent et la BC 3.0, ensuite, nous expliquons les notions de contrats intelligents et de jeton non fongible. Dans la section 3, nous soulignons les avantages de l'intégration de la BC dans les soins de santé, et nous citons ses différentes applications dans ce domaine, puis nous passons en revue ses limitations. Dans la section 4, nous définissons le contrôle d'accès ainsi que ses exigences dans un environnement distribué, puis nous présentons les modèles de contrôle d'accès qui existent, et nous finissons cette section par un état de l'art sur l'utilisation de la BC pour appliquer le contrôle d'accès. Nous concluons ce chapitre dans la section 5.

II.2 Technologie blockchain

II.2.1 Définition

Une blockchain peut être considérée comme un registre distribué contenant des enregistrements de transactions qui sont partagés sur l'ensemble de ses utilisateurs appelés nœuds. Un ensemble de transactions est scellé dans un bloc, chaque nouveau bloc contient une référence (une valeur de hachage) au contenu du bloc précédent formant ainsi une chaîne de blocs. Ces blocs sont scellés de manière sûre et immuable. La chaîne est en constante augmentation et de nouveaux blocs sont ajoutés à la fin [44].

II.2.2 Technologie du registre distribué

Pour mieux comprendre la notion du registre distribué, il faut d'abord comprendre ce qu'est un système décentralisé. La Figure II.1 représente la différence entre un système centralisé et un système décentralisé. Nous remarquons que dans le système centralisé, il existe plusieurs registres, mais tous les enregistrements sont conservés en un seul endroit central, dans le cas des soins de santé cela pourrait être l'organisation régionale d'information sur la santé (ORIS) par exemple. En substance, l'ORIS maintient l'état du registre. En cas de désaccord entre deux nœuds sur « l'état réel » du registre, l'ORIS est consulté comme arbitre final pour déterminer « l'état réel » du grand registre. Contrairement au système décentralisé, où il n'y a pas qu'un seul grand registre, mais tous les nœuds ont une copie et un certain niveau d'accès à son contenu. Pour maintenir l'intégrité du grand registre, les nœuds doivent avoir un moyen de convenir de « l'état réel » de ce dernier, en l'absence d'une autorité centrale. Lorsque les nœuds se mettent d'accord sur un état du grand registre, on parle de consensus [45].

Les participants d'un réseau BC sont représentés comme des nœuds et chaque participant possède une paire de clés publiques et privées. La clé publique sert d'adresse publique à l'utilisateur tandis que la clé privée est utilisée pour l'authentifier. Lorsqu'une transaction est créée, elle doit inclure la clé publique de l'utilisateur qui l'a créé, la clé publique du destinataire de la transaction et le message de transaction. Tous ces éléments sont regroupés et signés cryptographiquement à l'aide de la clé privée de l'utilisateur et ensuite diffusés aux autres nœuds du réseau blockchain, ainsi, le mécanisme de chiffrement à clé publique est utilisé pour garantir la cohérence. La signature numérique d'une transaction à l'aide de la clé privée est utilisée pour permettre l'authentification et assurer l'intégrité d'une transaction, car seule la bonne clé privée peut déchiffrer les messages chiffrés avec la clé publique correspondante. Ce concept est connu sous le nom de cryptographie asymétrique. Lorsque cela est fait, l'utilisateur aura proposé une transaction [46].

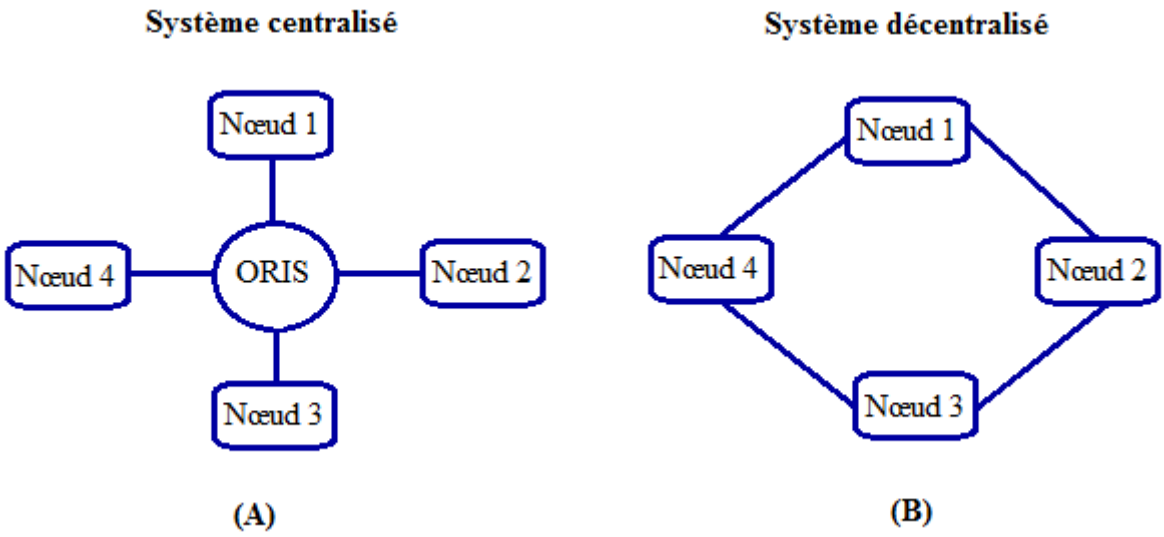

Figure II.1. Système centralisé vs système décentralisé [45].

Un bloc est un ensemble de propositions de transactions valides reçues dans un délai donné, par exemple 10 min dans le Bitcoin. Une proposition de transaction valide est une proposition qui satisfait aux exigences de validation. Le processus de validation garantit que la transaction proposée soit légitime, par exemple, qu'elle provient d'un utilisateur autorisé (nœud). L'algorithme de consensus détermine l'ordre dans lequel les blocs validés sont ajoutés au registre. Il existe des nœuds spéciaux dans un réseau BC qui sont responsables de l'exécution des algorithmes de consensus (c'est-à-dire de la validation des transactions et de la détermination de l'ordre dans lequel les blocs de transaction sont ajoutés à la blockchain). Ces nœuds spéciaux sont appelés mineurs (miners) et le processus de validation des transactions dans la blockchain est appelé exploitation minière (mining). Une fois qu'une proposition de transaction est reçue par un mineur, ce dernier procède à la vérification (si la transaction est valide). Les transactions validées sont incluses dans un bloc. Après une période de temps, le nouveau bloc de transactions validées est lié (ou enchaîné) aux blocs précédents, créant ainsi une chaîne de blocs, connue sous le nom de blockchain. La blockchain est répliquée sur tous les nœuds du réseau, de sorte que chaque nœud possède une copie identique de toutes les transactions du réseau [45, 47].

Un utilisateur souhaitant interagir avec la BC se connecte au réseau de la BC via un nœud. Chaque nœud de la BC a comme tâche de base [48, 49] :

- Connexion au réseau blockchain ;
- Stockage d'un registre à jour ;
- Écoute des transactions ;
- Acceptation ou refus d'une transaction ;

- Transmission des transactions valides au réseau ;
- Écoute des blocs nouvellement scellés ;
- Validation des nouveaux blocs scellés.

II.2.3 Protocole de consensus distribué

Pour que le réseau BC reste fonctionnel, ses pairs doivent se mettre d'accord sur un certain état du registre distribué et sur un moyen de regrouper les données en blocs. Un tel accord est appelé un protocole de consensus. Il assure que la majorité des pairs du réseau blockchain s'accorde sur l'état précis du registre partagé, et donc l'ordre dans lequel de nouveaux blocs sont ajoutés à ce dernier. Les protocoles de consensus distribué les plus utilisé sont les suivants [46, 48] :

- Preuve de travail (Proof of work : PoW).
- Preuve de d'enjeu (Proof of Stake : PoS).
- Preuve d'enjeu déléguée (Distributed PoS : DPoS).

Mais il en existe de nombreux autres protocoles de consensus, tel que :

- Preuve d'importance (Proof of Importance : PoI).
- Preuve d'activité (Proof of Activity : PoA).
- Preuve de brûlure (Proof of Burn : PoB).
- Preuve de dépôt (Proof of Deposit : PoD).

Un protocole de consensus distribué définit aussi la façon dont un réseau détermine quel pair préparera et scellera le bloc le plus récent avec des données non confirmées et non formatées. Le moyen le plus simple est de le déterminer au hasard, mais une telle approche n'est pas efficace en termes de longévité du réseau et peut même être dangereuse pour le réseau, car les pairs pourraient décider d'attaquer l'ensemble du réseau. L'idée derrière les protocoles de consensus est que le nœud choisi (mineur) apporte quelque chose de précieux et que le meilleur nœud soit récompensé. La récompense favorise la compétition et une compétition où les adversaires vérifient le travail et les objets de valeur des autres réduit également les chances d'une éventuelle attaque, dans ce qui suit nous allons expliquer les consensus les plus utilisés.

Le protocole de consensus PoW est utilisé dans le réseau Bitcoin. Il utilise la puissance de calcul comme mécanisme pour déterminer l'homologue choisi. La concurrence entre les pairs est basée sur le hachage des transactions non confirmées. Par conséquent, la chance qu'a un pair d'être choisi est proportionnellement liée à sa puissance de calcul. Chaque fois qu'un pair gagne et est

choisi, il obtient une récompense. La récompense actuelle dans le réseau Bitcoin est composée de 6,5 bitcoins nouvellement générés [49], qui sont ajoutés au compte du pair choisi. Le minage est basé sur le calcul d'un bloc, contenant des transactions validées, un nombre aléatoire appelé nonce et une référence de hachage au bloc précédent. Il est requis que le résultat du hachage soit égal à une valeur prédéfinie. Si un mineur atteint la valeur requise, il diffuse le bloc nouvellement généré sur le réseau. D'autres pairs le valident ensuite et s'il est correct, il est répliqué dans le réseau.

Le protocole de consensus PoS est basé sur les actifs qu'un pair possède (c.-à-d. la valeur de l'état du réseau qu'un pair a sous contrôle). La chance qu'a un pair d'être choisi pour confirmer un nouveau bloc est proportionnellement liée à ses actifs (sa richesse). Dans la pratique, cela est réalisé en ayant un pair déposer un nombre prédéfini de ses actifs, cela achète au nœud un ticket. Le gagnant est choisi de façon déterministe, d'une façon pseudo-aléatoire à partir d'un groupe de pairs avec des billets. La concurrence dans ce cas n'est pas basée sur la puissance de calcul des pairs, ce qui signifie qu'il y a une consommation d'énergie minimale en comparaison au PoW. Cependant, une telle approche est similaire à une société actionnaire, où les riches ont un avantage. Cela fonctionne, car il est peu probable qu'un pair attaque le réseau, car dans ce cas, il attaquerait ses propres actifs. Il existe plusieurs versions du protocole de consensus PoS, où chacun introduit une approche différente sur la façon de choisir le validateur afin de garantir justice. L'une de ces versions est le DPoS. Alors que le consensus de PoS autorise chaque membre du réseau à valider les blocs à condition d'avoir un montant minimum de crypto-monnaies, celui du DPoS met en place un système de vote dans lequel les utilisateurs de la plateforme devront voter pour des représentants chargés de valider les blocs à leur place [50].

II.2.4 Processus de chaînage des blocs

Le chaînage des blocs est atteint grâce à une autre primitive cryptographique qui implique l'utilisation de fonctions de hachage (ex : SHA256). La fonction de hachage prend un message de longueur arbitraire et le croque en une sortie de hachage de longueur fixe, appelée un résumé de message ou une empreinte numérique. Une propriété intéressante de la fonction de hachage est qu'elle est résistante aux collisions, c'est-à-dire qu'il n'y a pas deux messages différents qui produiront la même sortie de hachage. Cette propriété est la base du chaînage de blocs. Pour chaîner un nouveau bloc à la blockchain, le hachage du bloc précédent est inclus dans l'en-tête du nouveau bloc. Ainsi, le dernier bloc de la blockchain contient l'empreinte de transactions du bloc précédent, qui à son tour contient l'empreinte de transactions du bloc précédent et ainsi de suite comme illustré dans la Figure II.2. Si l'une des transactions d'un bloc est modifiée, même si légèrement, la sortie de hachage correspondante changera radicalement, ce qui cassera la chaîne au bloc suivant. Par

conséquent, toute modification du contenu d'un bloc dans la blockchain est facilement détectée dans le réseau. Pour cette raison, une fois qu'une transaction est ajoutée à un bloc et enchaîné à la BC, cette transaction ne peut pas être modifiée ou annulée. Ainsi, l'information sur la BC serait immuable. L'immuabilité est une propriété importante de la BC, qui garantit que les enregistrements, une fois créés, ne peuvent pas être supprimés ou modifiés. Pour mettre à jour un enregistrement sur la blockchain, un nouvel enregistrement doit être créé. Le processus de chaînage des blocs à la blockchain garantit également que les transactions soient horodatées, créant ainsi une piste d'audit de qui a fait quoi et quand [45, 51].

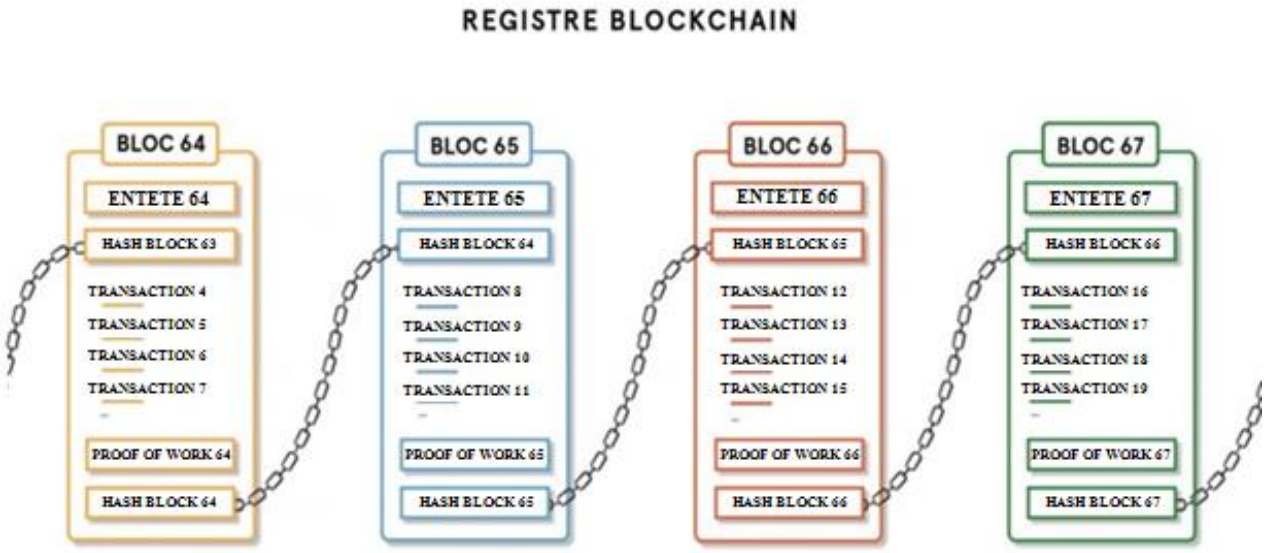

Figure II.2. Chaînage des blocs [45].

II.2.5 Type de blockchain

Généralement, il existe différents types de BC en fonction des données gérées, de la disponibilité de ces données, et sur quelles actions peuvent être effectuées par l'utilisateur. Ceux-ci inclus les blockchains : Publique (sans autorisation), Consortium (autorisé par le public) et Privé (autorisé) [44-46].

Dans certaines implémentations de la BC, par exemple, Bitcoin, tout nœud est libre de rejoindre le réseau et de participer au processus de minage sans aucune autorisation. Ce type d'implémentation est appelée BC publique ou sans autorisation. Cependant, certaines parties de la BC pourraient être cryptées afin de préserver l'anonymat d'un participant. En revanche, la BC autorisée est celle dans laquelle les participants doivent être autorisés et avoir les bonnes autorisations d'accès avant de pouvoir rejoindre et participer au réseau. Dans la BC autorisée, seuls certains nœuds peuvent être autorisés à participer au processus minier. En raison de leurs

caractéristiques, les réseaux BC autorisés sont plus susceptibles d'être plus petits, plus rapides et plus sûrs que les réseaux publics. Une BC autorisée peut en outre être classée comme une blockchain privée ou un consortium. La BC de type consortium ne permet qu'à un groupe sélectionné de nœuds de participer au processus de consensus distribué. Il peut être utilisé dans une ou plusieurs industries. Quand une BC consortium est établie au sein d'une industrie (par exemple, le secteur financier), elle est ouverte à usage public limité et partiellement centralisé. D'autre part, un consortium entre les industries (ex : compagnies d'assurance, institutions financières, institutions gouvernementales) est ouvert au public tout en ayant établi une fiducie partiellement centralisée.

Une BC privée permet uniquement aux nœuds choisis de rejoindre le réseau. C'est donc un réseau centralisé. Les BC privées sont des réseaux autorisés qui choisissent quels nœuds peuvent effectuer des transactions, exécuter des contrats intelligents ou agir en tant que mineurs. Ils sont gérés par une organisation qui est la partie de confiance. Tissu Hyperledger et Ripple sont des exemples de plateformes de BC qui ne prennent en charge que les réseaux de BC privés.

Il est à noter que de telles classifications font encore l'objet de débats et que différentes définitions peuvent être trouvées dans la littérature.

Une distinction entre les BC peut également être établie en fonction de leurs objectifs :

- Pour le suivi des actifs numériques (par exemple, Bitcoin).
- Pour exécuter certaines logiques (comme les contrats intelligents dans Ethereum).
- Certaines BC utilisent des jetons (par exemple, Ripple, Bitcoin, Ethereum), tandis que d'autres n'en utilisent pas (par exemple, HyperledgerFabric).

II.2.6 Contrats intelligents

Les contrats intelligents (Smart Contract ou SC) sont essentiellement des programmes stockés sur une BC qui s'exécutent lorsque des conditions prédéterminées sont remplies. Les conditions sont écrites suivant de simples instructions si …alors… Ils sont généralement utilisés pour automatiser l'exécution d'un arrangement ou d'actions afin de permettre aux participants d'être sûrs des résultats, sans intervention d'un tiers ni perte de temps. Ces actions peuvent inclure l'automatisation d'un flux de travail, un système de vote, l'enregistrement d'un véhicule, etc… Lorsque les actions sont exécutées (terminées), une transaction est ajoutée à la BC, ainsi cette dernière ne pourra pas être modifiée et seules les parties qui ont reçu le droit accordé pourront accéder aux résultats [44, 52].

II.2.7 Jetons non fongibles

Un jeton non fongible (Non-Fungible Token ou NFT) est un actif cryptographique numérique sur une BC avec des codes d'identification uniques et des métadonnées qui les distinguent des autres. Un jeton fongible est échangeable contre un autre tant que la valeur de chaque jeton reste la même. Les jetons fongibles comme les crypto-monnaies sont donc similaires à la monnaie fiduciaire, c'est-à-dire tout comme chaque billet ou pièce détient une certaine valeur, chaque jeton fongible peut également être interchangé. En revanche, chaque NFT ne peut pas être échangé car il est unique et ne peut être utilisé que pour enregistrer des informations uniques relatives à un actif numérique [53].

II.2.8 Blockchain 3.0

La BC a gagné en popularité en tant que technologie de registre distribué à la suite de l'article Bitcoin publié en octobre 2008 [54]. En tant que technologie sous-jacente pour Bitcoin, le principal utilitaire de la BC est qu'elle rend possible l'échange de pièces électroniques entre les participants dans un réseau sans avoir besoin d'un tiers centralisé de confiance. Les transactions impliquant l'échange des monnaies entre personnes physiques ou morales reposent traditionnellement sur un tiers de confiance (TDC), comme une banque, en tant que médiateur. Le recours à un TDC n'est pas souhaitable pour un certain nombre de raisons. Un tiers de confiance peut mal fonctionner, échouer ou être compromis de manière malveillante pour rendre le système financier indisponible ou précaire; ainsi, un TDC sape un système potentiellement comme un seul point d'échec. Un TDC facture également des frais de transaction et ajoute des retards de transaction. La motivation derrière Bitcoins est, par conséquent, de surmonter ces limitations associées à la dépendance au TDC dans les transactions électroniques.

Un an après la publication du célèbre article blanc sur Bitcoin, la crypto-monnaie Bitcoin a été implémentée avec le code publié en open-source, ce qui a permis à d'autres de modifier le code et de l'améliorer pour créer différentes générations de technologies basées sur la BC. Les premières implémentations de crypto-monnaies basées sur la BC, telles que le Bitcoin, constituent la première génération de la technologie blockchain, également appelée blockchain 1.0. Les autres technologies blockchain1.0 incluent Monero, Dash et Litecoin.

La deuxième génération de la technologie BC (blockchain 2.0) est associée à l'introduction de propriétés intelligentes et de contrats intelligents. Les propriétés intelligentes sont les propriétés ou actifs numériques dont la propriété peut être contrôlée par une plate-forme basée sur la blockchain, tandis que les contrats intelligents sont des programmes qui codent les règles de contrôle et de

gestion des propriétés intelligentes. Ethereum, EthereumClassic, NEO et QTUM sont des exemples de BC 2.0.

Sur la base de ce qui précède, la troisième génération de la technologie BC (blockchain 3.0) est désormais concernée par les applications non financières. À cette fin, des efforts ont été faits pour adapter la technologie à d'autres domaines d'application, en dehors de la finance, afin que d'autres secteurs et cas d'utilisation puissent bénéficier de ses fonctionnalités intéressantes. Par conséquent, la BC est désormais considérée comme une technologie à usage général, qui a trouvé des applications dans différentes industries et différents cas d'utilisation, telles que la gestion de l'identité, la résolution des litiges, la gestion des contrats, la gestion des chaînes d'approvisionnement, l'assurance et les soins de santé [55, 56].

II.3 Blockchain dans les soins de santé

II.3.1 Avantages de la blockchain dans les soins de santé

Une caractéristique importante de la BC qui est clairement bénéfique pour les applications de soins de santé est la décentralisation qui permet de mettre en œuvre des applications de soins de santé distribuées qui ne s'appuient pas sur une autorité centralisée. De plus, le fait que les informations de la BC soient répliquées entre tous les nœuds du réseau crée une atmosphère de transparence et d'ouverture, permettant aux acteurs de santé, et notamment les patients, de savoir comment leurs données sont utilisées, par qui, quand et comment. Plus important encore, compromettre un nœud dans le réseau ne signifie pas affecter l'état du grand registre puisque ces informations sont répliquées sur plusieurs nœuds dans le réseau. Par conséquent, la BC peut protéger les données des soins de santé contre la perte potentielle de données, la corruption ou les attaques de sécurité, telles que l'attaque par ransomware. De plus, la propriété d'immuabilité de la BC qui rend impossible d'altérer ou de modifier tout enregistrement qui a été ajouté au registre correspond très bien aux exigences du stockage des dossiers de santé (il est très important de garantir l'intégrité et la validité des DSE). De plus, l'utilisation d'algorithmes cryptographiques pour crypter les données stockées sur la BC garantit que seuls les utilisateurs disposant d'autorisations légitimes pour accéder aux données peuvent les décrypter, améliorant ainsi la sécurité et la confidentialité des données. De plus, l'identité des patients dans la BC est pseudonymisée par l'utilisation de clés cryptographiques, les données de santé des patients peuvent être partagées entre les acteurs de la santé sans révéler l'identité des patients. La BC prend également en charge les contrats intelligents, qui peuvent être utilisés pour programmer les règles qui permettent aux patients de contrôler la façon dont leurs DSE sont partagés ou utilisés. Ceci est particulièrement pertinent pour le règlement général sur la

protection des données (RGPD) qui interdit le traitement des données personnelles sensibles des patients, sauf accord explicite ou conditions spécifiques. Par conséquent, la BC peut faciliter le développement d'un système de gestion des dossiers médicaux partagés conforme au RGPD, en encodant dans le contrat intelligent un ensemble de règles qui garantissent que les données sensibles des patients ne puissent pas être partagées ou utilisées sans autorisations appropriées [45, 55].

Les avantages potentiels de la BC pour les applications de santé sont résumés dans le Tableau II.1

Décentralisation	La nature même des soins de santé dans laquelle il existe des parties prenantes réparties, nécessite une épine dorsale de gestion des données de santé décentralisée à partir de laquelle toutes les parties prenantes peuvent avoir un accès contrôlé aux mêmes dossiers de santé, sans que personne ne joue le rôle d'une autorité centrale sur les données de santé mondiales.
Amélioration de la protection des données et de la vie privée	La propriété d'immuabilité de la BC améliore considérablement la sécurité des données de santé stockées dedans, car les données, une fois enregistrées, ne peuvent pas être corrompues, modifiées ou supprimées. Toutes les données de santé sur la BC sont cryptées, horodatées et ajoutées dans un ordre chronologique. De plus, ils sont enregistrés en utilisant des clés cryptographiques ce qui contribuent à protéger l'identité ou la vie privée des patients.
Propriété des données de santé	Les patients doivent posséder leurs données et contrôler la façon dont leurs données sont utilisées. Les patients ont besoin de l'assurance que leurs données de santé ne sont pas utilisées à mauvais escient par d'autres parties prenantes et avoir un moyen de détecter quand une telle utilisation abusive se produit. La BC aide à répondre à ces exigences via des protocoles cryptographiques solides et des contrats intelligents bien définis.
Robustesse/ disponibilité	Étant donné que les enregistrements sur la BC sont répliqués dans plusieurs nœuds, la disponibilité des données de santé stockées est garantie car le système est robuste et résilient contre les pertes de données, la corruption de données et certaines attaques de sécurité contre la disponibilité des données.

Transparence et confiance	La BC, grâce à sa nature ouverte et transparente, crée une atmosphère de confiance autour des applications de santé distribuées. Cela facilite l'acceptation de telles applications par les acteurs de la santé.
Vérifiabilité des données (audit)	Même sans accéder au texte en clair des enregistrements stockés sur la BC, l'intégrité et la validité de ces enregistrements peuvent être vérifiées. Cette fonctionnalité est très utile dans le domaine des soins de santé où l'audit des dossiers est une exigence

Tableau II.1 Avantages de la blockchain pour les applications de santé [45].

II.3.2 Applications de la blockchain dans le domaine de la santé

La BC en tant que technologie décentralisée et distribuée a d'énormes applications dans le domaine de la santé. La technologie blockchain principalement introduite pour la banque et la finance, vise désormais à fournir des moyens plus sûrs de partager les données entre les prestataires et les patients dans les systèmes de soins de santé. En raison d'un potentiel immense et d'une applicabilité réaliste de cette technologie dans les systèmes de santé, les chercheurs ont commencé par explorer ces cas d'utilisation potentiels dans les soins de santé, quelques-uns sont cités ci-dessous [46, 57] :

A. Partage de données cliniques

Une application cruciale et clé de la BC dans les soins de santé est le partage de données médicales entre diverses entités du système. Les DSE contiennent des informations extrêmement critiques et sensibles, des informations médicales relatives au patient qui doivent être stockées, partagées, traitées et accessibles en toute sécurité. Ainsi, afin d'améliorer et de renforcer la qualité des services de santé, les informations médicales doivent être stockées et partagées fréquemment parmi divers participants concernés tels que les patients, les médecins, les prestataires de soins de santé, les pharmacies, les compagnies d'assurance et les chercheurs, entre autres. En général, ce type de partage de données critiques nécessite des mesures strictes de transparence et de responsabilité lors des transactions de données. Par conséquent, la BC ajoute plus de transparence dans de tels cas car elle maintient un registre distribué entre toutes les entités impliquées dans le réseau, elle fournit un moyen fiable et sécurisé de partage de données et de mécanismes de gestion où toutes les parties sont au courant des transactions.

B. Partage des données à l'échelle mondiale

Il y a aussi certaines occasions où les patients voyagent en dehors de leur propre pays à des fins touristiques ou pour autres raisons. Il existe également des situations qui peuvent nécessiter de consulter un médecin pour le traitement de toute maladie. Dans ce cas, afin de fournir de meilleurs services de santé, les médecins / les hôpitaux de l'autre pays devraient avoir connaissance des informations de santé du patient. Grâce à la BC, les informations médicales peuvent être facilement partagées avec les entités requises résidantes dans l'autre pays avec le consentement du patient en ayant le contrôle sur ses données. Afin de recevoir un meilleur traitement médical hors du pays, les antécédents médicaux du patient respectif doivent être connus par exemple, si le patient est concerné par une quelconque allergie à certains médicaments ou la connaissance de son traitement récent. Ainsi, ce genre de données doit être accessible en toute sécurité par le fournisseur de services.

C. Maintien des antécédents médicaux

La technologie de la BC peut également être utilisée pour stocker et maintenir les antécédents médicaux des patients. Ces derniers, peuvent par exemple visiter des hôpitaux déconnectés et la chaîne globale des antécédents médicaux peut ne pas être disponible ou bien entretenue. Afin de surmonter ces problèmes, la BC peut être utile pour maintenir l'historique des enregistrements pour chaque visite à n'importe quel hôpital. De plus, en raison de l'indisponibilité de quelques données déconnectées liées à la médecine comme les rapports de laboratoire, les patients doivent à nouveau répéter les mêmes tests. Cela augmenterait non seulement le coût de répétition du même test, mais il peut également être risqué de faire un test avec des radiations élevées encore et encore.

D. Recherches et essaies cliniques

Les essais cliniques représentent un autre processus clé et précieux dans le secteur de la santé qui nécessite une surveillance appropriée à chaque étape du parcours. Les étapes d'essai consomment des tas de ressources que plusieurs parties doivent coordonner avec d'autre. Ces phases nécessitent également une confiance massive des différentes entités impliquées. Par conséquent, la BC peut être un outil essentiel pour faire face à ces essais et recherches où chaque phase peut être correctement tracée et les données peuvent être gérées et analysées sans beaucoup de gaspillage de ressources.

E. Contrôle d'accès aux données de santé

Avec les récents progrès technologiques, l'emprise des utilisateurs sur leurs propres données s'affaiblit. En particulier les données des soins de santé, les utilisateurs ne savent généralement pas quelles entités accèdent à leurs données médicales et à quelles fins et si elles sont autorisées à y accéder. Par exemple, dans le cas des DSE, divers prestataires de soins de santé sont associés et les patients ne sont pas pleinement conscients des parties qui accèdent, stockent et partagent leurs

données médicales. La technologie BC peut permettre aux patients non seulement d'accéder à leurs informations médicales d'une manière plus sûre, mais garantie également que seules les entités autorisées peuvent y accéder.

F. Gestion de la chaîne d'approvisionnement des médicaments

La gestion de l'approvisionnement en médicaments est cruciale à l'industrie de la médecine, mais elle souffre encore de diverses complexités et de pertes dues aux contrefacteurs et aux pillages. La BC peut garder la traçabilité de ces opérations de la chaîne d'approvisionnement et améliorer l'intégrité de l'ensemble du processus, elle sera utile pour vérifier l'authenticité des médicaments et de leur chaîne d'approvisionnement aux parties autorisées. Donc, la BC peut être vitale dans le suivi des différentes phases de la gestion de la chaîne d'approvisionnement en médicaments.

G. Facturation

Les modes traditionnels de facturation des patients sont considérés comme très complexes et sont exposés à des fraudes liées à la facturation. En plus de cela, le processus dans la plupart des cas prend plus de ressources et de temps pour recevoir toutes les factures requises (laboratoire, assurances, etc..). Les solutions de paiement basées sur la BC devraient rendre le processus de facturation beaucoup plus facile par rapport aux approches de facturation traditionnelles.

II.3.3 Limitations de la blockchain dans la santé

Certains défis identifiés pour le développement d'applications de soins de santé basées sur la BC incluent l'interopérabilité, la sécurité, la confidentialité, la scalabilité, la vitesse et l'engagement des patients, ainsi que les vulnérabilités spécifiques à la blockchain [45, 58, 59].

Le défi de l'interopérabilité provient du fait qu'il n'existe pas encore de norme pour développer des applications de soins de santé basées sur la BC ; par conséquent, les applications développées par différents fournisseurs ou sur différentes plates-formes peuvent ne pas être en mesure d'interopérer. Considérez, par exemple, deux applications de télésurveillance des patients, dont une est développée sur la plateforme Ethereum tandis que l'autre est développée sur la plate-forme Hyperledger Fabric, il serait difficile d'échanger des informations d'une plateforme à l'autre.

En ce qui concerne la sécurité et la confidentialité des applications de soins de santé basées sur la BC, il existe une crainte que malgré les techniques de cryptage employées, il est encore possible de révéler l'identité d'un patient dans une BC publique en reliant suffisamment de données associées à ce patient. En outre, il existe également un risque potentiel d'atteintes à la sécurité qui pourraient survenir des attaques malveillantes intentionnelles à la BC des soins de santé par des

organisations criminelles qui pourraient compromettre la vie privée des patients. Il y a eu plusieurs cas d'attaques signalées contre les réseaux de BC qui alimentent les différentes crypto-monnaies. Les clés privées utilisées pour le chiffrement et le déchiffrement des données dans la BC sont également sujettes à un potentiel compromis qui pourrait entraîner un accès non autorisé aux données de santé stockées. Les principes de gestion de ces clés représentent aussi un défi, ils ne sont pas réalisables pour la BC, une clé pour tous les blocs n'est pas sûre car si la clé est compromise, alors toutes les données seront divulguées. Cependant, une clé par bloc n'est pas pratique non plus car elle nécessite un coût élevé pour stocker et récupérer le nombre élevé de clés pour chaque bloc individuel qui a été créé.

En outre, il y a aussi le souci de la propriété d'immuabilité de la BC qui n'augure pas bien avec le « droit à l'oubli » du RGPD, qui stipule que l'utilisateur a le droit de demander l'effacement complet de ses données. Étant donné que l'immuabilité de la BC garantit que les données une fois enregistrées ne peuvent plus être supprimée ou modifiée, elle pourrait s'avérer contre-productive lorsqu'il est souhaitable d'effacer complètement les antécédents médicaux d'un patient.

La scalabilité des solutions de santé basées sur la BC est un défi majeur quand il s'agit de volume de données, il n'est pas optimal, ni même réalisable dans certains cas, de stocker des données biomédicales à haut volume sur la BC car cela est susceptible d'entraîner une grave dégradation des performances, ou quand le nombre de participants au système augmente, en augmentant également les exigences de calcul de l'ensemble de l'infrastructure BC. Cela devient un problème de plus en plus difficile s'il existe un grand nombre d'appareils intelligents ou de capteurs dont la capacité de calcul est inférieure à un ordinateur moyen. Le compromis entre les capacités informatiques disponibles par rapport au montant des transactions médicales pourrait limiter l'évolutivité de ces systèmes de santé.

Il y a aussi le problème de la vitesse car le traitement basé sur la BC peut introduire une latence importante. Par exemple, le mécanisme de validation dans la configuration actuelle de la BC Ethereum nécessite que tous les nœuds d'un réseau participent au processus de validation. Cela entraîne un délai de traitement considérable, surtout si la charge de données est importante.

Dans un environnement IoT, un certain nombre de solutions spécifiques à la BC IoT sont coûteux en calcul et impliquent une surcharge de bande passante élevée qui entraîne des retards de données et une puissance de traitement importante. De telles exigences ne sont pas pratiques pour la plupart des appareils IoT car ce sont généralement des capteurs qui sont soumis à des contraintes de calcul. En d'autres termes, ces appareils peuvent ne pas avoir la puissance de calcul requise pour

utiliser les capacités de la BC, tout en remplissant leur objectif d'origine. Ce ralentissement de la vitesse de traitement peut entraîner les périphériques à fonctionner de manière sous-optimale ou potentiellement même surcharger l'appareil le rendant même incapable d'exécuter le logiciel ou le programme BC d'origine en même temps.

Un autre défi est de savoir comment impliquer les patients dans la gestion de leurs données sur la BC. Les patients, en particulier les personnes âgées et les jeunes, peuvent ne pas être intéressés ou capables de participer à la gestion de leurs données de santé. La société actuelle est généralement habituée à des processus de soins de santé auxquels on accède soit par le biais de procédures basées sur les documents ou, dans certains cas, par des moyens en ligne tels que les DSE et autres services de santé en ligne. Dans l'heure actuelle, les données des patients ne sont pas si communément partagées avec plusieurs parties. Ce changement culturel sera donc l'un des défis majeurs, changer le comportement des gens vers le partage de données de manière distribuée nécessitera certains efforts.

La technologie BC a également quelques vulnérabilités spécifiques qui sont uniques à la mise en œuvre et à l'architecture du système. Les vulnérabilités spécifiques de la BC incluent les attaques avec retenue, les attaques 51%, les attaques à double dépense, les attaques minières égoïstes, les attaques par éliminations de blocs, les attaques de pointes et les problèmes d'anonymat dans la BC.

Une attaque avec retenue (block with holding attack) a lieu lorsque des acteurs malveillants minent avec succès des blocs, mais ne soumettent pas ces blocs dans le système. Au lieu de cela, le mineur ne soumet que des parties du bloc qui ne sont pas les solutions complètes requises par le système. Ces parties ne sont qu'une partie de la solution et n'entraînent pas l'exploitation minière du bloc et diminuent les revenus du pool car chaque action réduit la capacité d'un autre mineur à soumettre une solution réussie. Un acteur en retenant les blocs valides limite la capacité pour d'autres de soumettre une solution complète, réduisant ainsi leurs revenus. Cependant, un tel acte augmente les récompenses de l'acteur malveillant puisque ce dernier peut soumettre autant de blocs partagés que possible à l'authentification gestionnaire du pool, qui ont chacun une partie de la solution.

Une attaque similaire à l’attaque avec retenue est l’attaque 51%, où un seul mineur a plus de ressources de calcul que le reste du réseau et domine la vérification et l'approbation des transactions sur la BC contrôlant le contenu de la BC. Une attaque à double dépenses (double spending attack) est le processus d'utilisation de la même crypto-monnaie numérique pour plus d'une transaction en reproduisant facilement les informations numériques pour la crypto-monnaie. Dans une attaque

minière égoïste (selfish mining), les mineurs égoïstes invalident intentionnellement le travail des mineurs honnêtes en publiant de manière stratégique la chaîne privée du pool minier à des états du pool pour influencer les récompenses. Le concept de l'exploitation minière égoïste est d'attirer d'autres mineurs pour continuer à travailler sur des blocs qui mènent à une impasse au lieu de les attacher à la plus longue chaîne. En n'exposant pas le bloc exploité à l'ensemble du réseau BC, un attaquant se donne une priorité dans l'exploitation du bloc suivant. Une attaque par élimination de bloc (block discarding attack) est le processus consistant à saisir les connexions réseau par rapport aux autres nœuds et utiliser cette connexion pour être informé des blocs minés avant le reste du réseau. Par la suite l'attaquant publie le bloc miné avant le mineur légitime et élimine son bloc. Une attaque de pointe difficile (difficult spike attack) est quand l'attaquant tire parti de la puissance de hachage du processus cryptographique d'extraction de blocs pour manipuler le niveau de difficulté.

Une autre vulnérabilité potentielle de la BC est son pseudo-anonymat, qui est le fait que toutes les transactions sont enregistrées en permanence dans le grand registre public et qui peuvent être vu par tout le monde. Les informations privées sont gardées secrètes jusqu'à ce qu'elles soient révélées dans certaines circonstances, qui permettrait à quiconque d'utiliser ces informations pour rechercher les transactions passées de cet utilisateur. Ces informations peuvent, cependant, être utiles dans une piste d'audit ou une enquête médico-légale. Une autre considération pour la BC est d'inciter correctement les mineurs. Si les mineurs ne reçoivent aucun avantage pour le calcul et les ressources qu'ils consacrent à faciliter la BC, alors ils ne peuvent pas faciliter les transactions ou la création de nouveaux blocs.

La BC est également vulnérable à certaines vulnérabilités logicielles générales qui permettent des attaques malveillantes. Ces attaques malveillantes peuvent ensuite être utilisées pour faciliter d'autres délits tels que l'usurpation d'identité et l'exfiltration de données. Le vol d'identité se produit sur la BC si un utilisateur de la blockchain a sa clé privée volée, ce qui permet à l'acteur malveillant d'avoir accès à tout ce que la victime a publié sur la BC. Les activités illégales telles que les armes à feu illégales, les drogues et autres objets interdits peuvent être vendues via le grand registre distribué dans un mode pseudo anonyme.

II.4 Contrôle d'accès

L'IoT dans les soins de santé a été proposé comme un moyen prometteur d'améliorer considérablement l'efficacité et la qualité des soins aux patients. Les dispositifs médicaux dans l'IoT des soins de santé mesurent les signes vitaux des patients et regroupent ces données dans des DSE qui sont téléchargés sur le Cloud pour le stockage et accessibles par les professionnels de la santé.

Étant donné que les DSE contiennent des informations physiologiques sensibles, et afin de protéger la vie privée des patients, le contrôle d'accès aux fichiers externalisés est essentiel pour interdire l'accès non autorisé aux données.

Les systèmes de contrôle d'accès centralisés - autrement appelés paradigme client / serveur - ont été conçus pour répondre aux besoins des scénarios internet orientés homme-machine traditionnels où les périphériques sont dans le même domaine de confiance, qui nécessite une gestion centralisée des accès. Cependant, des inconvénients importants surviennent lorsque des approches centralisées sont envisagées sur un vrai déploiement IoT. D'une part, l'inclusion d'une entité centrale pour chaque demande d'accès compromet clairement les propriétés de sécurité de bout en bout. D'autre part, la nature dynamique des scénarios IoT avec une énorme quantité potentielle d'appareils complique la gestion de la confiance avec l'entité centrale, ce qui affecte l'évolutivité. De plus, ils sont construits autour d'un seul serveur logique et de plusieurs clients. Par conséquent, le contrôle d'accès se fait souvent dans l'application côté serveur, une fois le client a été authentifié [60]. L'IoT renverse ce paradigme en ayant de nombreux périphériques et éventuellement de nombreux clients servant de serveurs. Ainsi, une question cruciale se pose : comment pouvons-nous réaliser un contrôle d'accès distribué et fiable dans l'IoT ? La réponse peut résider dans la technologie BC.

II.4.1 Définition du contrôle d'accès

Il existe plusieurs définitions du contrôle d’accès dans la littérature, nous en avons retenu les deux qui nous ont semblé importantes :

1. Traditionnellement, le contrôle d'accès est basé sur l'identité d'un utilisateur demandant l'exécution d’une capacité à effectuer une opération (par exemple, lire, écrire…) sur un objet (par exemple, un fichier) [61].

2. Le contrôle d’accès est un mécanisme puissant et flexible pour accorder des autorisations aux utilisateurs. Il offre des contrôles d'autorisation sophistiqués pour garantir que les utilisateurs ne puissent effectuer que les activités pour lesquels ils disposent d'autorisations [58].

II.4.2 Contrôle d’accès dans les systèmes basés sur le fog computing

Il est inévitable que le FC dans les systèmes de soins de santé recueille et traite profondément les informations personnelles. Par conséquent, sans sécurité et sans mécanismes de protection de la vie privée, il ne peut pas être adopté malgré son utilité. Le contrôle d'accès est particulièrement

important pour assurer la sécurité des données des patients. Dans le contrôle d'accès traditionnel, les utilisateurs stockent leurs données sur des serveurs de confiance comme le cloud. Ensuite, ces serveurs vérifient si l'utilisateur demandé a le privilège d'accéder aux données. Dans le FC, comme les utilisateurs et les serveurs se trouvent dans différents domaines de confiance, ce type de modèle est un handicap [62].

Les problèmes de contrôle d'accès dans le FC sont classés en trois types [63] :

- Si les utilisateurs souhaitent utiliser les services de stockage et de calcul, ils doivent être autorisés par le Fog ou le Cloud, et certaines politiques devraient être utilisées pour contrôler l'accès aux données et aux services.
- Le Fog et le Cloud ont besoin d'un contrôle d'accès réciproquement.
- Les machines virtuelles (VMs) ont besoin d'un mécanisme de contrôle d'accès pour éviter les attaques sur le canal latéral.

Par conséquent, le contrôle d'accès dans les systèmes de soins de santé basés sur le FC est un outil important pour préserver la confidentialité des patients et assurer la sécurité du système.

II.4.3 Exigences du contrôle d'accès dans un environnement distribué

Afin de construire un contrôle d'accès sécurisé et efficace dans un environnement distribué, les exigences suivantes doivent être prises en considération [62] :

- **Latence :** le temps d'exécution, le temps de déchargement d'une tâche et le temps de prise de décision peuvent entraîner une latence. Fournir à l'utilisateur final des services et des applications avec une faible latence garantie est essentiel dans les systèmes distribués et spécialement dans les systèmes de soins de santé. Les systèmes de contrôle d'accès doivent accorder les décisions d'accès dans un délai raisonnable.
- **Efficacité :** la plupart des dispositifs sont riches en ressources, mais certains sont limités par rapport à cette contrainte (par exemple, les appareils IoT). Cela peut conduire à un retard du processus de décision, ce qui entraînera une latence inacceptable à d'autres parties du réseau. Les systèmes de contrôle d'accès doivent être efficaces.
- **Agrégation :** les données sont collectées par les appareils des utilisateurs qui sont géo-distribués. Afin de réduire la latence, elles doivent être agrégées par les dispositifs fog les plus proches des utilisateurs. Le changement d'autorité des données avant et après l'agrégation doit être pris en considération.

- **Protection de la vie privée :** puisqu'il est inévitable d'échanger des données entre les différentes administrations des soins de santé, la protection de la confidentialité des données est une exigence critique dans le contrôle d'accès.
- **Gestion des politiques :** c'est un élément clé dans les systèmes distribués ainsi que dans les systèmes de soin de santé. En conséquence, le contrôle d'accès doit avoir la capacité de prendre en charge la création, l'invocation, et la suppression d'une politique.

II.4.4 Modèles de contrôle d'accès

Il existe différents modèles pour concevoir le contrôle d'accès. Cette sous-section décrit les modèles existants et évalue la possibilité de leurs applications dans un système de soins de santé distribué [64-66].

- **Matrices de contrôle d'accès (MCA)** : la MCA est le modèle le plus traditionnel et encore souvent utilisé. Il utilise simplement des listes d'usagers avec indication de toutes les permissions de contrôle d'accès pour chaque usager. Les permissions sont données et modifiées par les administrateurs du système, suivant la politique de l'organisation, mais ce modèle est difficile à gérer, car les usagers et les objets doivent être considérés individuellement, surtout dans les grandes organisations telles que les soins de santé qui ont des milliers d'usagers et d'objets, chaque fois qu'un patient change de médecin ou qu'un médecin change d'hôpital ou de poste, il faut s'assurer de lui enlever et ajouter individuellement toute une série de permissions.
- **Contrôle d'accès discrétionnaire (CAD) :** le CAD est une extension du modèle MCA. Dans un modèle CAD, le propriétaire des données a la possibilité de décider des autorisations d'accès pour les autres et de les définir en conséquence. Il est plus flexible et moins sécurisé, il est donc généralement utilisé dans des environnements qui mettent l'accent sur la commodité et ne nécessitent pas un niveau élevé de sécurité, tels que le système d'exploitation UNIX. Les modèles CAD sont généralement appliqués uniquement dans les applications d'héritage. Ce modèle entraînera une grande quantité de gestion dans un environnement distribué avec plusieurs applications et plusieurs utilisateurs.
- **Contrôle d'accès obligatoire (CAO) :** dans le CAO, les usagers ne sont pas propriétaires des objets, l'accès aux objets est sujet à des règles fixes que les usagers ne peuvent pas modifier. Il est conçu en fonction de l'exigence de mappage utilisateur-ressource. Par conséquent, il est mieux adapté à un système distribué que le modèle CAD. Le CAO est généralement appliqué dans les systèmes de sécurité multicouches, où chaque objet ainsi

que le sujet est identifié avec différents niveaux de sécurité, par exemple : classifier les usagers et les objets dans des classes de confidentialité (Très secret, Secret, Confidentiel …), fixer la règle qu'un usager à un certain niveau ne peut pas lire un objet classifié à un niveau supérieur ex : un infirmier simple, étant au niveau 'Confidentiel' ne peut pas lire des informations classifiées 'Secret'.

- **Contrôle d'accès basé sur les rôles (CABR) :** le concept du CABR a commencé avec les systèmes en ligne multi-utilisateurs et multi-applications lancés dans les années 1970 [67]. La notion fondamentale du CABR est que les autorisations sont associées aux rôles et que les utilisateurs sont affectés aux rôles. Cela simplifie considérablement la gestion des autorisations. Des rôles sont créés pour les diverses fonctions dans une organisation et les utilisateurs se voient attribuer des rôles en fonction de leur responsabilités et qualifications. Les utilisateurs peuvent être facilement réaffectés d'un rôle à un autre. De nouveaux droits peuvent être accordés aux rôles à mesure que de nouvelles applications et de nouveaux systèmes incorporés et des autorisations peuvent être révoquées des rôles selon les besoins. Un rôle est correctement considéré comme une construction sémantique autour de laquelle la politique de contrôle d'accès est formulée. La collection particulière d'utilisateurs et d'autorisations réunis par un rôle est transitoire. Le rôle est plus stable car les activités d'une organisation ou les fonctions changent généralement moins fréquemment. Un rôle peut représenter la compétence pour effectuer des tâches spécifiques, comme un médecin ou un pharmacien. Un rôle peut incarner l'autorité et la responsabilité, par exemple, un superviseur de projet. L'autorité et la responsabilité sont distinctes de la compétence. Une personne peut être compétente pour diriger plusieurs départements, mais est chargée d'en diriger un. Les rôles peuvent aussi refléter des affectations spécifiques de service qui sont permutées par plusieurs utilisateurs, par exemple, un médecin de service. Les modèles et implémentations du CABR peuvent accueillir toutes ces manifestations du concept de rôle. La différence majeure entre la plupart des implémentations de groupe et le concept des rôles est que les groupes sont généralement traités comme une collection d'utilisateurs et non comme une collection d'autorisations. Un rôle est à la fois une collection d'utilisateurs d'un côté et une collection d'autorisations de l'autre. Le rôle sert d'intermédiaire pour ces deux collections ensemble.

Les modèles ci-dessus ont été développés pour une attribution statique des autorisations des utilisateurs. Cependant, la relation entre les ressources et les utilisateurs est dynamique dans les soins de santé. Pour combler les exigences du contrôle d'accès dans un système distribué, les modèles de contrôle d'accès traditionnels ont été améliorés comme suit :

- **Le contrôle d'accès basé sur les attributs (CABA)**: également connu sous le contrôle d'accès basé sur la stratégie, le CABA définit un contrôle d'accès dans lequel les droits d'accès sont accordés aux utilisateurs par l'utilisation de politiques qui combinent les attributs ensemble. Les politiques peuvent utiliser tout type d'attributs (utilisateur, ressource, objet, environnement, etc.). Ce modèle prend en charge la logique booléenne, dans laquelle les règles contiennent ***« si… alors… »***, au sujet de qui fait la demande, la ressource, et l'action. Par exemple, si le demandeur est un médecin, permettre alors un accès en lecture/écriture aux données sensibles [66].
- **Contrôle d'accès basé sur le contrôle de l'utilisation (CABU)** : le principal objectif du CABU est de gérer les sessions utilisées par les utilisateurs une fois les droits d'accès accordés. C'est un modèle basé sur des attributs, et ses droits d'accès aux ressources sont attribués en fonction de l'objet, du sujet ou des propriétés d'environnement, qui sont définies dans la forme de politiques, conditions et autorisations. Il traite les problèmes générés dans la phase d'autorisation, avant l'exécution de l'accès, après l'exécution de l'accès, ou même pendant l'exécution. De plus, il a la capacité de supporter la mutabilité des attributs, en d'autres termes, si un problème se produit dans la politique de sécurité (pendant l'exécution) suite à une altération de certains attributs d'accès, l'accès autorisé est annulé et l'utilisation devient invalide [62].
- **Contrôle d'accès de surveillance de référence (CASR)** : ce modèle se compose d'un ensemble de mécanismes de validation de référence et de demandes de décision d'accès généré par un point de décision qui applique une politique de contrôle d'accès sur la capacité des sujets à procéder à des opérations sur des objets distribués. Le mécanisme de validation de référence doit répondre à plusieurs caractéristiques, telles que l'évaluabilité, l'invocabilité, la capacité de non-contournement et la capacité inviolable. Cependant, le modèle CASR a une architecture basée sur des systèmes traditionnels, qui entraîneront des coûts de calcul élevés et une latence importante lors des demandes d'accès [62].
- **Contrôle d'accès basé sur la capacité (CABCap) :** dans les systèmes basés sur CABCap, les droits d'accès sont accordés aux sujets en se basant sur le concept de capacité, qui est un transfert et une preuve d'autorité infalsifiable (par exemple, une clé, un ticket, un jeton), et décrit un ensemble de droits d'accès pour chaque sujet. Il est à noter que, dans ce modèle, la validation des droits d'accès des sujets sont généralement conduits par une entité, qui se révèle être un point de défaillance unique. Afin de résoudre ce problème, des modèles CapBAC distribués ont été proposé, où la validation des droits d'accès est effectuée par les objets IoT plutôt qu'une entité centralisée. Cependant, les objets IoT sont généralement à

faible capacité et peuvent donc être facilement compromis, de sorte qu'on ne peut pas leur faire entièrement confiance pour agir en tant qu'entités de validation des droits d'accès [68].

- **Contrôle d'accès basé sur la blockchain (CABBC) :** grâce à l'invention des SC, la BC est devenue une plateforme prometteuse pour développer des applications distribuées et fiables, et a attiré une attention considérable de la part des chercheurs de la communauté IoT dans le domaine du contrôle d'accès. Nous avons distingué trois modèles de contrôle d'accès basés sur BC : le CA basé sur les transactions, le CA basé sur les contrats intelligents et le CA basé sur les jetons. Toute action ou transfert de valeurs dans la BC est considéré comme une transaction. Un contrat intelligent (SC) quant à lui est un script créé pour effectuer diverses opérations telles que l'exécution d'un programme, le stockage de valeurs, etc. et le jeton est un actif numérique qui sert à enregistrer des informations.

II.5 Etat de l'art sur l'utilisation de la BC pour appliquer le contrôle d'accès

II.5.1 Contrôle d'accès basé les transactions

Les auteurs dans [69], ont utilisé des transactions (Tx) pour appliquer le contrôle d'accès. Ils ont défini deux types de transactions, la première initialement défini par le propriétaire de la ressource et permet de donner l'accès au demandeur et la seconde permet de transférer l'accès d'un demandeur à un autre sans l'intervention du propriétaire de la ressource. Ainsi, tout utilisateur peut les consulter à tout moment afin de vérifier qui détient actuellement les droits pour effectuer une action donnée sur une ressource donnée. Les auteurs dans [70] ont proposé une plateforme de contrôle d'accès par attributs où ils ont utilisé quatre types de transactions, la première permet l'enregistrement du sujet, la deuxième permet le dépôt et la publication de l'objet, la troisième permet la demande d'accès et la dernière l'octroi de l'accès. Dans [71], les auteurs ont présenté Healthchain, un Framework de préservation de la vie privée dans les soins de santé basé sur les transactions, où ils ont utilisé deux sous-chaînes de blocs, la première stocke les hachages des données de santé de l'utilisateur et utilise deux types de transactions : la transaction IoT utilisée pour protéger l'intégrité des données IoT et la transaction de clé utilisée pour le contrôle d'accès, et la seconde sous-chaîne stocke les hachages du diagnostic du médecin et utilise un type de transaction appelé transaction de diagnostic. Cependant, le contrôle d'accès basé sur les transactions n'a pas la capacité de prendre en charge la création, l'invocation, et la suppression d'une politique d'accès qui est une exigence du CA (passé en revue dans la section II.3.3), ni l'exécution de programmes.

II.5.2 Contrôle d'accès basé sur les contrats intelligents

Dans [72], les auteurs ont présenté une architecture pour arbitrer les rôles et les autorisations dans l'IoT, pour cela ils ont utilisé un seul contrat intelligent pour y définir toutes les fonctionnalités du système afin de simplifier tout le processus dans le réseau BC et de réduire le surcoût de communication entre les nœuds ; les managers (entité responsable de la gestion des autorisations d'un ensemble d'appareils IoT) sont les seules entités capables d'interagir avec le SC afin de définir de nouvelles politiques dans le système. Cependant, dans cette approche, le propriétaire de la ressource ne peut pas contrôler ses propres données.

Un mécanisme de CABR combiné au SC a été proposé dans [73] pour partager des images filmées par les caméras de surveillance en toute sécurité et empêcher l'accès indiscriminé par des tiers, ou les données sensibles telles que les images filmées et les politiques d'accès sont enregistrées dans le système de gestion et seul le hachage de ces derniers est enregistré dans les blocks de la BC. Ils ont utilisé un seul type de SC qui a comme objectif de vérifier si le demandeur de la création du bloc est un utilisateur légitime ou non. Les auteurs dans [74] ont proposé un schéma de contrôle d'accès où ils ont combiné le CABA et les SC dans un système avec plusieurs autorités qui participent à la validation de différents attributs. Ils ont utilisé quatre types de contrats intelligents pour définir les interactions entre les différentes entités, deux entre les propriétaires de données et les utilisateurs de données, et deux autres entre les utilisateurs de données et les autorités d'attribution. Un autre Framework de contrôle d'accès combinant le CABA et les SC a été proposé dans [75]. Ce dernier se compose de quatre types de SC qui définissent la gestion des politiques, la gestion des attributs du sujet, la gestion des attributs de l'objet et la gestion du contrôle d'accès. Un autre contrôle d'accès aux attributs basé sur SC a été proposé dans [76], où chaque politique XACML (eXtensible Access Control Markup Language) est traduite en un SC. Dans [77] les auteurs ont utilisé deux types de contrats, le contrat de transfert des droits pour transférer les droits d'accès et le contrat de contrôle d'accès pour appliquer le contrôle d'accès. La fonction du contrat de transfert consiste principalement à confirmer si l'identité de l'expéditeur du message est légale et à vérifier si le droit peut être transféré et la fonction principale du contrat de contrôle d'accès est de juger si le récepteur répond aux exigences de la politique de contrôle d'accès. Cependant, la relation entre les ressources et les utilisateurs est dynamique dans les systèmes distribués, cela signifie que les systèmes de contrôle d'accès doivent pouvoir s'adapter automatiquement aux informations de contexte d'accès changeant dynamiquement, comme le temps, le lieu et les situations.

Un contrôle d'accès dynamique basé sur les contrats intelligents a été proposé dans [78, 79]. Dans [78], l'architecture du système se compose de trois types de contrats intelligents. Chaque

propriétaire de ressource définit un contrat de contrôle d'accès pour un couple sujet-ressource. Dans [79], l'architecture du système se compose de quatre formes de contrats intelligents utilisés pour la vérification des utilisateurs, l'autorisation d'accès, la détection des comportements inappropriés et la révocation d'accès. Cependant, ces approches ainsi que celles citées ci-dessus consomment beaucoup de gaz lié au déploiement des contrats et à la communication entre les contrats et les nœuds.

Les contrats intelligents ont également été utilisés pour déployer le contrôle d'accès dans [80, 81]. Dans [80], un retour sur l'état du système est généré à chaque exécution du contrat puis envoyé au propriétaire de la ressource pour mettre à jour dynamiquement la politique de sécurité. Dans [81], les contrats intelligents sont utilisés pour tenir compte des différents rôles des patients, des prestataires et des tiers sur la BC.

Les auteurs dans [82] ont modifié la structure classique de la BC en regroupant les mineurs en clusters afin de stocker et traiter les données au niveau du cluster le plus proche du patient afin de réduire la latence d'accès.

Le Tableau II.2 présente une analyse comparative des modèles de contrôle d'accès basé sur les SC existants, en précisant les avantages et les inconvénients de chaque proposition.

Référence	Modèle	Avantages	Inconvénients
[69-71]	TX	Réduit la latence d'accès	Ne prends pas en charge l'ajout, la modification et la suppression des politiques d'accès Ne prend pas en charge l'exécution de programme
[72,73]	SC	Réduit le nombre de contrats déployé ainsi que la latence d'accès	Empêche le propriétaire de la ressource de contrôler ses données
[74,75,77]	SC	Chaque contrat a un rôle précis	Augmente le nombre de transactions et la latence d'accès
[78-80]	SC	Mets à jour dynamiquement les politiques d'accès	Augmente la latence d'accès
[81]	SC	Réduit le nombre de contrats déployé	Le control d'accès n'est pas précis

Tableau II.2 Analyse comparative des travaux connexes sur les SC.

II.5.3 Contrôle d'accès basé sur les jetons

Dans [61], les auteurs ont utilisé les jetons pour représenter les droits d'accès où les jetons peuvent être délivrés d'un pair à un autre par le biais de transactions. Lors de la remise d'un jeton, l'expéditeur intègre le contrôle d'accès dans les scripts de verrouillage de la sortie de transaction. Le destinataire du jeton doit déverrouiller les scripts de verrouillage pour prouver la possession du jeton (prouver son droit d'accès à une certaine ressource). En utilisant ce schéma, un pair peut se voir accorder l'accès en recevant un jeton, accorder des droits d'accès à un autre sujet en délivrant un jeton, et accéder à un objet en dépensant un jeton. Bien que l'utilisation de scripts de verrouillage pour le contrôle d'accès soit une excellente idée, la capacité informatique de verrouillage des scripts est considérablement limitée, et ils n'ont pas pris en considération le fait que les appareils IoT sont généralement de faible capacité et ne peuvent pas faire partie de la blockchain. Dans [83], les auteurs ont défini TRABAC, un système de contrôle d'accès qui allie le CABR et le CABA. Il est couplé à l'utilisation des NFT qui représentent les actifs virtuels et les attributs du sujet et des objets dans les applications de chaîne d'approvisionnement. Dans ce travail, les auteurs ont utilisé deux types de jetons NFT, le premier type appelé AgliveToken (AGL) contenant trois membres : le type de jeton, la balise d'attribut du jeton et les métadonnées attachées au jeton, le deuxième type est un jeton enfant appelée Activité (AC) qui suit toutes les modifications apportées à un jeton AGL qui avait été ajouté à la BC. OAuth 2.0 est un protocole standard de l'industrie pour l'autorisation d'accès présenté dans [84]. Tous les flux protocolaires de ce protocole aboutissent à la création d'un jeton d'accès de type NFT, qui est ensuite utilisé par un utilisateur pour demander l'accès à une ressource protégée. Lorsqu'un jeton est créé, son identifiant et ses métadonnées sont spécifiés : ces deux propriétés sont en lecture seule et ne peuvent pas être modifiées. De plus, les propriétaires de jetons ne peuvent pas transférer leurs jetons, la seule entité qui peut invoquer la méthode de transfert est le propriétaire du contrat qui a créé le jeton. Les auteurs dans [85], ont présenté un Framework décentralisé d'authentification, d'autorisation et de comptabilité pour les appareils IoT utilisant des jetons basés sur les capacités, le contrat intelligent de contrôle d'accès utilise le concept de jeton NFT pour créer le Jeton CapBAC. La structure du jeton CapBAC est définie par un nombre de champs tels que l'identifiant du jeton, le nom du jeton, l'identifiant du sujet, de l'objet, et du propriétaire du jeton, l'URI, etc…Lorsqu'un utilisateur envoie une demande d'accès à la ressource, il est redirigé à l'adresse du contrat intelligent de contrôle d'accès déployé. Le demandeur émet une transaction à cette adresse pour obtenir un jeton, le contrat intelligent génère un jeton avec les arguments spécifiés. Le jeton est ensuite émis est transmis au demandeur. En fonction de la durée d'expiration du jeton d'accès, le demandeur peut émettre des commandes d'accès.

Référence	Modèle	Avantages	Inconvénients
[61]	Jeton fongible	L'utilisation des scripts rajoute une couche de sécurité	La capacité informatique de verrouillage des scripts est limitée. Le jeton est facilement échangeable avec un autre
[83]	NFT	L'utilisation d'un jeton enfant améliore la sécurité	Augmente la consommation de gas lié à la création de deux types de jetons
[84]	NFT	L'utilisation du NFT rajoute une couche de sécurité	Déploie beaucoup de contrats pour déployer les NFT ce qui augmente la consommation de gas
[85]	NFT	L'utilisation du NFT rajoute une couche de sécurité	La structure du jeton est trop grande ce qui augmente la consommation de gas lors de la création du jeton

Tableau II.3 Analyse comparative des travaux connexes sur les NFT.

II.6 Conclusion

Dans ce chapitre nous avons passé en revue la blockchain, nous avons vu que son intégration optimise et améliore les systèmes de soins de santé de différentes façons. Parmi les applications prometteuses de cette technologie dans ce domaine le contrôle d'accès. Ce dernier basé sur la blockchain améliore la protection de la vie privée du patient, lui donne la possibilité de gérer l'accès à ses ressources, et offre une plateforme d'audit. Dans le chapitre suivant, nous allons présenter l'architecture des systèmes de soins de santé proposée dans le cadre de cette thèse ainsi que le mécanisme de contrôle d'accès que nous avons suggéré.

Chapitre III : Architectures proposées du contrôle d'accès

III.1 Introduction

Les soins de santé basés sur l'IoT impliquent que les patients partagent à distance leurs données personnelles et physiologiques avec le personnel hospitalier, ce qui peut mettre en danger la vie privée du patient. Ainsi, la mise en place d'un contrôle d'accès est obligatoire. Par conséquent, l'objectif de ce chapitre est de proposer un contrôle d'accès distribué et fiable pour les systèmes de soins de santé basés sur l'IoT en utilisant la technologie blockchain activée par les contrats intelligents. Pour ce faire, nous proposons trois architectures différentes de contrôle d'accès basées sur contrats intelligents. La première proposition de CA se fait via un type de SC nommé ACC (Access Control Contract), où chaque propriétaire de ressource établit un seul contrat intelligent dans lequel il définit les politiques d'accès à ses ressources pour tous les sujets du système. La deuxième proposition se fait également via un type de SC, mais chaque propriétaire de ressource établit un contrat intelligent pour chaque sujet, où il définit les politiques d'accès pour ce sujet. Et enfin, dans la troisième proposition, le CA se fait à travers trois types de SC : le ACC, le contrat du registre (RC ou Register Contract) et le contrat de juge de mauvaise conduite (MJC Misconduct juge contract). Chaque propriétaire de ressource établit un ACC pour tous ces sujets, et le MJC et le RC sont définit par l'administrateur du système.

Ce chapitre est organisé comme suit : dans la section 2, nous passons en revue quelques travaux connexes relatifs aux architectures de soins de santé et de contrôle d'accès basé les SC qui ont été proposées dans la littérature ces dernières années. Dans la section 3, nous présentons en détail l'architecture générale de soins de santé basée sur l'IoT proposée dans le cadre de cette thèse et nous expliquons le rôle de chaque entité qui la compose. Dans la section 4, nous présentons les éléments de base du contrôle d'accès basé sur les SC, nous expliquons son flux de travail, ainsi que la fonction de contrôle d'accès dynamique que nous utilisons dans le contrat intelligent, puis nous définissons les trois architectures proposées en détails. Dans la section 5, nous présentons une autre architecture de contrôle d'accès mais cette fois ci basée sur les NFT, nous expliquons son flux de travail, et nous définissons la structure du jeton. Nous concluons ce troisième dans la section 6.

III.2 Travaux connexes

III.2.1 Architectures des systèmes de soins de santé

Une architecture des systèmes de soins de santé a été proposée dans [86] contenant principalement trois composants : l'éditeur, le courtier et l'abonné. L'éditeur représente un réseau de capteurs connectés et d'autres dispositifs médicaux. Le courtier est responsable du traitement et du stockage des données acquises dans le cloud et l'abonné se livre au suivi continu des informations

du patient accessibles et visualisables via un smartphone, un ordinateur, une tablette, etc. Dans [79], les auteurs ont aussi proposé une architecture de systèmes de soins de santé basée sur le cloud, qui est composée de plusieurs entités telles que : les hôpitaux, les patients, les appareils de santé intelligents, les unités de contrôle médical et le cloud. Les appareils de santé intelligents incluent tout appareil qui permet le suivi en temps réel de la santé des patients. Les unités de contrôle médical sont les appareils avec une certaine puissance de calcul qui agissent comme un pont entre les appareils de santé intelligents et les nœuds de la BC qui sont installés dans les hôpitaux. Ils cryptent les DSE générés à partir des appareils de santé intelligents et les transmettent à l'hôpital respectif via le canal de communication sécurisé. Le cloud est responsable du stockage des DSE et de l'authentification des informations d'identification d'une entité. Cependant et comme déjà mentionné dans le chapitre I, le cloud computing peut représenter un point de défaillance unique, augmentant ainsi la latence, alors que dans les systèmes de soins de santé, la réponse doit être en temps-réel.

Les auteurs dans [87] ont proposé une architecture à base de fog computing, où ils ont divisé le fog en deux sous couches : la couche fog et la couche passerelle afin de minimiser le temps de calcul, et d'assurer la mobilité et la disponibilité. Dans [78], les auteurs ont proposé une architecture où ils ont utilisé des passerelles afin de minimiser la latence, mais leur travail ne mentionne pas l'intégration du fog dans ces passerelles afin d'en optimiser l'utilisation. Dans l'architecture proposée par les auteurs dans [88], des applications mobiles jouent le rôle du fog en permettant la visualisation et l'analyse des facteurs environnementaux et en aidant les infirmières à mesurer les signes vitaux des patients et de préparer des rapports médicaux. Dans [82], les auteurs ont modifié la structure classique de la BC en regroupant les mineurs en clusters afin de stocker et traiter les données au niveau du cluster le plus proche du patient afin de réduire la latence d'accès.

III.2.2 Architectures de contrôle d'accès

Les auteurs dans [73] ont proposé une architecture de CA basée sur un seul SC qu'ils utilisent pour vérifier si le demandeur de la création du bloc est un utilisateur légitime ou non. Dans [77], les auteurs ont proposé une architecture de CA basée sur deux contrats, le premier contrat est utilisé pour transférer les droits d'accès et le second pour appliquer le contrôle d'accès. Cependant nous reprochons à ces deux architectures le fait qu'elles privent le propriétaire de l'objet du contrôle de ses données.

Les auteurs dans [74] ont proposé une architecture de contrôle d'accès basée sur quatre types de contrats intelligents pour définir les interactions entre les différentes entités. En premier lieu, le

propriétaire de l'objet (PO) crée un contrat pour chaque sujet et il y définit ses politiques d'accès. Ensuite le sujet crée un autre contrat avec chaque autorité d'attribut qui contient la fonction checkA qui demande la validation de ses attributs. Par la suite, le sujet crée un contrat avec le PO qui contient la fonction checkAT qui vérifie que le sujet détient suffisamment d'attributs pour envoyer la clé secrète AES. Pour finir chaque autorité d'attribut crée un contrat entre elle-même et le sujet qui contient la fonction checkAttribute pour valider les attributs de sujet et la fonction sendToken pour l'octroi des attributs. L'architecture de contrôle d'accès proposée dans [75] se compose de quatre types de SC, les administrateurs des sujets déploient le contrat de gestion des attributs du sujet, les administrateurs des objets déploient le contrat de gestion des attributs des objets, les administrateurs des propriétaires d'objets déploient le contrat de gestion des politiques et des contrats de contrôle d'accès sont déployés mais il n'est pas mentionné dans l'article par qui ces derniers sont déployés. Tout comme dans l'architecture précédente, le nombre de contrats déployés dans cette architecture est très élevé, le coût ainsi que la latence résultante de la surcharge de communication entre les contrats sont élevés. L'architecture proposée dans [76] traduit chaque politique écrite en langage XACML en un SC résultant ainsi en un nombre énorme de contrats. Dans [78], l'architecture du CA se compose de trois types de contrats intelligents, le contrat de contrôle d'accès, le contrat juge de mauvaises conduites et le contrat d'enregistrement où toutes les adresses des contrats déployés sont enregistrées. Chaque propriétaire de ressource définit un contrat de contrôle d'accès pour un couple sujet-ressource. Dans [79], l'architecture se compose de quatre formes de contrats intelligents utilisés pour la vérification des utilisateurs, l'autorisation d'accès, la détection des comportements inappropriés et la révocation d'accès. Comme pour les architectures précédentes, ces deux dernières consomment beaucoup de gaz lié au déploiement des contrats et à la communication entre les contrats et les nœuds. De même pour les architectures proposées dans [80, 81].

III.3 Architecture considérée pour les systèmes de soins de santé

L'architecture du système considéré dans notre contribution est applicable dans des cas d'utilisation de téléconsultation, et de télésurveillance, où le patient est à son domicile équipé de capteurs et d'actionneurs, et le professionnel de santé est à l'hôpital ou à son cabinet et accède à distance aux données collectées du patient et à son DSE.

Comme illustré sur la Figure III.1, l'architecture se compose principalement d'entités telles que les patients, les professionnels de santé, les nœuds fog qui sont représentés par les appareils des patients et du personnel hospitalier, les nombreux appareils IoT (capteurs et actionneurs), les

serveurs fog, le serveur Cloud, et la technologie des contrats intelligents Blockchain. Les principaux rôles de ces entités sont expliqués ci-dessous :

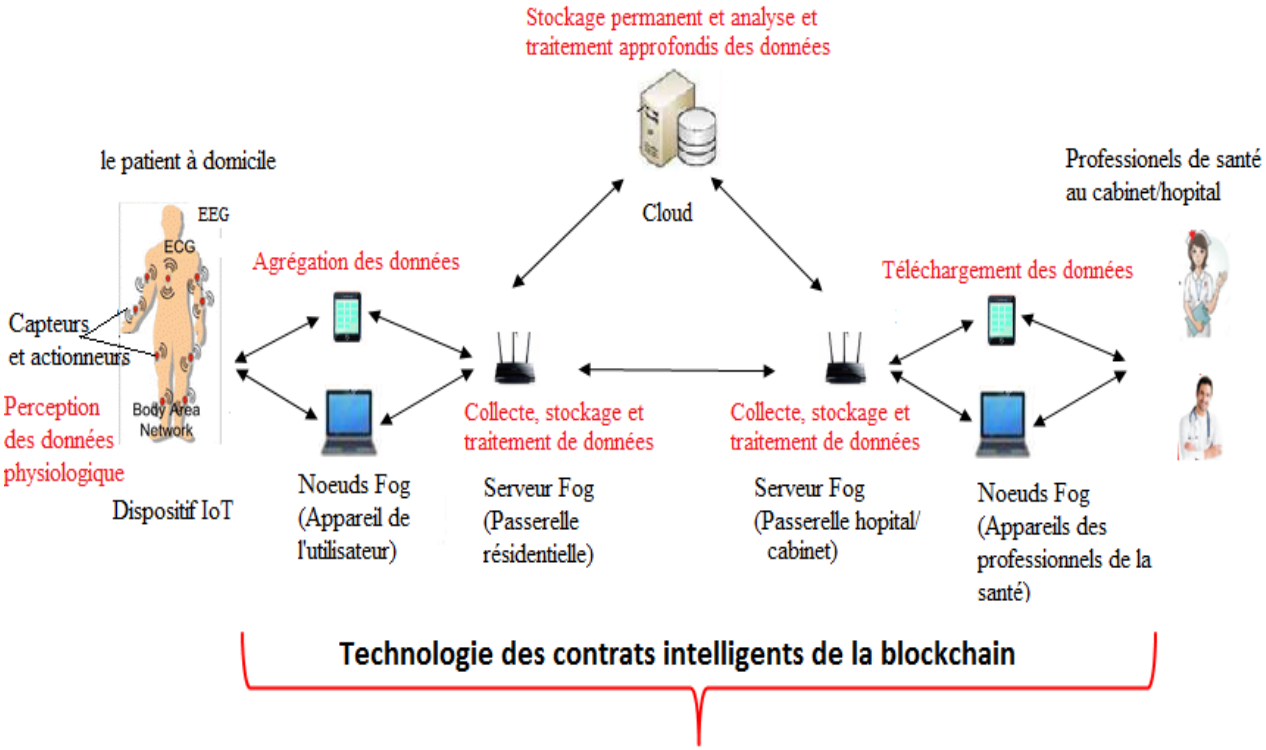

Figure III.1. Architecture proposée des systèmes de soins de santé [89].

- **Patients** : le patient à son domicile est une partie essentielle de notre système. Il est le véritable propriétaire de ses données. Il peut accéder à son DSE, et il peut également gérer les demandes d'accès à ce dernier via des contrats intelligents, puisque c'est lui qui définit ses politiques d'accès et déploie son contrat intelligent dans la BC sur la base de conseils de son médecin et de l'assistance du personnel informatique de l'hôpital si besoin.
- **Professionnels de la santé** : les professionnels de la santé dans notre système comprennent les médecins, les infirmiers et le support informatique, qui accèdent au DSE du patient stocké dans les serveurs fog/cloud depuis leurs cabinets pour vérifier son état de santé.
- **Dispositifs IoT** : les dispositifs IoT considérés dans notre architecture sont les capteurs et les actionneurs. Les capteurs peuvent être divisés en deux groupes : (1) Les capteurs physiques intelligents sont utilisés pour surveiller numériquement la santé des patients et suivre leur bien-être physique. Dans l'architecture ci-dessus, ils sont utilisés pour surveiller l'ECG, la température corporelle, la fréquence cardiaque, l'hémoglobine, le lecteur de glucose…etc (2) Les capteurs intelligents virtuels sont utilisés pour capturer les informations sur la santé des patients. Ils assurent la surveillance, le diagnostic, la consultation à distance et le suivi à distance des patients. Ces capteurs sont extrêmement

fiables, non invasifs et peu coûteux. Par conséquent, ils peuvent être largement distribués à divers hôpitaux ou cliniques publics pour surveiller en permanence l'état de santé des patients. Les actionneurs quant à eux exécutent un programme, par exemple, des pilules intelligentes implantées sous la peau qui libèrent des médicaments, lorsqu'ils reçoivent une commande des médecins, ou la pompe à insuline qui libère de l'insuline périodiquement ou quand elle reçoit une commande, etc.

- **Nœuds fog** : un nœud fog est un appareil utilisateur (ordinateur de bureau, ordinateurs portables, smartphones et tablettes) grâce auquel les utilisateurs (patients ou professionnels de la santé) peuvent profiter de services (ex : vérifier la température corporelle actuelle, vérifier le niveau d'insuline du patient) via une application. Ces appareils prennent en charge les protocoles de communication sans fil (Wifi, Bluetooth, 4G) pour agréger les données de divers appareils IoT hétérogènes dans des DSE. Par la suite, les DSE seront transféré sur les serveurs fog pour le prétraitement et le stockage temporaire.
- **Serveurs Fog** : ce sont des nœuds répartis géographiquement et placés plus près de la source de données (sur la passerelle résidentielle et sur la passerelle de l'hôpital/ cabinet). Le serveur fog est un petit serveur cloud chargé de collecter, stocker, traiter et analyser les données sensibles au facteur temps des nœuds fog et de fournir des services à la demande. De plus, il doit faire un filtrage des données en d'autres termes il doit décider quelles informations doivent être envoyées au cloud (ex : celles qui nécessitent un stockage permanent ou un traitement en profondeur), dans quel format de données et quand. Les ressources fog sont intégrées aux points d'accès, aux routeurs et aux passerelles réseau, aux côtés des fonctions réseau génériques.
- **Cloud** : le cloud dans notre architecture offre un calcul à grande échelle, facilite le stockage et assure la fiabilité et l'évolutivité du système. Les ressources cloud (infrastructure de calcul, stockage, etc.) sont structurellement orchestrées et virtualisées. Les données nécessitant un stockage permanent ou un traitement approfondi sont envoyés vers le cloud.
- **Contrat intelligent Blockchain** : les contrats intelligents de la technologie BC sont utilisés dans notre système pour fournir un contrôle d'accès distribué. Notre approche applique une conception spécifique pour éviter d'intégrer la technologie blockchain dans les appareils IoT. La BC est implémentée par tous les pairs à l'exception des capteurs et des actionneurs car ils sont limités par le calcul. Cela augmente l'applicabilité de notre solution dans un grand nombre de scénarios IoT avec des capacités limitées.

Dans notre architecture, la couche fog est divisée en deux sous couches : les nœuds fog et les serveurs fog. Le nœud fog sera placé dans les appareils utilisateurs, il servira à collecter les données

et les rassembler dans les DSE, mais le stockage et le calcul se font au niveau du serveur fog qui est placé au niveau de la passerelle. Ainsi nous augmentons l'évolutivité du système par rapports aux architectures proposées dans [78, 79], et nous assurons une capacité de stockage et de calcul plus élevé que celles des architectures proposées dans [87, 88].

Dans notre architecture, chaque entité dispose de ressources (ex : des services, des données, de l'espace de stockage) qui sont nécessaires à d'autres entités. Par conséquent, les droits d'accès doivent être définis par les propriétaires de données dans le SC pour empêcher l'utilisation illégale de leurs ressources. Les sujets n'auront accès aux données que s'ils respectent les politiques définies dans le SC. Plus de détails sur le SC seront donnés dans la section suivante.

Les DSE dans notre architecture contiennent des renseignements sur la santé du patient notamment les données collectées par les capteurs, ses visites à l'hôpital, à la clinique et chez le médecin, les résultats de ses analyses ou d'imagerie (radiographie), ainsi que ses antécédents médicaux notamment ses allergies et ses ordonnances.

Il faut noter que dans cette architecture, les données comme les DES ne sont pas enregistrées dans la BC. Un hachage des DSE est calculé en utilisant la fonction de hachage SHA-256, et le résultat de ce hachage est stocké dans la BC afin d'augmenter la sécurité des données.

III.4 Architectures proposées du contrôle d'accès basé sur les contrats intelligents

Cette section présente le flux de travail proposé pour le contrôle d'accès basé sur les contrats intelligents, la méthode de contrôle d'accès, et les trois architectures proposées.

Dans le schéma de contrôle d'accès que nous proposons, le propriétaire de l'objet (notamment le patient) est l'entité responsable de la définition des droits d'accès et du déploiement du SC, cela garantit que seules les entités autorisées peuvent accéder au DSE tandis que les tierces parties ne sont pas en mesure d'y accéder. Notre méthode de contrôle d'accès proposée facilite la protection de l'identité et l'intégrité des données.

III.4.1 Hypothèses

- Nous supposons que l'administrateur du système hospitalier a enregistré toutes les entités de notre architecture dans le système d'identification et qu'ils ont un numéro d'identification unique. Ensuite, nous enregistrons chaque entité dans la BC avec son identifiant unique afin d'obtenir une adresse unique.

- Étant donné que les capteurs et les actionneurs ne peuvent pas effectuer de calculs intensifs, ils ne font pas partie de la BC, nous supposons donc que ce sont des entités de confiance.
- Nous supposons que l'adresse des différents contrats est connue des nœuds fog (ils ont besoin de l'adresse pour récupérer le SC correspondant).

III.4.2 Eléments de base du contrôle d'accès

Le sujet, l'objet et le propriétaire de l'objet sont les éléments de base pris en compte dans notre système de contrôle d'accès. Le sujet est une entité active qui demande l'accès à une entité passive appelée objet. Lorsqu'un sujet accède à un objet, il accède à ses informations. Le propriétaire de l'objet est l'entité responsable de la définition des politiques d'accès et de l'octroi des autorisations au sujet. Le tableau III.1 définit les éléments de base considérés dans notre système.

Toute entité disposant de ressource doit implémenter le contrôle d'accès, et les droits d'accès doivent être définis par les propriétaires de l'entité dans le SC pour empêcher l'utilisation illégale de leurs ressources.

Sujet	**Objet**	**Propriétaire de l'objet**
Patient/Personnel de la santé	Dispositif IoT (capteurs et actionneurs)	Patient
Patient/Personnel de la santé	Nœuds fog (appareils des utilisateurs)	Patient/Personnel de la santé
Patient/Personnel de la santé	Serveurs fog	Personnel de la santé (administrateur)
Patient/Personnel de la santé	Cloud	Personnel de la santé (administrateur)

Tableau III.1 Eléments de base du contrôle d'accès proposé.

III.4.3 Flux de travail du contrôle d'accès proposé

Comme illustré dans la Figure III.2, notre contrôle d'accès basé sur la BC passe par les étapes suivantes :

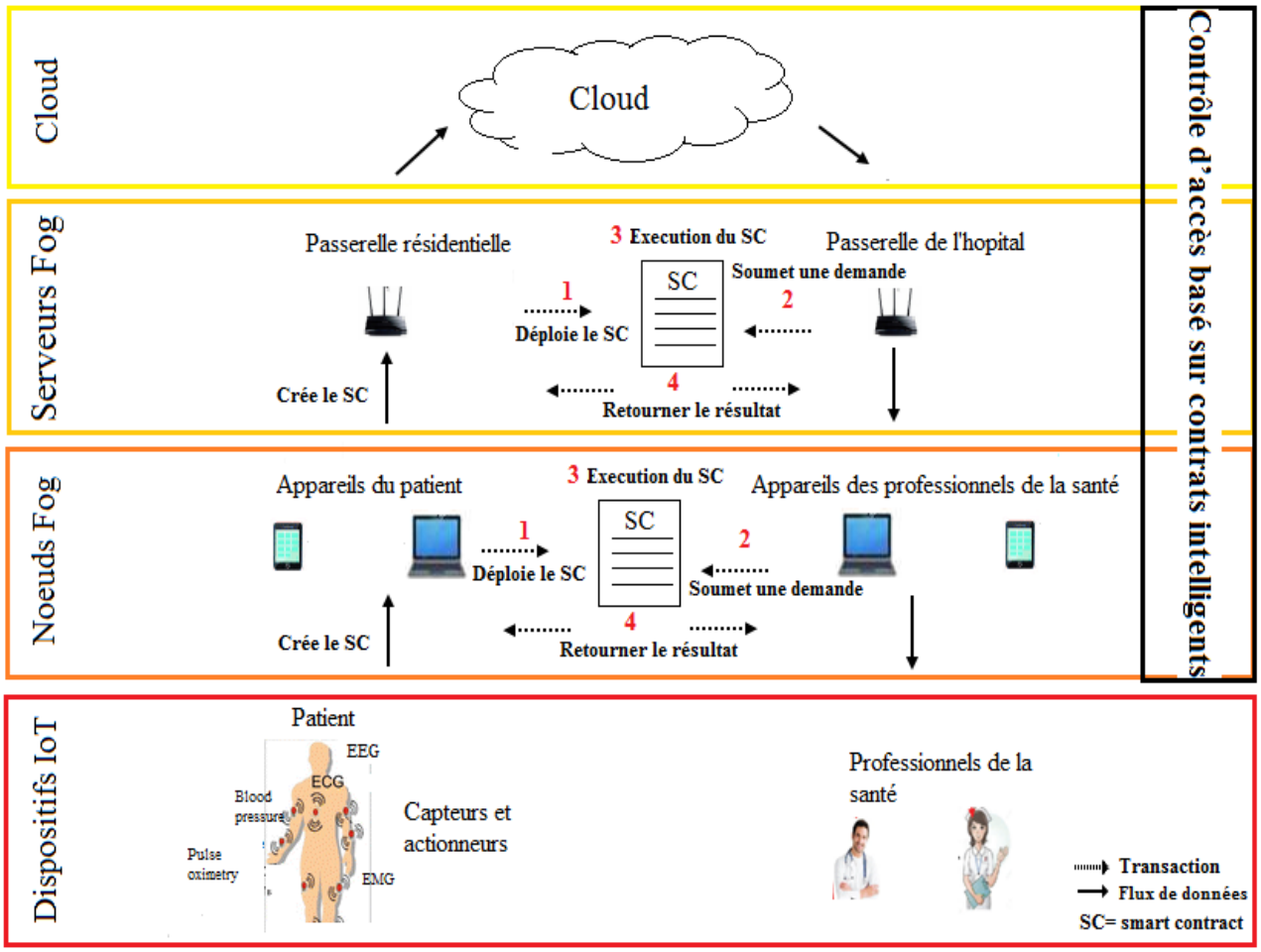

Figure III.2. Etapes du contrôle d'accès [90].

- **Étape 1 :** le propriétaire de la ressource crée un contrat intelligent appelé ACC et y définit les politiques d'accès et le déploie dans la BC.
- **Étape 2 :** le sujet demande l'accès à l'objet en envoyant les informations nécessaires dans une transaction à l'ACC de l'objet. Si l'objet est un appareil IoT, la requête est transmise au nœud fog associé.
- **Étape 3 :** lorsque l'ACC reçoit une transaction du sujet, il déclenche la méthode de contrôle d'accès qui sera expliquée dans la section suivante.
- **Etape 4 :** dès que l'exécution de la méthode de contrôle d'accès est terminée, l'ACC envoie le résultat au sujet et au propriétaire de l'objet.

Toute action qui se passe dans la BC comme le déploiement du ACC, la demande d'accès, l'exécution de la méthode de contrôle d'accès ou encore l'envoie du résultat est enregistrée dans une transaction qui sera validée et scellée dans un bloc afin d'empêcher toute modification, et de faciliter la vérification et l'audit en cas de problèmes de sécurité.

Dans l'étape 1, et afin de définir les politiques d'accès, nous avons défini des fonctions pour ajouter, mettre à jour et supprimer des politiques de contrôle d'accès dans l'ACC.

- **PolicyAdd() :** cette méthode reçoit les informations définies par le propriétaire de l'objet (créateur de l'ACC) et les ajoute à la liste des politiques.
- **PolicyUpdate() :** cette méthode reçoit les informations définies par le propriétaire de l'objet pour mettre à jour la politique.
- **PolicyDelete() :** cette méthode est appelée par le propriétaire de l'objet pour supprimer une politique.
- **DeleteACC() :** cette méthode est invoquée par le propriétaire de l'objet. Elle effectue l'opération d'autodestruction pour supprimer le code ACC, afin que l'ACC ne puisse plus être disponible.

III.4.4 Méthode de contrôle d'accès

Lorsque le sujet appelle l'ACC pour demander l'accès, la méthode AccessControl() est exécutée. Dès qu'une éventuelle erreur ou un mauvais comportement est détecté, une pénalité est calculée et retournée avec le résultat, sinon, le sujet obtient l'accès.

Le processus de validation de la méthode AccessControl() proposée passe par trois étapes, authentification, validation des droits statiques, et validation dynamique des droits comme illustré dans la Figure III.3.

- **Authentification** : à cette étape, le contrat vérifie si l'identifiant correspond à une adresse sur la BC, si oui, il passe à l'étape suivante, sinon la demande est rejetée.
- **Validation des droits statiques :** dans cette étape, le contrat procède à la vérification des droits d'accès, si le sujet a le droit d'accéder à la ressource, il passe à l'étape suivante sinon la demande est rejetée.
- **Validation dynamique des droits :** dans cette étape, le contrat vérifie dans la liste des mauvaises conduites implémentée dans l'ACC (indiquée dans le tableau III.2) si le sujet a fait des demandes d'accès fréquentes. Si c'est le cas, la demande est rejetée et une pénalité est calculée et renvoyée avec le résultat, sinon la demande est approuvée et le sujet accède à l'objet.

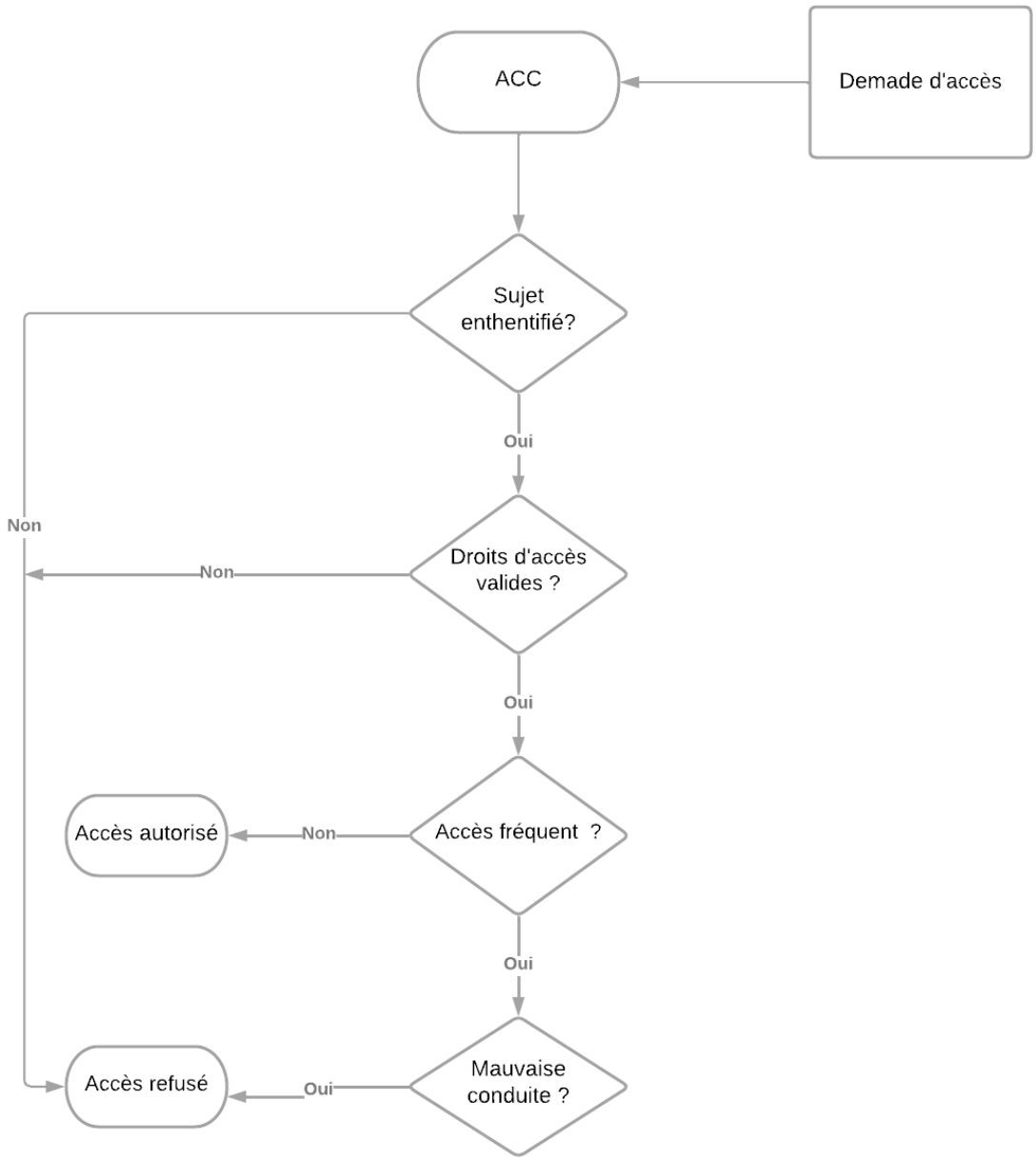

Figure III.3. Organigramme de la méthode AccessControl().

Sujet	Ressource	Action	Temps de la mauvaise conduite	Pénalité	Temps de déblocage
0x5fF07...	Fichier A	Lire	01/02/2022; 11:45	1 min	01/02/2022; 11:46
0x5fF07...	Programme B	Exécuter	02/02/2022; 17:20	3 min	02/02/2022; 17:23
0x6f803...	Programme B	Exécuter	01/02/2022; 12:30	2 min	01/02/2022; 12:32

Tableau III. 2 Liste de mauvaises conduites du ACC du patient 1

Le tableau III.2 représente la liste des mauvaises conduites implémentée dans l'ACC du patient 1 où le champs « sujet » contient l'adresse du sujet qui a effectué la mauvaise conduite, le champs « ressource » représente la ressource sur laquelle la mauvaise conduite a été effectuée, le champs « action » contient l'action qui a été produite lors de la mauvaise conduite, le champs « temps de mauvaise conduite » représente le moment où la mauvaise conduite s'est déroulée, le champs « pénalité» représente le temps pendant lequel le sujet est bloqué, et enfin le champs « temps de déblocage » représente le moment où le sujet sera autorisé de nouveau à effectuer des demandes d'accès.

Lorsqu'une mauvaise conduite se produit, la méthode Accesscontrol() comporte la fonction définie dans l'équation III.1 qu'elle utilise lors de l'étape de validation dynamique pour déterminer la pénalité correspondante (s'applique aux trois architectures).

$$Pénalité = base * intervalle \quad \text{(III.1)}$$

Où :

base : est un nombre initialisé par le propriétaire de la ressource lors du déploiement de l'ACC (ex : 10 secondes).

intervalle : est le nombre d'inconduites manifestées par le sujet.

III.4.5 Architecture de la proposition A

Le contrôle d'accès dans la proposition A se fait par le biais de plusieurs ACC. Chaque propriétaire de ressource établit un seul ACC où il définit les droits d'accès de tous les sujets du système, comme le montre la Figure III.4. Chaque ACC implémente les trois fonctions de validation (authentification, validation statique et dynamique) expliquées dans la section précédente.

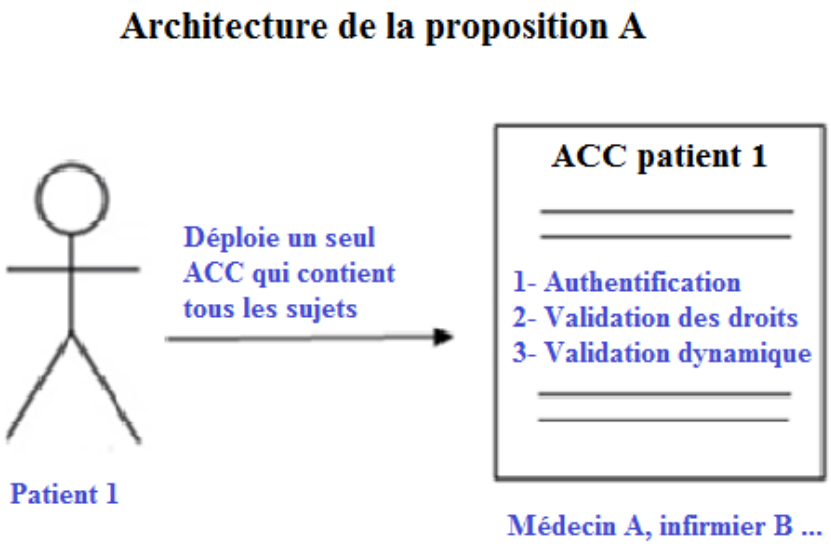

Figure III.4. Schéma de la proposition A [90].

Le diagramme de séquence représenté par la Figure III.5, montre les étapes du contrôle d'accès de la proposition A. En premier lieu, le patient 1 définit les politiques d'accès à ses ressources dans un seul contrat ACC, puis le déploie dans la BC. Ensuite lorsque le Médecin A soumet une demande d'accès aux ressources (données) du patient 1, il récupère le ACC du patient 1 (nous supposons qu'il connait son adresse), ce qui déclenche l'exécution de la méthode Accesscontrol(), il passe par les trois étapes de validation (authentification, validation statique et dynamique). Si une erreur est détectée à une des étapes, l'accès est refusé, sinon l'utilisateur se voit attribué l'accès.

Il faut noter que dans cette proposition, lorsqu'un autre sujet par exemple l'infirmier B demande accès aux ressources (données) du patient 1, le contrôle d'accès est réalisé par l'ACC du patient 1 (il passe par le même contrat à travers lequel le médecin A est passé), et dans le cas où il veut accéder à une ressource (donnée) d'un autre patient par exemple patient 2, le contrôle d'accès sera effectué par l'ACC du patient 2.

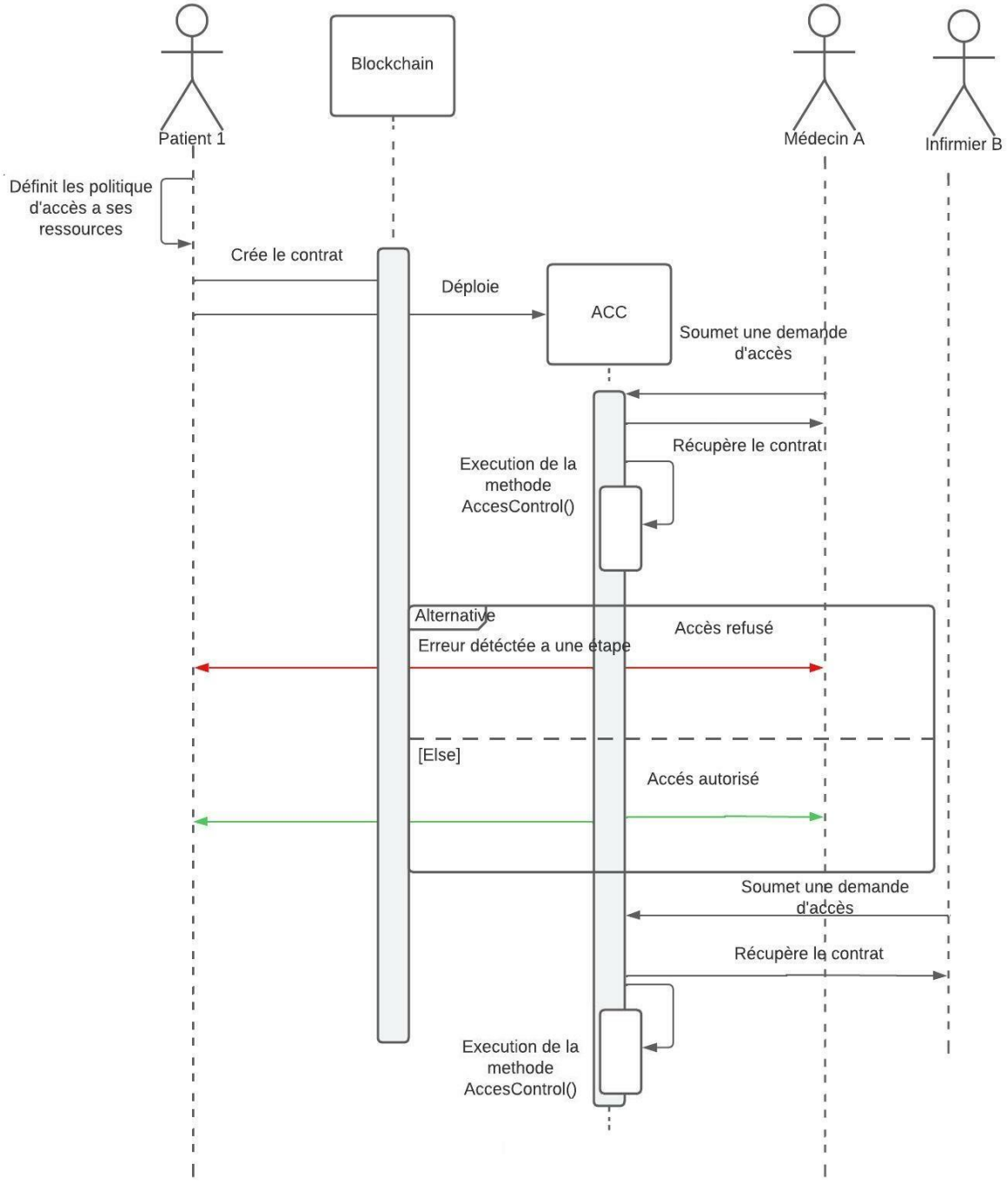

Figure III.5. Diagramme de séquence de la proposition A.

Lorsque le patient crée son ACC, il doit renseigner les champs des politiques d'accès représenté par le Tableau III.3 comme l'adresse du sujet (adresse du médecin A, de l'infirmier B…), la ressource à laquelle le sujet peut ou non accéder (résultats des analyses, données physiologique capturées…), l'action qui peut être effectuée sur la ressource (lire, écrire, exécuter) l'autorisation d'accès (autoriser, refuser), MinInt : le temps minimum autorisé entre deux requêtes successives et Limite : limite de requêtes fréquentes. Les deux derniers champs sont utilisés pour caractériser la mauvaise conduite.

Adr sujet	Ressource	Action	Permission	MinInt	Limite
0x5fF07...	Ordonnance	Ecrire	Autoriser	1 min	2
0x5fF07...	Résultat des analyses	Lire	Autoriser	30 sec	5
0x8F907...	Pompe à insuline	Exécuter	Refuser	1 min	1
0x8F907...	Données ECG	Lire	Autoriser	2 min	3

Tableau III.3 Liste des politiques d'accès d'un patient dans la proposition A.

III.4.6 Architecture de la proposition B

Le contrôle d'accès dans la proposition B consiste en plusieurs ACC comme dans la proposition A, la différence réside dans le nombre de contrats intelligents créés par le propriétaire de l'objet. Dans la proposition B, chaque propriétaire de ressource établit un contrat intelligent pour chaque sujet du système (pour N sujets, il crée N contrats) comme illustré par la Figure III.6, tandis que dans la proposition A, le propriétaire de ressource établit un seul ACC pour tous les sujets du système. Chaque contrat intelligent implémente les trois fonctions de validation (authentification, validation statique et dynamique) expliquées dans la section précédente.

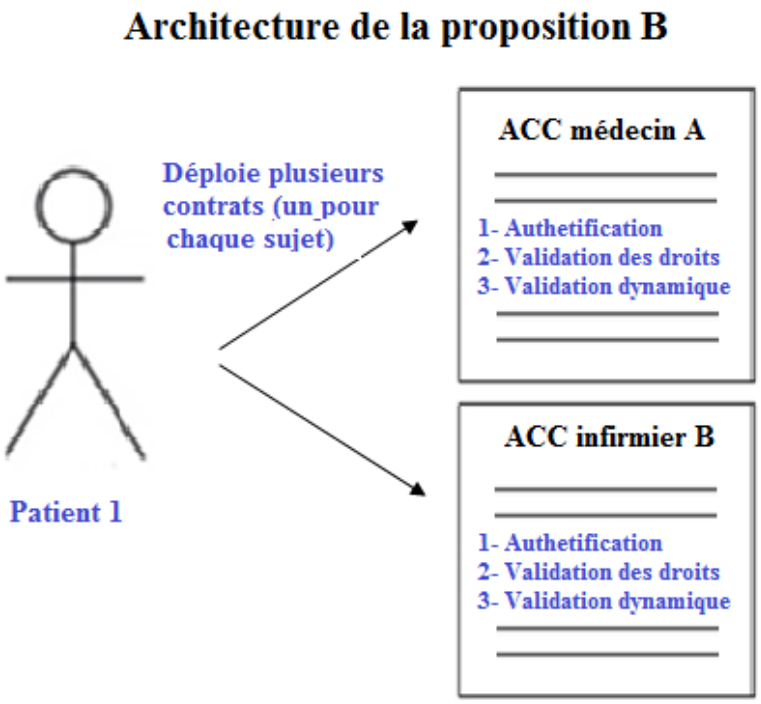

Figure III.6. Schéma de la proposition B [90].

Le diagramme de séquence représenté par la Figure III.7, montre les étapes du contrôle d'accès de la proposition B. En premier lieu, le patient définit les politiques d'accès à ses ressources dans plusieurs contrats, un ACC pour chaque sujet (un pour le médecin A, un autre pour l'infirmier B…) puis les déploie dans la BC. Ensuite lorsque le Médecin A soumet une demande d'accès, il récupère l'ACC qui contient ses droits d'accès (ACC médecin A : nous supposons qu'il connait son adresse), ce qui déclenche l'exécution de la méthode Accesscontrol(), il passe par les trois étapes de validation. Si une erreur est détectée à une quelconque étape, l'accès est refusé, sinon l'utilisateur se voit attribué l'accès. Il faut noter que dans cette proposition, lorsqu'un autre sujet par exemple l'infirmier B demande accès aux ressources du patient A, il ne passe pas par le même contrat que le médecin A, il récupère l'ACC qui contient ses droits d'accès (ACC infirmier B).

Lors de la création des contrats intelligents, le propriétaire de l'objet (le patient) devra définir les politiques d'accès pour ses ressources. Puisque l'ACC est déployé pour un couple objet-sujet, il n'a pas besoin de définir l'adresse du sujet comme dans la proposition A, il n'aura qu'à renseigner la ressource à laquelle le sujet peut ou ne peut pas accéder, l'action qu'il peut effectuer sur cette ressource, l'autorisation, MinInt et Limite, comme illustré dans le Tableau III.4.

La liste de mauvaise conduite implémentée dans chaque ACC, ne contient pas l'adresse du sujet comme définit dans le tableau III.2, elle contient juste les ressources, action, temps de mauvaise conduite, pénalité et temps de déblocage car chaque ACC est propre à un seul sujet.

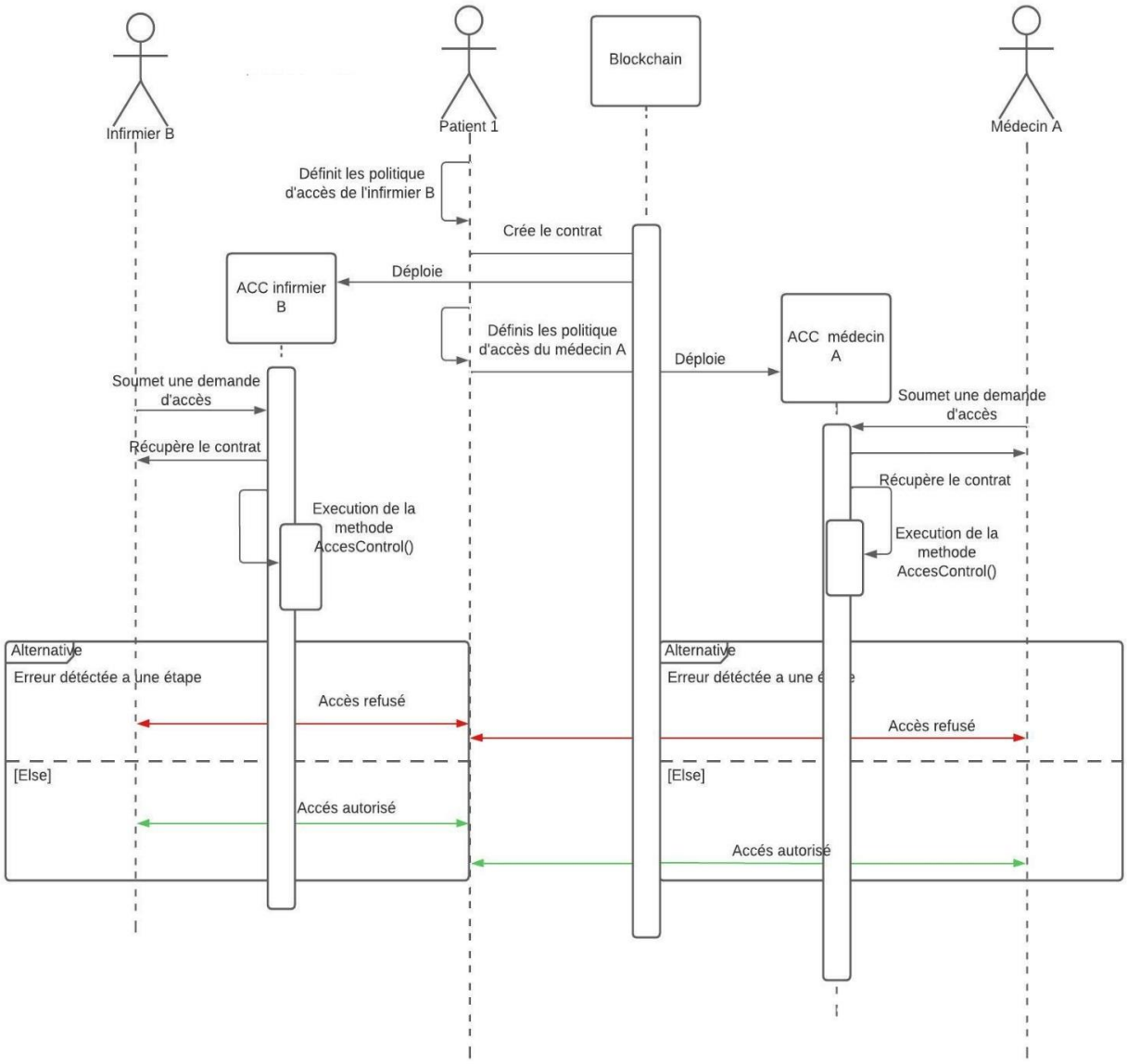

Figure III.7. Diagramme de séquence de la proposition B.

Ressource	Action	Permission	MinInt	Limite
Fichier A	Ecrire	Autoriser	1 min	2
Fichier A	Lire	Autoriser	30 secs	5
Programme B	Exécuter	Refuser	3 min	1

Tableau III.4 Liste des politiques d'accès d'un objet dans la proposition B.

III.4.7 Architecture de la proposition C

Le contrôle d'accès dans la proposition C consiste en plusieurs ACC, un MJC et un RC. Chaque propriétaire de ressource établit un seul ACC pour tous les sujets du système comme dans la proposition A, mais la différence avec la proposition A est que l'ACC implémente uniquement la méthode de validation des droits statiques, l'authentification est implémentée dans le RC, et la validation dynamique des droits est implémentée dans le MJC comme illustré sur la Figure III.8.

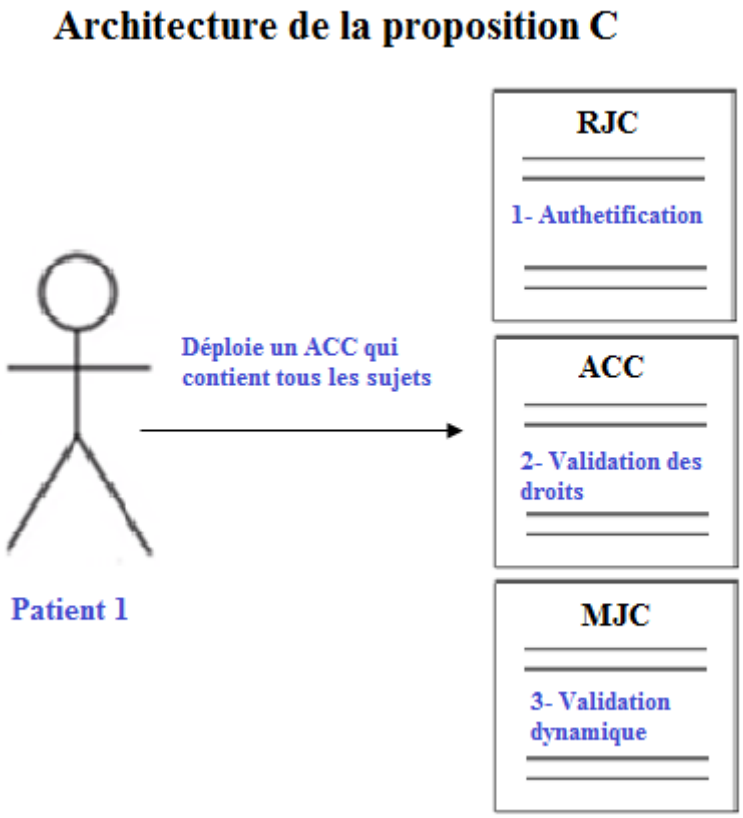

Figure III.8. Schéma de la proposition C [90].

Le diagramme de séquence représenté par la Figure III.9, montre les étapes du contrôle d'accès de la proposition C. En premier lieu, l'administrateur du système hospitalier déploie un contrat RC qui implémente la fonction d'authentification, et un contrat MJC qui implémente la fonction de validation dynamique. Ensuite, le patient définit les politiques d'accès à ses ressources dans un seul contrat ACC, puis le déploie dans la BC. Par la suite, lorsque le Médecin A soumet une demande d'accès, il récupère le ACC, ce qui déclenche l'exécution de la méthode Accesscontrol(), l'ACC déclenche le RC pour qu'il exécute l'authentification. Si le sujet n'est pas authentifié dans le système, le RC le signale à l'ACC et ce dernier va refuser l'accès, sinon l'ACC exécute la vérification des droits d'accès statique. Si le sujet n'a pas le droit d'y accéder, l'ACC va refuser l'accès sinon il déclenche le MJC pour qu'il exécute la vérification des mauvaises conduites. Si une

mauvaise conduite est détectée, le MJC le signale à l'ACC, qui à son tour va refuser l'accès, sinon l'ACC autorise l'accès. Il faut noter que dans cette proposition, lorsqu'un autre sujet par exemple l'infirmier B demande accès aux ressources du patient 1, il passe par le même contrat que le médecin A.

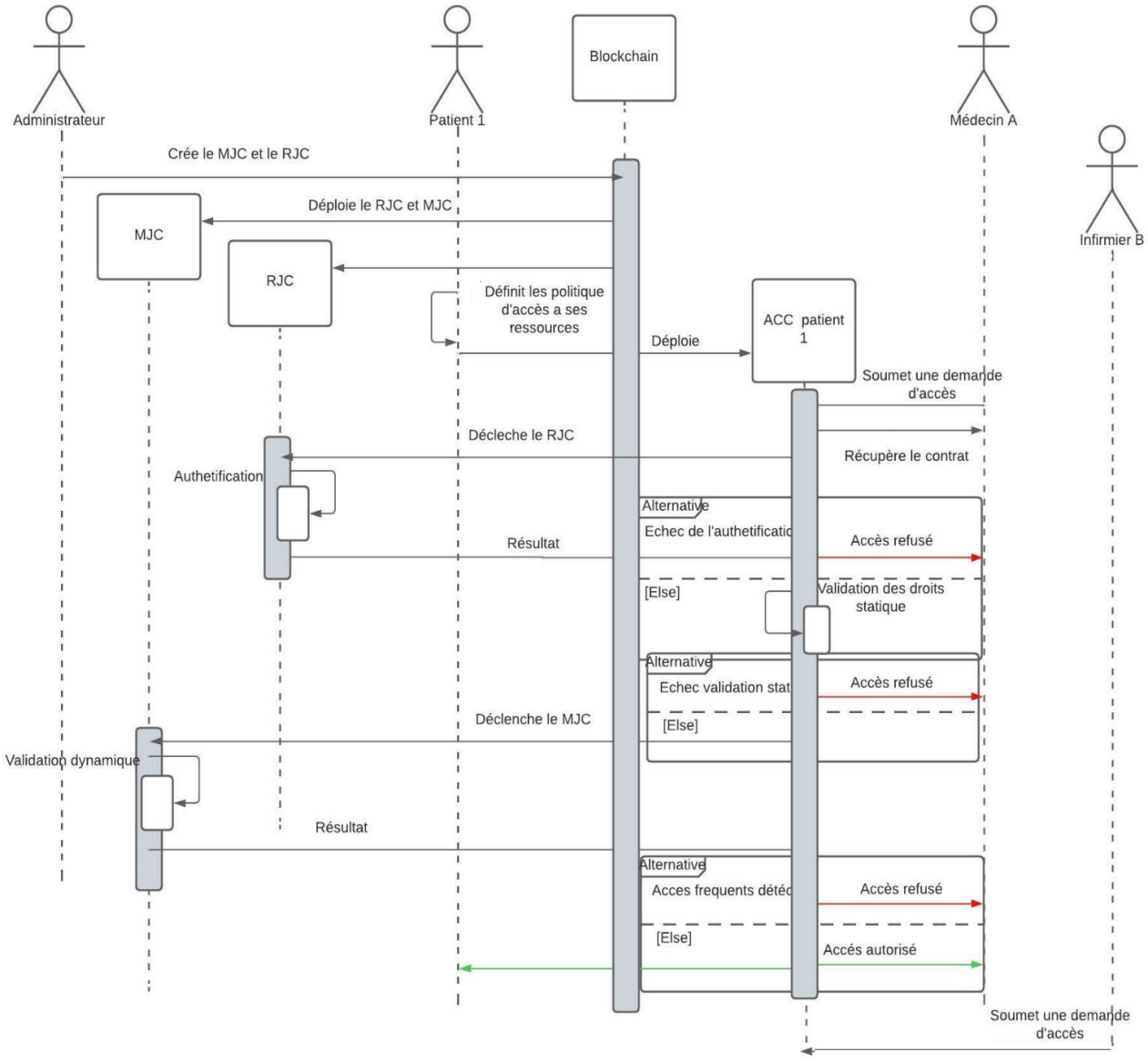

Figure III.9. Diagramme de séquence de la proposition C.

Lors de la création du ACC, le propriétaire de l'objet devra définir les politiques d'accès à sa ressource en renseignant l'adresse du sujet, le type de ressource, l'action, autorisation d'accès, MinInt et Limit comme dans la proposition A.

III.5 Architecture proposée du contrôle d'accès basé sur les NFT

III.5.1 Flux de travail du contrôle d'accès basé sur les NFT

Pour rajouter une couche de sécurité à l'architecture précédente (basée sur contrats intelligents), dans cette section, nous proposons de combiner les SC avec les NFT pour réaliser le contrôle d'accès. L'ACC utilise le concept de jeton NFT basé sur le standard ERC-721 (Ethereum Request for Comments) pour créer le jeton d'accès appelé (AcToken ou Access Token). Le AcToken est associé à une clé secrète, de sorte que seuls les utilisateurs pouvant prouver qu'ils possèdent cette clé peuvent utiliser le jeton. Le contrôle d'accès utilise l'architecture de la proposition A, en d'autres termes, chaque patient déploiera un ACC pour tous les sujets du système qui fournis des jetons d'accès selon les règles définies pour chaque ressource.

Comme illustré dans la Figure III.10, notre contrôle d'accès basé sur les NFT passe par les étapes expliquées ci-dessous :

- **Étape 1 (création des contrats intelligents) :** le propriétaire de la ressource crée un ACC pour tous les sujets du système, y définit les politiques d'accès, et le déploie dans la BC.
- **Étape 2 (soumettre la demande) :** le sujet soumet une demande d'accès à l'objet en envoyant les informations nécessaires dans une transaction au propriétaire de l'objet.
- **Étape 3 (demande de création de jeton) :** si le sujet n'a pas de jeton d'accès (cas de la première demande d'accès), le propriétaire de l'objet envoie une demande de création de jeton à l'ACC, sinon on passe à l'étape 6.
- **Étape 4 (génération du jeton) :** l'ACC génère le jeton d'accès (le AcToken) avec les politiques d'accès définit par le propriétaire de l'objet à l'étape 1.
- **Étape 5 (transmission du jeton) :** l'ACC transmet l'AcToken au sujet.
- **Étape 6 (vérification du jeton) :** le sujet utilise l'AcToken pour demander l'accès au nœud fog qui va effectuer une vérification de ce dernier.
- **Étape 7 (Résultat)** : le nœud fog envoie le résultat au sujet et au propriétaire de l'objet.

La vérification au niveau du nœud fog consiste à confirmer le propriétaire du jeton et les droits d'accès en interrogeant l'ACC.

Toute action qui se passe dans la BC comme le déploiement du ACC, la demande d'accès, la demande de création du jeton, la création du jeton, sa transmission, sa vérification ou encore l'envoie du résultat est enregistrée dans une transaction qui sera validée et scellée dans un bloc afin d'empêcher toute modification, et de faciliter la vérification et l'audit en cas de problèmes de sécurité.

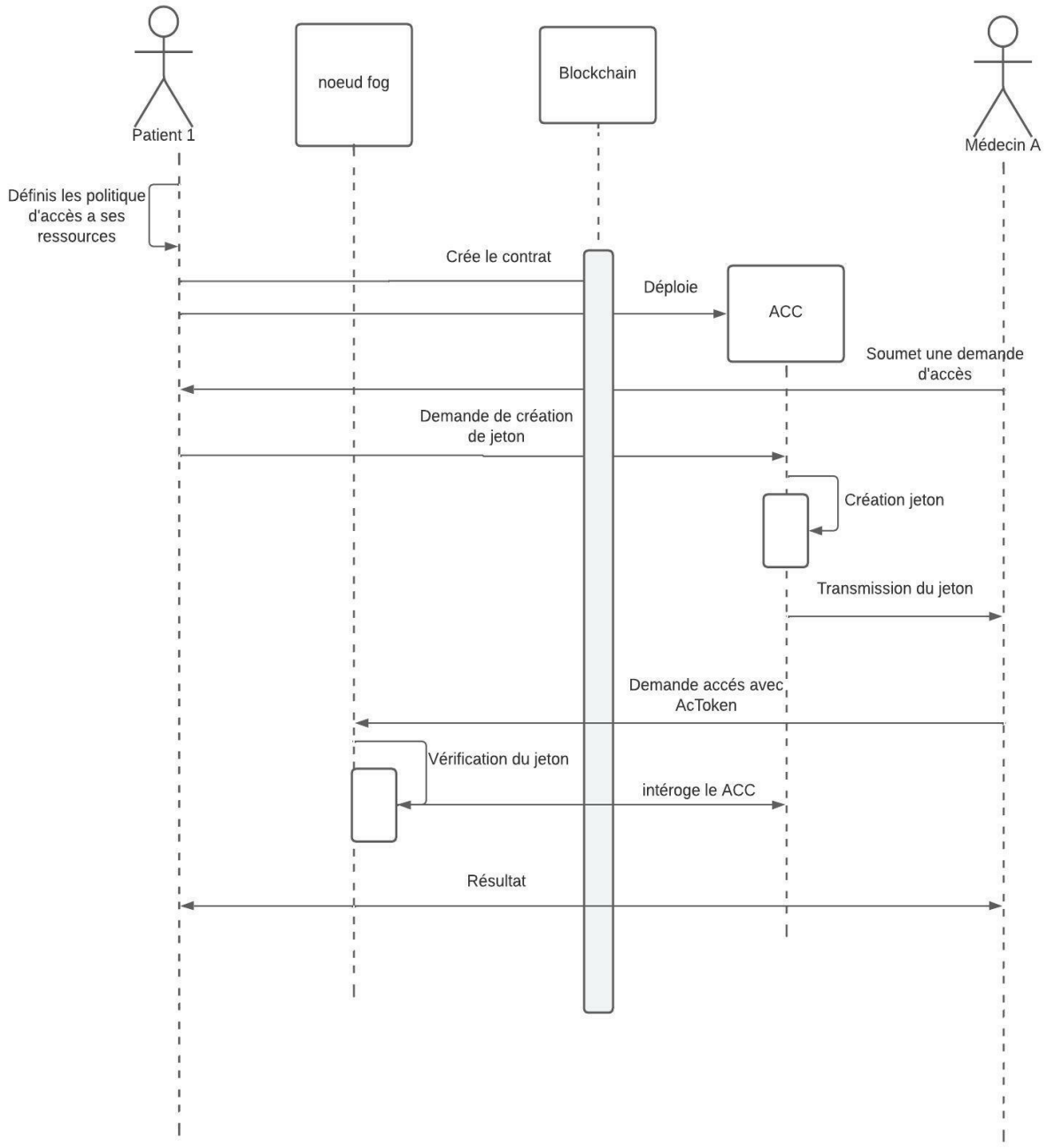

Figure III.10. Diagramme de séquence du contrôle d'accès basé sur NFT.

Dans cette architecture, seul le propriétaire du contrat peut créer de nouveaux jetons. Lorsqu'un jeton est créé, son identifiant et ses métadonnées sont spécifiés : ces deux propriétés sont en lecture seule et ne peuvent pas être modifiées. De plus, les propriétaires de jetons dans notre système ne peuvent pas transférer leurs jetons, la seule entité qui peut invoquer la méthode transferFrom expliquée ci-dessous est le propriétaire de l'ACC.

Tous les jetons basés sur le standard ERC-721 sont identifiés par un identifiant unique que nous appelons tokenid, et ils ne peuvent appartenir qu'à un seul utilisateur.

Afin de permettre la gestion dynamique des jetons, le modèle principal de l'ACC comprend un ensemble de fonctions de gestion, cela comprend :

- **CréerNFT() :** est utilisée pour générer de nouveaux jetons par le propriétaire de l'objet lorsqu'une nouvelle ressource est disponible ou une nouvelle politique est requise.
- **SupprimerNFT() :** permet la suppression des jetons d'accès lorsqu'une stratégie existante devient obsolète ou qu'une ressource n'est plus disponible.
- **URI-NFT() :** retourne les métadonnées du jeton.

En plus de ces fonctions, il y en a d'autres qui sont définies par défaut dans la norme ERC721 sur la quelle est basée notre ACC, et qui nous permettent de gérer les jetons, telles que :

- **Ownerof (tokenid) :** accepte en entrée un tokenid et renvoie l'adresse du propriétaire du jeton.
- **TransferFrom (from, to, tokenid) :** transfère un tokenid d'une adresse à une autre.
- **GetApproved (tokenid) :** récupère l'adresse du compte qui est autorisé à gérer le tokenid (autre que le propriétaire).

III.5.2 Structure du jeton AcToken

L'ACC utilise le concept du jeton NFT pour créer le Jeton AcToken. La structure du jeton, comme illustré dans la Figure III.11, est définie par les quatre champs décrits ci-dessous :

- **NFTId :** il s'agit d'un identifiant unique du jeton qui est composé du hachage de deux autres identifiants, l'identifiant du sujet et l'identifiant de l'objet.
- **AddSujet :** il s'agit de l'adresse du sujet qui veut accéder à l'objet, le détenteur du jeton.
- **Action :** il s'agit de la permission accordée au sujet (ex : lire, écrire…).
- **OwnerId :** ce champ gère la propriété du jeton et est également utilisé dans la délégation du jeton, différentes valeurs de addSujet et du OwnerId signifient que les droits d'accès du jeton ont été délégués.

L'approche recommandée par l'ERC pour associer un jeton avec certaines métadonnées via l'extension metadata, fournit une méthode qui mappe un tokenid à un URI où les métadonnées sont stockées (c'est-à-dire, tokenURI). Néanmoins, nous désapprouvons cette approche, car elle va à l'encontre du principe de décentralisation et des principes d'immutabilité de la BC, puisqu'avec cette approche, le fichier de métadonnées est stocké dans un emplacement centralisé, où il peut facilement être modifié sans qu'il soit possible de suivre ou même de détecter les changements. Pour cette raison, dans notre système, les métadonnées sont enregistrées dans des structures. Par

conséquent, dans notre système, la fonction URI-NFT est utilisée pour récupérer les métadonnées elles-mêmes et non un URI qui pointe vers les métadonnées.

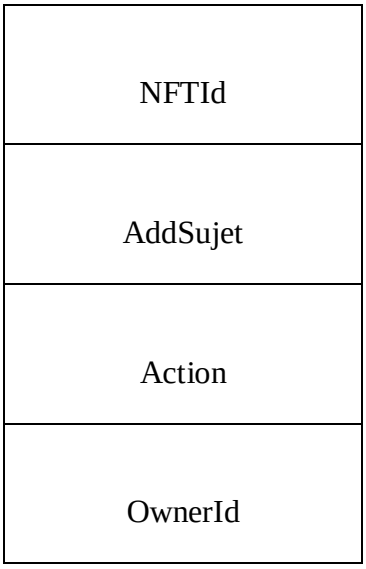

Figure III.11. Structure du jeton NFT.

III.6 Conclusion

Dans ce chapitre nous avons passé en revue les différentes architectures proposées, à commencer par une architecture des systèmes de soins de santé, ensuite trois architectures du contrôle d'accès basé sur les contrats intelligents et pour finir une architecture de contrôle d'accès basé sur le NFT. Dans le chapitre suivant nous présenterons les résultats obtenus et nous discuterons des avantages de toutes ces architectures proposées.

Chapitre IV : Implémentations réalisées et résultats obtenus

IV.1 Introduction

Afin de partager les données des patients en toute sécurité dans les systèmes de soins de santé, nous avons proposé trois schémas différents de contrôle d'accès basés sur les contrats intelligents de la blockchain, et un schéma basé sur les NFT.

Dans la première proposition du contrôle d'accès basé sur les contrats intelligents, chaque propriétaire d'objet déploie un seul contrat intelligent qui va gérer les demandes d'accès de tous les sujets du système. Tandis que dans la deuxième proposition, il déploie un contrat intelligent pour chaque sujet du système, chaque contrat va gérer les demandes d'accès d'un seul sujet, et dans la troisième proposition, il déploie un contrat intelligent pour gérer tous les sujets du système, tandis que l'administrateur du système en déploie deux autres. En d'autres termes cette proposition consiste en trois types de contrats contrairement aux deux premières qui consistent en un seul type de contrat. Dans le contrôle d'accès basé sur les NFT, le propriétaire de l'objet déploie un seul contrat intelligent qui va gérer les demandes d'accès de tous les sujets du système comme dans la première proposition du CA basé sur SC, la seule différence est que le droit d'accès est encapsulé dans un jeton unique propre au sujet.

Ce chapitre est organisé comme suit : dans la section 2, nous expliquons les algorithmes que nous avons utilisés pour chacune des quatre propositions. Nous présentons ensuite le matériel et les logiciels que nous avons utilisés afin de mener à bien nos expériences dans la section 3. Dans la section 4, nous expliquons comment nous avons implémenté chaque proposition ainsi que les interfaces sujet et propriétaire de l'objet. Les résultats obtenus sont présentés dans la section 5, ainsi qu'une analyse des performances. Dans la section 6, nous présentons une analyse de sécurité contre quelques attaques, et enfin nous concluons ce chapitre dans la section 7.

IV.2 Algorithmes proposés

IV.2.1 Algorithme de la proposition A

Lorsque l'ACC reçoit une demande d'accès, il exécute la méthode AccessControl() avec les informations reçues dans la demande d'accès. L'algorithme IV.1 représente la méthode Accesscontrol() de la proposition A.

Algorithme IV.1 Méthode Accesscontrol()

Entrée: sujet, ressource, action, temps
Sortie: résultat, pénalité
`Initialisation:` *pénalité* ← *0*,*résultat* ← *vrai*, ListePolitiqueSujet Politiques, ListEnregistrementSujet Enregistrement, ListeMauvaiseconduite-Sujet listeMc

```
1: R ← Enregistrement [sujet]
2: P ← Politiques [sujet][ressource][action]
3: M ← ListeMc [sujet]
4: if R == vrai then
5:     if M.tempsdeDéblocage ≤ temps then
6:         M.tempsdeDéblocage ← 0
7:         if P.permission == 'autoriser' then
8:             F ← temps - P.TDA
9:             if F ≤ P.MinInt then
10:                P.NAF ← P.NAF ++
11:                if P.NAF ≥ P.Limite then
12:                    Détection d'une mauvaise conduite MC
13:                    S ← M.taille+1
14:                    pénalité ← base*S
15:                    M.tempsdeDéblocage ← temps+pénalité
16:                    Ajouter MC a ListeMc
17:                    résultat ← faux
18:            else
19:                P.NAF ← 0
20: else
21:    P.TDA ← time
22: Déclencher l'évenement RetournerRésultatAccès(résultat, pénalité)
```

La méthode prend comme entrée l'adresse du sujet (sujet), la ressource à laquelle le sujet veut accéder, l'action qu'il veut effectuer et l'heure à laquelle il essaye d'y accéder (temps), et elle donne comme sortie le résultat de l'accès qui peut être soit autorisé (vrai) soit refusé (faux) et la pénalité qui est représentée par un temps de blocage, cette dernière est calculée en cas de détection de mauvaise conduite.

La ligne 4 de l'algorithme 1 représente l'authentification, la ligne 7 représente la validation des droits statiques, et les lignes de 9 à 16 représentent la validation dynamique.

L'algorithme IV.1 commence par initialiser le R avec l'enregistrement qui correspond au sujet (ligne 1), P avec les politiques d'accès qui correspondent au sujet, avec la ressource qu'il veut accéder et à l'action qu'il veut effectuer sur cette ressource (ligne 2), et le M avec la liste de mauvaises conduites qui correspond au sujet (ligne 3). Après avoir fini l'authentification (ligne 4),

l'ACC vérifie d'abord si le sujet n'est pas bloqué en comparant le temps de déblocage avec le temps d'accès, si le sujet n'est plus bloqué, on remet le temps de blocage à zéro (ligne 5), ensuite il vérifie si le sujet a le droit d'accéder à la ressource ou pas (ligne 6). Si le sujet a bien le droit d'accéder à la ressource, l'ACC vérifie ensuite si cet accès est un accès fréquent en calculant la différence entre l'heure de cet accès et l'heure du dernier accès (ligne 8) et en le comparant ensuite avec MinInt (ligne 9) déclaré par le sujet lors de l'ajout de la politique d'accès du sujet. Si ce dernier dépasse ce MinInt, il est considéré comme un accès fréquent, et le NAF (nombre d'accès fréquents) sera augmenté (ligne 10). Ensuite ce NAF est comparé à la limite d'accès fréquent définit par le sujet (ligne 11), si ça dépasse, il est considéré comme une mauvaise conduite (ligne 12), une pénalité qui dépend du nombre de mauvaises conduites effectuées par ce sujet (lignes 13-15) lui sera attribuée, et sera ajoutée à la liste de mauvaises conduites (ligne 16) et l'accès sera refusé (ligne 17), sinon l'accès sera autorisé (résultat est initialisé a vrai). Si cet accès n'est pas considéré comme un accès fréquent, le NAF sera remis à zéro (ligne 19), et pour finir le temps du dernier accès recevra la valeur du temps d'accès (ligne 21). L'événement RetournerRésultatAccès (résultat, pénalité) à la ligne 22 est utilisé pour renvoyer le résultat d'accès et la pénalité à la fois au sujet et au propriétaire de l'objet.

IV.2.2 Algorithme de la proposition B

Comme mentionné dans le chapitre III, la méthode Accesscontrol() est la même pour la proposition A et B, mais vu que dans la proposition B le contrat est déployé pour chaque sujet, les paramètres d'entrée et de sortie changent. La méthode n'a pas besoin de l'adresse du sujet car l'ACC la connait déjà vu qu'il a été déployé que pour cette adresse (pour le patient 1 par exemple).

L'algorithme IV. 2 représente la méthode Accesscontrol() de la proposition B.

Algorithme IV.2 Méthode Accesscontrol()

Entrée: ressource, action, temps
Sortie: résultat, pénalité
`Initialisation:` *pénalité* ← *0*,*résultat* ← *vrai*, AddresseSujet sujet, ListePolitiqueRessource Politiques, ListeEnregistrementSujet R, ListeMauvaiseconduiteRessource ListeMC

```
1: P ← Politiques [ressource][action]
2: M ← ListeMc [ressource]
3: if R == vrai then
4:     if M.tempsdeDéblocage ≤ temps then
5:         M.tempsdeDéblocage ← 0
6:         if P.permission == 'autoriser' then
7:             F ← temps - P.TDA
8:             if F ≤ P.MinInt then
9:                 P.NAF ← P.NAF ++
10:                if P.NAF ≥ P.Limite then
11:                    Détection d'une mauvaise conduite MC
12:                    S ← M.taille+1
13:                    pénalité ← base*S
14:                    M.tempsdeDéblocage ← temps+pénalité
15:                    Ajouter MC a ListeMc
16:                    résultat ← faux
17:            else
18:                P.NAF ← 0
19: else
20:    P.TDA ← temps
21:            Déclencher l'évenement RetournerRésultatAccès(résultat,
   pénalité)
```

L'algorithme IV.2 prend comme entrée la ressource à laquelle le sujet veut accéder, l'action qu'il veut effectuer et l'heure à laquelle il essaye d'y accéder (temps), et il donne comme sortie le résultat de l'accès qui peut être soit « autoriser » soit « refuser » (vrai ou faux respectivement) et la pénalité en cas de détection de mauvaise conduite qui est représentée par un temps de blocage.

La ligne 4 représente l'authentification, la ligne 7 représente la validation des droits statiques, et les lignes de 9 à 16 représentent la validation dynamique.

IV.2.3 Algorithmes de la proposition C

Trois algorithmes seront définis dans ce qui suit étant donné que la proposition C comporte trois types de contrats et que chaque contrat comporte une étape de validation.

Algorithme IV.3 Méthode Accesscontrol()

Entrée: sujet, ressource, action, temps
Sortie: résultat, pénalité
`Initialisation:` *pénalité* ← *0*,*résultat* ← *vrai*, ListePlitiqueSujet Politiques, InstanceRCavecSujet Enregistrement,InstanceMJCavecsujet juge

```
1:  R ← Enregistrement.ValidationSujet [sujet]
2:  P ← Politiques [sujet][ressource][action]
3:  if R == vrai then
4:      if juge.tempsdeDéblocage ≤ temps then
5:          juge.tempsdeDéblocage ← 0
6:          if P.permission == 'autoriser' then
7:              F ← temps - P.TDA
8:              if F ≤ P.MinInt then
9:                  P.NAF ← P.NAF ++
10:                 if P.NAF ≥ P.Limite then
11:                     Détection d'une mauvaise conduite MC
12:                     pénalité ← juge.jugemauvaiseconduite[sujet, ressource,
13:                     action, temps]
14:                     résultat ← faux
15:             else
16:                 P.NAF ← 0
17: else
18:     P.TDA ← temps
19:  Déclencher l'évenement RetournerRésultatAccès(résultat, pénalité)
```

L'algorithme IV.3 représente la méthode Accesscontrol() de la proposition C, il prend comme entrée l'adresse du sujet (sujet), la ressource à laquelle le sujet veut accéder, l'action qu'il veut effectuer et l'heure à laquelle il essaye d'y accéder (temps), et il donne comme sortie le résultat de l'accès qui peut être soit autoriser soit refuser (vrai ou faux respectivement) et la pénalité en cas de détection de mauvaise conduite qui est représentée par un temps de blocage. La ligne 3 représente l'authentification, la ligne 6 représente la validation des droits statiques, et les lignes de 8 à 12 représentent la validation dynamique.

Pour vérifier l'enregistrement du sujet, l'ACC appelle le RC pour qu'il exécute la méthode ValidationSujet (ligne 1) définit par l'algorithme IV.4, et quand il détecte une mauvaise conduite (ligne 11), il appelle le MJC pour qu'il exécute la méthode JugeMauvaiseconduite (ligne 12) définit par l'algorithme IV.5.

L'événement RetournerRésultatAccès (résultat, pénalité) à la ligne 19 est utilisé pour renvoyer le résultat d'accès et la pénalité à la fois au sujet et au propriétaire de l'objet.

Algorithme IV.4 Méthode ValidationSujet()

Entrée: sujet
Sortie: état
`Initialisation:` *état* ← *faux*, ListeEnregistrementSujet Enregistrement

```
1: R ← Enregistrement [sujet]
2: if R == vrai then
3:     état ← vrai
4: else
5:     état ← faux
```

L'algorithme IV.4 représente la méthode ValidationSujet () de la proposition C, il prend comme entrée l'adresse du sujet (sujet) et donne comme sortie l'état du sujet : vrai quand le sujet est enregistré dans le système hospitalier et la blockchain, et faux quand il ne l'est pas.

L'algorithme IV.4 commence par initialiser le R avec l'enregistrement du sujet (ligne 1), ensuite il vérifie que le sujet est bien enregistré (ligne 2), si c'est le cas, il affecte la valeur « vrai » a l'état (ligne 3), sinon il lui affecte la valeur « faux » (ligne 5).

Algorithme IV.5 Méthode JugeMauvaiseconduite

Entrée:sujet, ressource, action, temps
Sortie: pénalité
`Initialisation:` *pénalité* ← *0*, ListeMauvaiseconduiteSujet ListeMc

```
1: M ← ListeMc [sujet]
2: S ← M.taille+1
3: pénalité ← base*S
4: M.tempsdeDéblocage ← temps+pénalité
5: Ajouter MC a ListeMc =0
```

L'algorithme IV.5 représente la méthode JugeMauvaiseConduite () de la proposition C, il prend comme entrée l'adresse du sujet (sujet), la ressource à laquelle le sujet veut accéder, l'action qu'il veut effectuer et l'heure à laquelle il essaye d'y accéder (temps), et il donne comme sortie une pénalité. Cette méthode est appelée uniquement quand une mauvaise conduite est détectée.

L'algorithme IV.5 commence par initialiser le M avec la liste des mauvaises conduites du sujet (ligne 1) et le S avec la taille du la liste M (le nombre de mauvaises conduites de ce sujet) (ligne 2). Il calcule la pénalité (ligne 3), ensuite il met à jour le temps de déblocage du sujet (ligne 4), et enfin, cette mauvaise conduite est ajoutée dans la liste des mauvaises conduites du sujet (ligne 5).

IV.2.4 Algorithme de la proposition NFT

Algorithme IV.6 Méthode créerNFT

Entrée: NftId ,sujet, ressource, action
Sortie: résultat
Initialisation: *résultat* ← *faux*, ListeURIJeton URI

1: $R \leftarrow Enregistrement\ [sujet]$
2: **if** $R == vrai$ **then**
3: $safeMint(sujet, NftId)$
4: $URI[NftId] \leftarrow sujet,\ ressource,\ action$
5: $résultat \leftarrow vrai$
6: Déclencher l'évenement TransfererNFT(NftId, sujet)

L'algorithme IV.6 représente la méthode CréerNFT() de la proposition NFT. Il prend comme entrée l'identifiant du jeton (obtenu après la hachage de l'identifiant du sujet, et de l'identifiant de l'objet), l'adresse du sujet pour lequel le jeton est créé, la ressource à laquelle ce jeton permet l'accès, et enfin l'action que le sujet peut effectuer sur la ressource avec ce jeton, et il donne comme sortie le résultat de la création qui peut être soit vrai si le jeton a bien été créé ou faux si une erreur est survenue et que le jeton n'a pas pu être créé. En premier lieu, la méthode initialise le R avec l'enregistrement du sujet (ligne 1), ensuite elle vérifie si le sujet pour lequel elle va créer un jeton est bien enregistré dans le système (ligne 2), puis elle exécute la fonction safeMint (ligne 3) qui est une fonction interne de L'ERC721 qui permet de créer un jeton si ce dernier n'a pas encore été créé, par la suite les données telles que le sujet, la ressource et l'action sont ajoutées aux métadonnées du jeton (ligne 4). Le jeton est ensuite ajouté à la liste des jetons, et le résultat reçoit la valeur « vrai » (ligne 5). Dans le cas où l'enregistrement n'est pas valide, la valeur résultat ne change pas et reste à « faux » comme initialisée. L'évènement TransfererNFT (ligne 6) est utilisé pour renvoyer le résultat de la création du jeton au sujet et au propriétaire de l'objet.

IV.3 Matériels et logiciels utilisés

Modèle	Processeur	Système d'exploitation	Mémoire	Carte graphique
Dell Latitude E5440	Intel® Core™ I7-4600 CPU @ 2.10GHz 2.70GHz	Windows 7 professionnel 64 bits	6 Go	NVIDIA GeForce Gt 720M

Tableau IV. 1 Spécifications de l'appareil utilisé.

Le tableau IV.1 montre les caractéristiques de la machine que nous avons utilisée pour la réalisation de nos simulations.

Nous avons utilisé le protocole Ethereum BC pour créer notre BC privé car il est fortement axé sur les contrats intelligents. Nous avons installé le client Geth sur l'appareil, puis nous avons créé plusieurs nœuds puis pour chaque nœud nous avons créé plusieurs comptes afin de former une BC privée, chaque utilisateur ou participant de la blockchain a un compte. Notre machine agit comme mineur et déploie les contrats en envoyant des transactions au réseau BC.

Nous avons utilisé Remix [91] pour compiler le code du contrat intelligent (Algorithmes IV 1-6) et nous l'avons connecté en local avec la BC privée créée pour déployer les contrats.
Nous avons utilisé JavaScript Ethereum pour créer l'interface du sujet et du propriétaire de l'objet. Nous avons également utilisé web3.js [92], une API JavaScript Ethereum du côté du sujet et du propriétaire de l'objet pour communiquer avec le client geth (nœud local) correspondant via des connexions HTTPS pour maintenir leurs divers événements.

IV.4 Implémentations réalisées

Dans cette section, nous allons présenter l'implémentation des algorithmes déployés pour nos quatre propositions ainsi que l'implémentation de l'interface coté sujet et propriétaire de l'objet.
A noter que le code de l'implémentation est disponible dans [93].

IV.4.1 Proposition A

Dans l'implémentation de l'ACC de la proposition A, nous avons utilisé une structure (struct) avec un tableau dynamique pour stocker la liste des mauvaises conduites (tableau III.2), et nous avons utilisé des structures pour stocker les champs d'enregistrement du sujet, et la liste des politiques (tableau III.3).

Un mappage (mapping) a été appliqué sur la première structure à partir des champs d'adresse du sujet afin de construire la liste des enregistrements des sujets. En d'autres termes, nous avons lié chaque adresse avec son enregistrement. Un mappage tridimensionnel (à 3 clés) a été appliqué sur la deuxième structure à partir des champs d'adresse du sujet (clé primaire), de la ressource (clé secondaire) et de l'action (clé tertiaire) pour construire la liste des politiques du sujet. En d'autres termes, nous avons lié chaque action de chaque ressource de chaque adresse avec sa liste de politiques. Enfin, un mappage a été appliqué sur la troisième structure du champ de l'adresse du sujet pour construire la SubjectMisconductList. En d'autres termes, nous avons lié chaque adresse avec sa liste de mauvaises conduites.

IV.4.2 Proposition B

Dans l'implémentation de l'ACC de la proposition B, nous avons utilisé la même implémentation que dans la proposition A pour construire la liste d'enregistrement du sujet. Nous avons utilisé une structure pour stocker les champs de politiques comme dans la proposition A, mais nous n'avons appliqué qu'un mappage bidimensionnel (à deux clés) à partir des champs de ressource (clé primaire) et d'action (clé secondaire) pour construire la liste de politiques du sujet. Nous n'avons pas besoin de préciser l'adresse du sujet car elle est déjà connue par l'ACC vu qu'il a été déployé que pour cette adresse. Enfin, nous avons utilisé une structure avec un tableau dynamique pour stocker les enregistrements de mauvaise conduite comme dans la proposition A sauf que nous avons appliqué un mappage du champ de ressource (et non de l'adresse du sujet comme dans la proposition A) à cette structure pour construire ResourceMisconductList.

IV.4.3 Proposition C

La même implémentation que la proposition A a été utilisée, sauf que la méthode ValidationSujet () est implémentée dans le RC, la méthode de Accesscontrol () est implémentée dans l'ACC et la JugeMauvaiseConduite () est implémenté dans le MJC.

IV.4.4 Proposition NFT

Nous avons utilisé le contrat intelligent ERC-721 OpenZeppelin [94] pour implémenter notre ACC pour accélérer le processus. Les contrats OpenZeppelin sont stables et ont fait l'objet d'audits de sécurité par des sources indépendantes. Par conséquent, il est plus sûr d'utiliser un contrat intelligent OpenZeppelin que de coder un contrat ERC-721 de zéro. Cette approche peut aider à éviter ou à limiter certaines vulnérabilités de sécurité.

Nous avons aussi utilisé une structure pour stocker les champs d'enregistrement de l'URI (NftId, sujet, ressource, action) et nous avons appliqué un mappage sur cette structure à partir du champ NftId afin de construire l'URI du jeton.

IV.4.5 Interface sujet et propriétaire de l'objet

Pour créer les interfaces sujet et propriétaire de l'objet, nous avons utilisé JavaScript. La communication avec les différents nœuds se fait en mode script. Le sujet appelle l'ACC, exécute la méthode accessControl, puis l'événement RetournerRésultatAccès() renvoie les résultats d'accès au sujet et à l'objet.

IV.5 Résultats obtenus et discussions

Dans cette section, nous évaluons les performances de nos trois propositions basées sur SC, qui dépendent d'une part du coût du déploiement du contrat, du coût de l'exécution des fonctions et des réponses, et d'autre part de la latence de réponse aux demandes d'accès. Nous évaluons aussi les performances de la proposition NFT en termes de coût de création de jeton. Il faut noter que nous avons utilisé le consensus PoW, qui utilise une méthode de validation compétitive pour confirmer les transactions et ajouter de nouveaux blocs à la BC. Pour un système réel, nous conseillons d'utiliser des consensus à faible coût tels que PoS ou DPoS qui utilisent des mineurs sélectionnés au hasard pour valider les transactions, et consomment ainsi moins de gaz.

IV.5.1 Résultats obtenus

Sur la base du code, du matériel et du logiciel présentés ci-dessus, nous avons mené des expérimentations pour montrer la faisabilité du contrôle d'accès proposé. Nous avons ajouté une politique à l'ACC avec "MinInt" défini sur 100 secondes, une "Limite" définie sur 2 et une "base" définie sur 30.

Les deux figures IV.1 et IV.2 montrent les résultats d'accès obtenus au niveau du sujet et de l'objet respectivement. Comme nous pouvons le remarquer, l'adresse du contrat, le numéro de la transaction, le numéro de bloc, le hash du bloc, le temps, le message d'erreur, et le résultat sont les mêmes des deux côtés. Le tableau IV.2 définit les paramètres indiqués dans les figures IV.2 et IV.3.

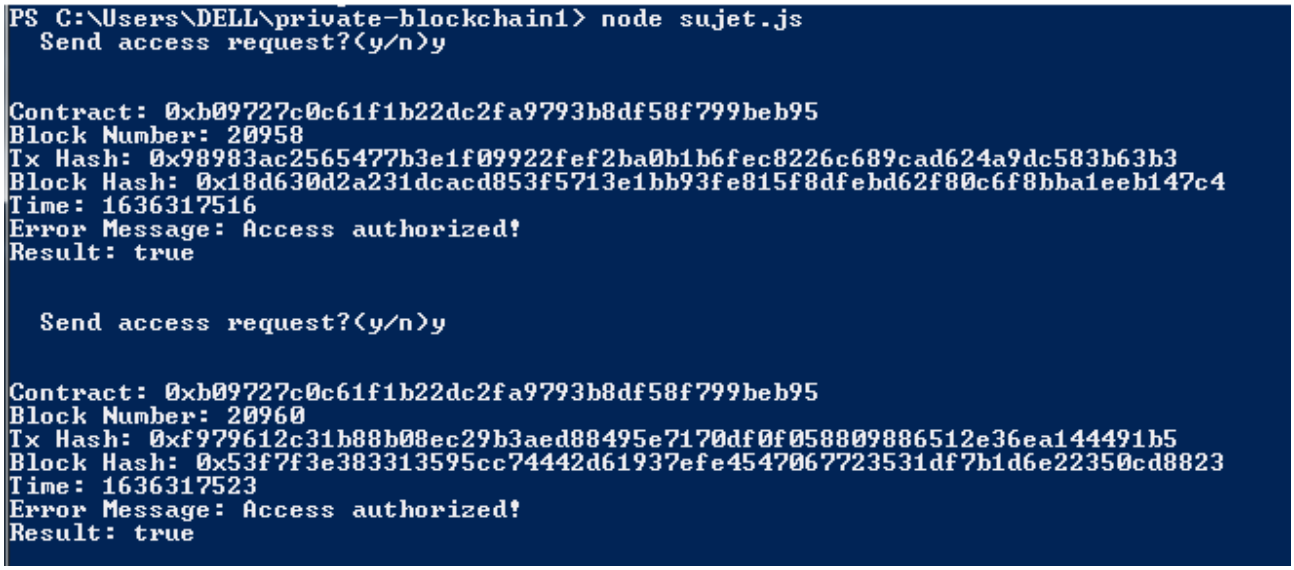

Figure IV.1. Résultat de l'accès coté sujet.

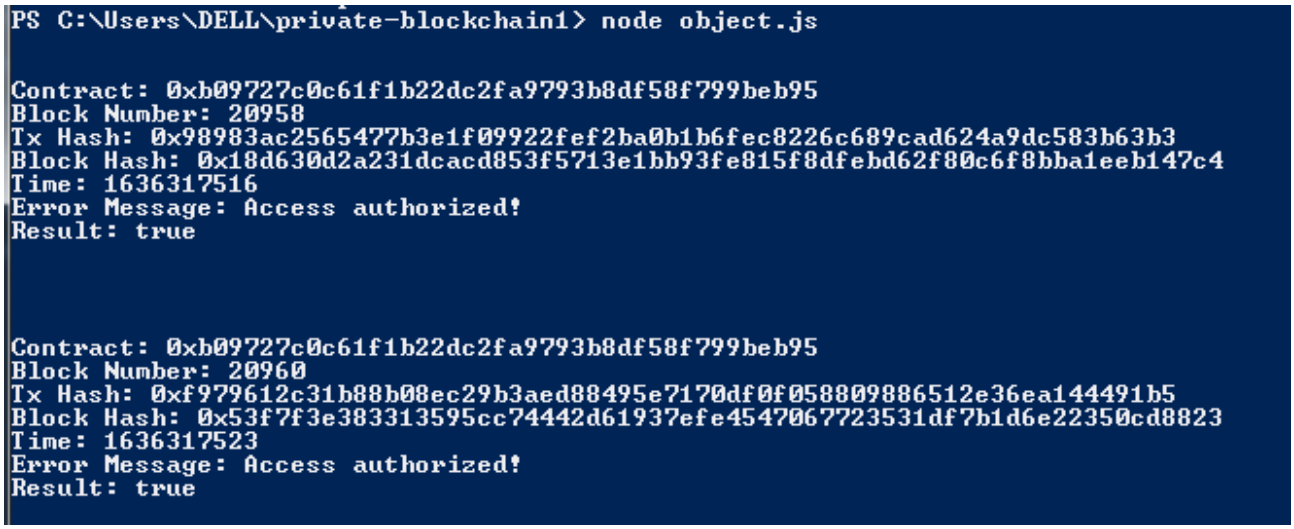

Figure IV.2. Résultat de l'accès coté propriétaire de l'objet.

Contract	Adresse du contrat
Block number	Numéro du bloc ou la transaction a été scellée
Tx hash	Hash de la transaction qui contient la demande d'accès
Block hash	Hash du bloc ou la transaction est scellée
Time	Horodatage (en secondes) quand la demande d'accès est envoyée
Error message	Message qui affiche le résultat de la demande d'accès
Result	Vrai si la validation a réussi, sinon faux

Tableau IV.2 Paramètres clés de l'interface de contrôle d'accès.

Les résultats de la figure IV.3 représentent le refus d'accès coté sujet en cas d'échec de l'authentification du sujet qui demande l'accès (échec de l'étape 1 du processus de validation). Dans ce cas, le message « Validation failure ! Invalid requester » ou « Échec de la validation ! Demandeur invalide » s'affiche et le résultat de l'accès est affiché comme « false ou faux ».

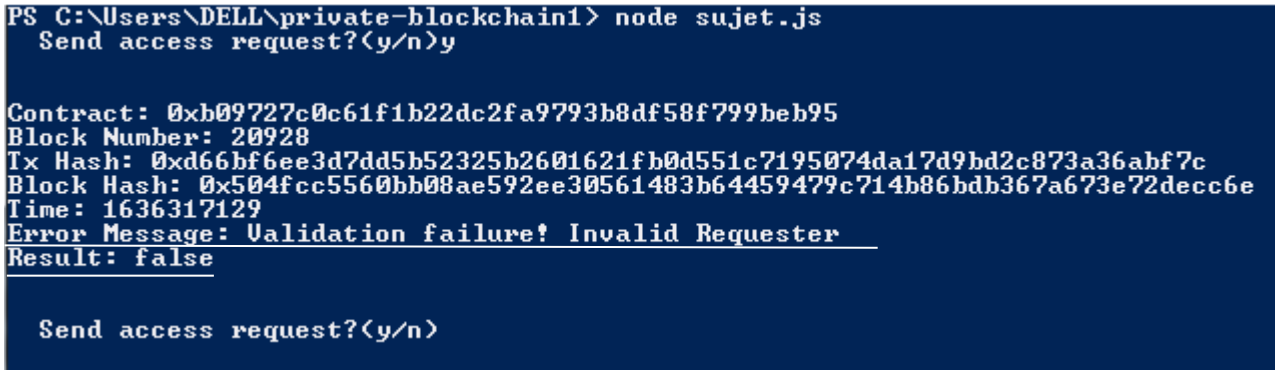

Figure IV.3. Echec de l'authentification.

Les résultats de la figure IV.4 représentent le refus d'accès en cas d'échec de la validation statique (étape 2 de la méthode de control d'accès). Dans ce cas, le message « Static check failed ! ! » ou « La vérification statique a échoué !! » s'affiche et le résultat de l'accès est affiché comme « false » ou « faux ».

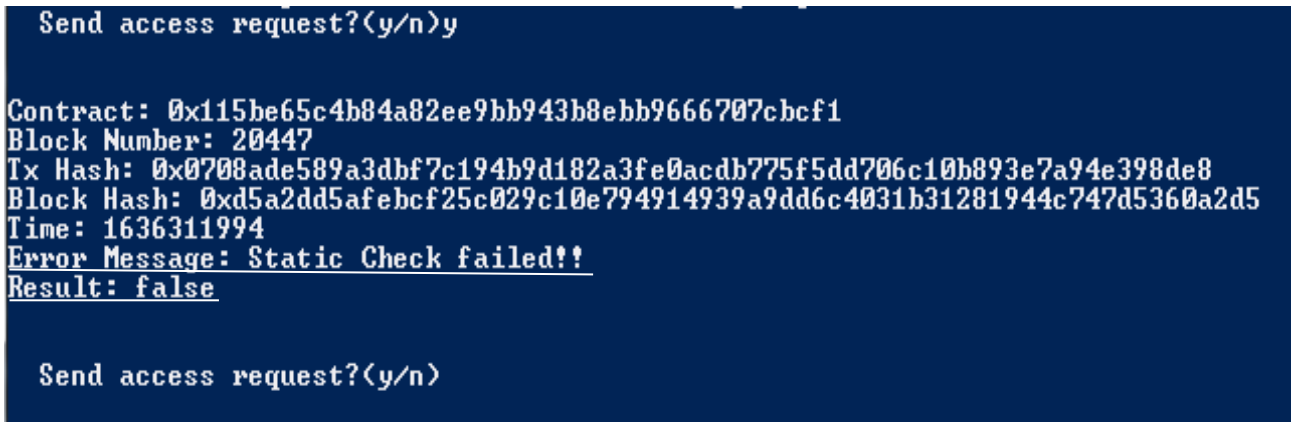

Figure IV.4. Echec de la validation des droits statiques.

Les deux premières lignes soulignées dans la figure IV.5 montrent les résultats d'une tentative d'accès coté sujet. Puisque le sujet est autorisé à accéder à la ressource, le message « Access autorized ! » ou « Accès autorisé ! » est apparu. De plus, lors de la troisième tentative, le NAF a dépassé la limite (3 > 2), donc une faute a été détectée. Le message « Misbehaviour detected ! Too frequent access » ou « Mauvaise conduite détectée ! Accès trop fréquent » est apparu et la requête a été bloquée pendant une durée de 30 secondes puisqu'il s'agit de la première inconduite (la liste des mauvaises conduites est vide).

```
PS C:\Users\DELL\private-blockchain1> node sujet.js
  Send access request?(y/n)y


Contract: 0xb09727c0c61f1b22dc2fa9793b8df58f799beb95
Block Number: 20958
Tx Hash: 0x98983ac2565477b3e1f09922fef2ba0b1b6fec8226c689cad624a9dc583b63b3
Block Hash: 0x18d630d2a231dcacd853f5713e1bb93fe815f8dfebd62f80c6f8bba1eeb147c4
Time: 1636317516
Error Message: Access authorized!
Result: true


  Send access request?(y/n)y


Contract: 0xb09727c0c61f1b22dc2fa9793b8df58f799beb95
Block Number: 20960
Tx Hash: 0xf979612c31b88b08ec29b3aed88495e7170df0f058809886512e36ea144491b5
Block Hash: 0x53f7f3e383313595cc74442d61937efe4547067723531df7b1d6e22350cd8823
Time: 1636317523
Error Message: Access authorized!
Result: true


  Send access request?(y/n)y


Contract: 0xb09727c0c61f1b22dc2fa9793b8df58f799beb95
Block Number: 20962
Tx Hash: 0x0120e63fed8efd38b5f9d15c8d3bb91fc2fd4e5d50aac96b5a3134982b03ef2e
Block Hash: 0x9a15579a04bd2e2926350a3eed0a429d3066ccc131f13f3006b5973ee115ce89
Time: 1636317531
Error Message: Misbehavior detected! too frequent access
Result: false
Requests are blocked for 30 seconds!
```

Figure IV.5. Accès accordé et refusé par le sujet.

Dans les deux figures IV.6 et IV.7, nous pouvons voir que le sujet n'a pas réussi à accéder à l'objet (résultat : false ou faux) en raison de son accès fréquent (Message d'erreur : Misbehaviour detected, too frequent access) et puisqu'il s'agit du troisième et du sixième accès fréquent du sujet, il est bloqué respectivement pendant 90 et 180 secondes. Les fonctions (1) et (2) montrent comment ces pénalités sont calculées.

$$Pénalité1 = 3 * 30 = 90\ secondes \qquad (1)$$

$$Pénalité2 = 6 * 30 = 180\ secondes \qquad (2)$$

```
  Send access request?(y/n)y


Contract: 0xb09727c0c61f1b22dc2fa9793b8df58f799beb95
Block Number: 20440
Tx Hash: 0x8531594180435fd0141e095e787a16082563ee79ee885e692cd0694f51ff9c08
Block Hash: 0x2b2eb6375d00ef93583ab8944c47cf4b2149c4c8aff3ffc3394239bcf4c9eb25
Time: 1636323372
Error Message: Misbehavior detected! too frequent access
Result: false
Requests are blocked for 90 seconds!


  Send access request?(y/n)
```

Figure IV.6. Résultats d'accès après trois mauvaises conduites détectées.

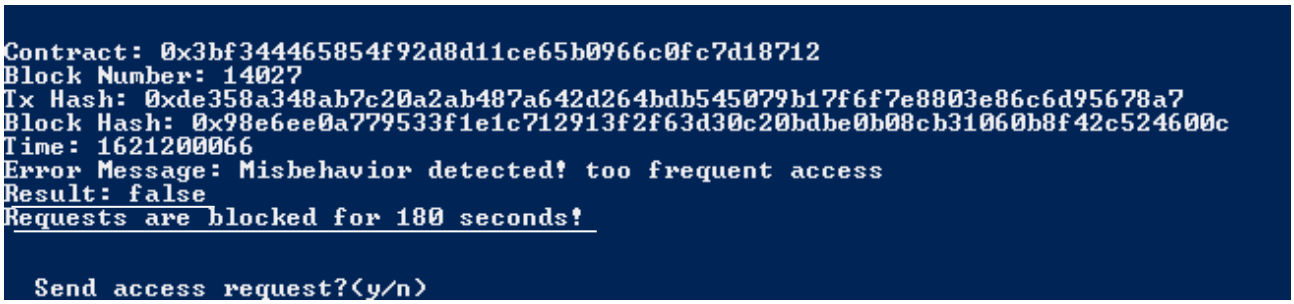

Figure IV.7. Résultats d'accès après six mauvaises conduites détectées.

IV.5.2 Coût de transaction associé au déploiement des contrats

Le tableau IV.3 montre le nombre de contrats déployés dans les différentes propositions en supposant qu'il y ait cinq sujets dans le système. Par exemple, dans [78], pour un patient du système, cinq ACC, un JC et un RC sont déployés, et pour deux patients, dix ACC, un JC et un RC sont déployés. Dans [74], le sujet déploie un contrat avec chaque autorité d'attribut, et chaque autorité d'attribut déploie un contrat avec le sujet. Dans le tableau IV.3, nous avons pris comme exemple deux attributs autorités. Comme illustré, la proposition A est celle qui déploie le moins de contrats, suivie de la proposition C, puis de la proposition B. Notons que la proposition NFT utilise le même principe que la proposition A en termes de déploiement de contrats.

Dans le tableau IV.3, nous n'avons pas pris en considération les travaux [72, 73, 77] où ils déploient un seul contrat dans tout le système, car cela prive le propriétaire de l'objet du contrôle de ses données. En effet, cela va à l'encontre de notre proposition. Nous n'avons pas non plus considéré le travail réalisé dans [75] car il ne mentionne pas comment ni par qui le contrat de contrôle d'accès est créé et donc nous ne pouvons pas connaitre le nombre de SC déployés.

Nos propositions vs quatre travaux connexes	1 patient	2 patients	N patients
[78]	7	12	5*N + 2
[79]	8	13	5*N + 3
[76]	6	12	6*N
[74]	30	40	10*N + 20
Proposition A	1	2	N
Proposition B	5	10	5*N
Proposition C	3	4	N + 2
Proposition NFT	1	2	N

Tableau IV.3 Nombre de contrat déployés pour cinq sujets.

Afin de déployer un contrat ou exécuter une fonction dans la BC, des frais sont requis. La plate-forme Ethereum BC désigne ces frais sous la forme d'une unité appelée gas.

Le tableau IV.4 représente le gas consommé par un patient pour déployer les différents contrats des trois propositions. Sur la base des résultats obtenus, nous pouvons remarquer que la proposition B est celle qui consomme le moins de gas. Cependant, le patient doit déployer un contrat pour chaque sujet dans le système, il va donc déployer N contrats pour N sujets alors que dans la proposition A, le patient va déployer 1 contrat pour N sujets.

	Proposition A	**Proposition B**	**Proposition C**		
Contrat	ACC	ACC	ACC	RJC	MJC
Coût	2879606	1663270	2678388	257756	499857

Tableau IV.4 Coût de déploiement des contrats.

Afin d'évaluer le coût de déploiement des contrats dans nos propositions, nous avons considéré un système avec 2 patients et 5 sujets. La figure IV.8 illustre les résultats obtenus. Par exemple, dans la proposition C, nous devons déployer 2 ACC, un MJC et un RC résultant en une consommation de gas de 6114389. Comme nous pouvons le voir, la proposition A est celle qui consomme le moins de gas suivie des propositions C et B, qui restent meilleures que les résultats obtenus dans [74, 76, 79]. Dans la figure IV.8, nous n'avons pas considéré les travaux réalisés dans [75, 77, 78, 80] car ils ne mentionnent pas les couts de déploiement de leurs contrats déployés.

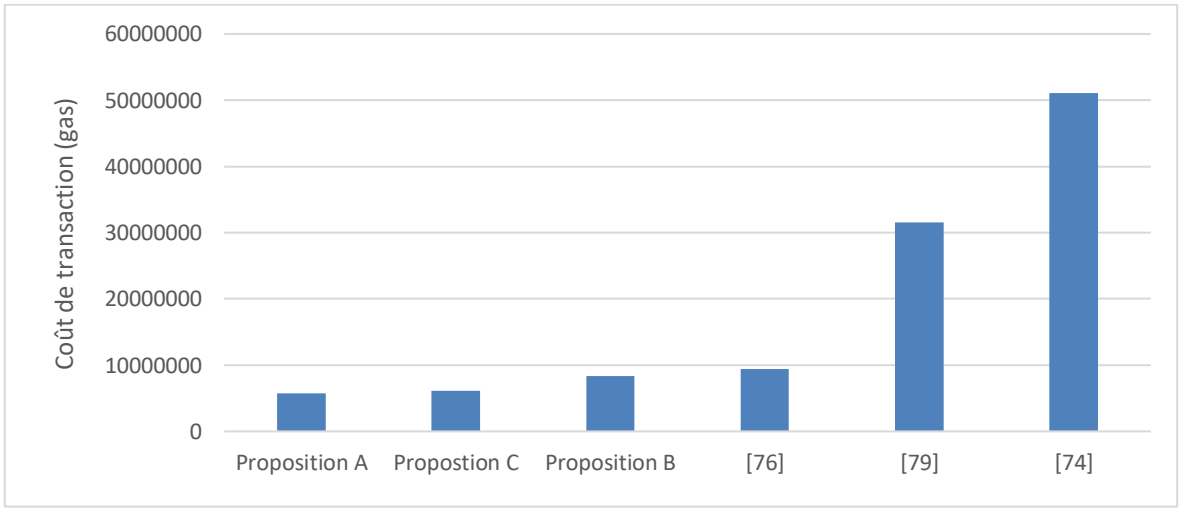

Figure IV.8. Coût de déploiement des contrats.

Nous pouvons donc conclure que la proposition A est la mieux adaptées aux systèmes de soins de santé en termes de coût de transaction associés au déploiement des contrats.

IV.5.3 Coût de transaction associé à l'exécution des fonctions

Le coût de transaction associé à l'exécution des fonctions des trois propositions est représenté sur la figure IV.9. Nous avons calculé la moyenne de 100 mesures de coût de transaction pour avoir la valeur finale. Nous pouvons remarquer qu'il n'y a pas de grande différence entre les trois propositions. Toutefois, la proposition B a obtenu de meilleurs résultats en termes de coût de transaction de l'exécution des fonctions.

Nous pouvons donc conclure que la propositions B est la mieux adaptées aux systèmes de soins de santé en termes de coût de transaction associés à l'exécution des fonctions.

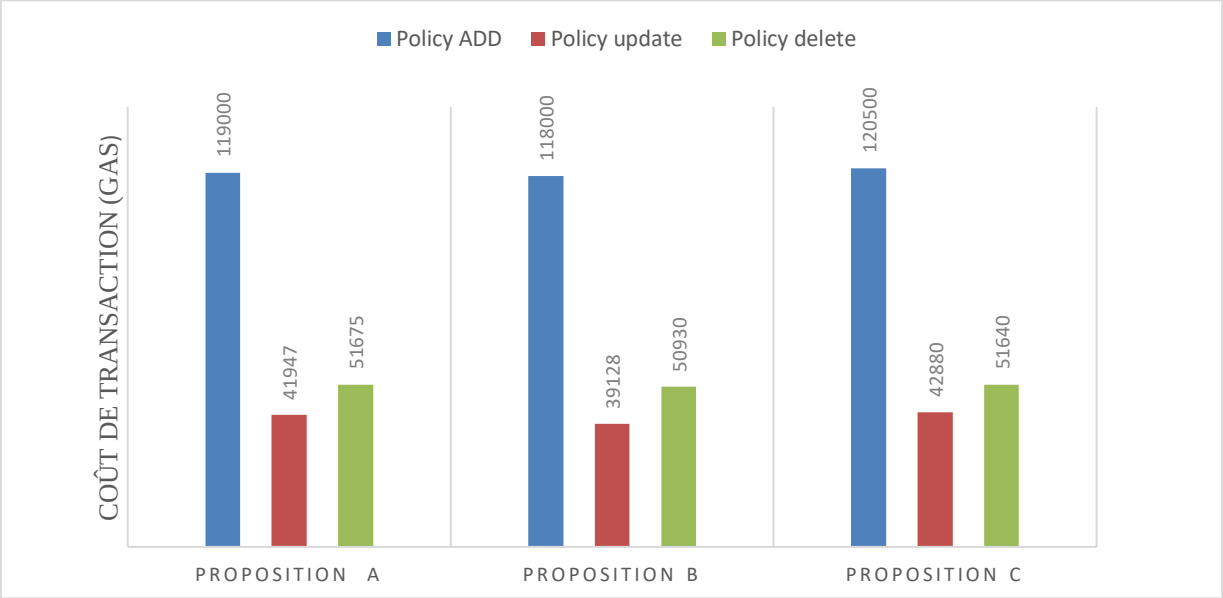

Figure IV.9. Coût de transaction associé à l'exécution des fonctions.

Ensuite le coût de transaction associé à l'exécution de la fonction de la création du jeton (créerNFT) de la proposition NFT est représenté sur la figure IV.10. Comme nous pouvons le constater notre proposition a obtenu de meilleurs résultats par rapport aux travaux connexes.

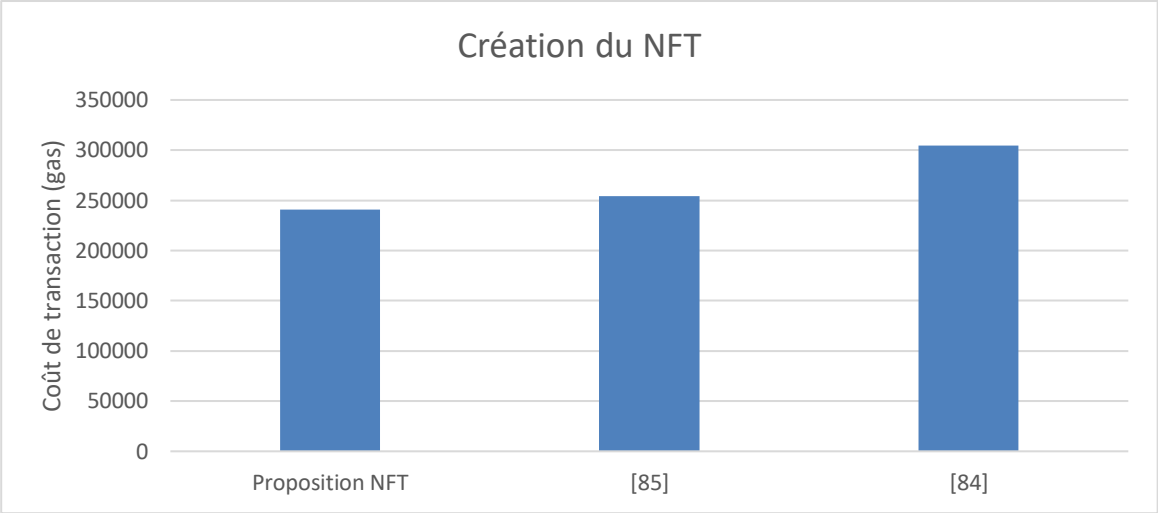

Figure IV.10. Coût de transaction associé à la création du NFT.

IV.5.4 Coût de transactions associées aux différentes réponses

Dans cette section, nous avons fait une comparaison entre nos trois propositions en termes de coût de transaction des différentes réponses. Les figures IV.11, IV.12, IV.13 et IV.14 représentent le coût de transaction des réponses Accès autorisé, mauvaise conduite détectée, demande rejetée, et échec de l'authentification respectivement. L'authentification et la mauvaise conduite ne sont pas comparées par rapport au travail réalisé dans [76] car les auteurs n'ont pas traité ces cas-là dans leur contrôle d'accès.

Nous remarquons que le gas consommé par A est meilleur que celui consommé par B pour : *Accès autorisé et Demande rejetée*, et que le gas consommé par B est meilleur que celui consommé par A pour : *Mauvaise conduite détectée et Echec de l'authentification*. De plus, les résultats obtenus dans les propositions A et B sont meilleurs que ceux de la proposition C.

Nous avons ensuite comparé notre travail avec les travaux réalisés dans [76] et [79]. Nous pouvons remarquer que nous avons obtenu de meilleurs résultats en termes d'accès autorisé, de mauvaise conduite détectée et d'échec de l'authentification dans les propositions A et B que dans les travaux connexes et cela est dû au fait que nous n'utilisons qu'un seul type de contrat intelligent dans les deux propositions pour effectuer l'authentification, la validation des droits statiques et la validation dynamique. Nous pouvons aussi remarquer que dans [76], les auteurs ont obtenu de meilleurs résultats dans la réponse 'demande rejetée'. Mais notez qu'ils n'ont donné que le cas d'échec de la première condition, si c'était le cas de la nième condition ils auraient consommé beaucoup plus.

Nous pouvons donc conclure que les propositions A et B sont les mieux adaptées aux systèmes de soins de santé en termes de coût de transaction associés aux réponses.

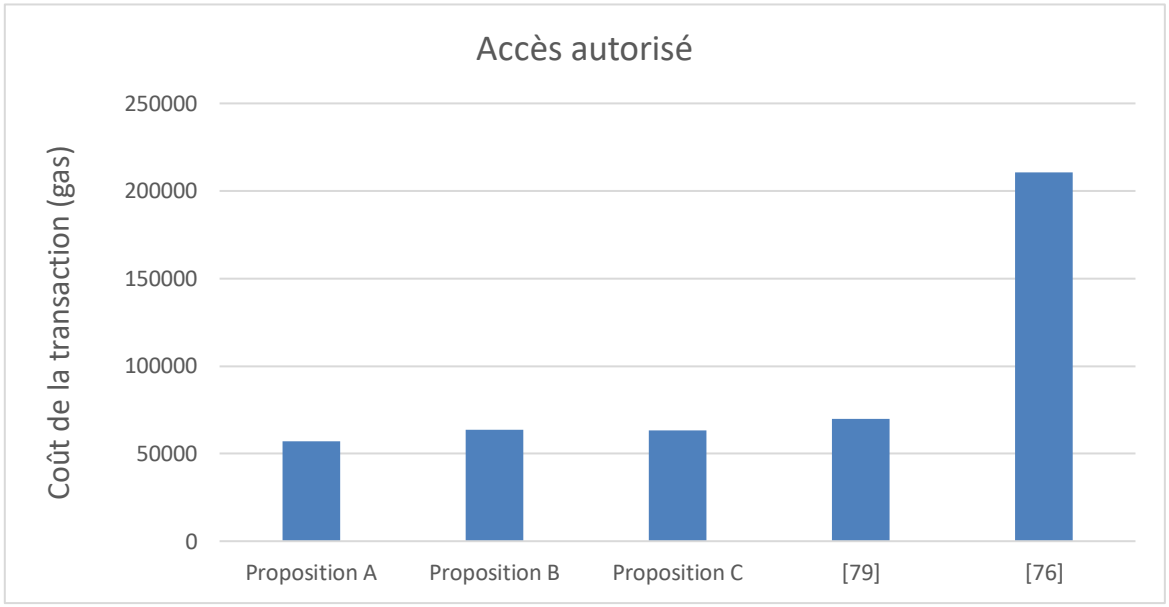

Figure IV.11. Comparaison du coût de transaction de la réponse « Accès autorisé ».

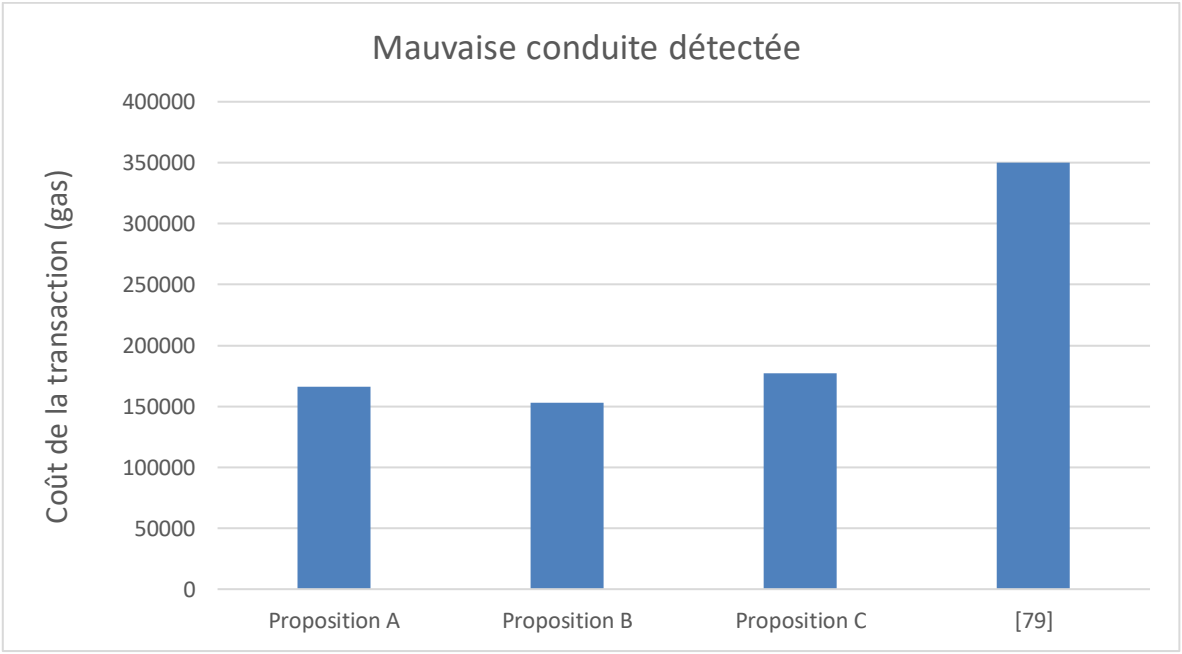

Figure IV.12. Comparaison du coût de transaction de la réponse « Mauvaise conduite détectée ».

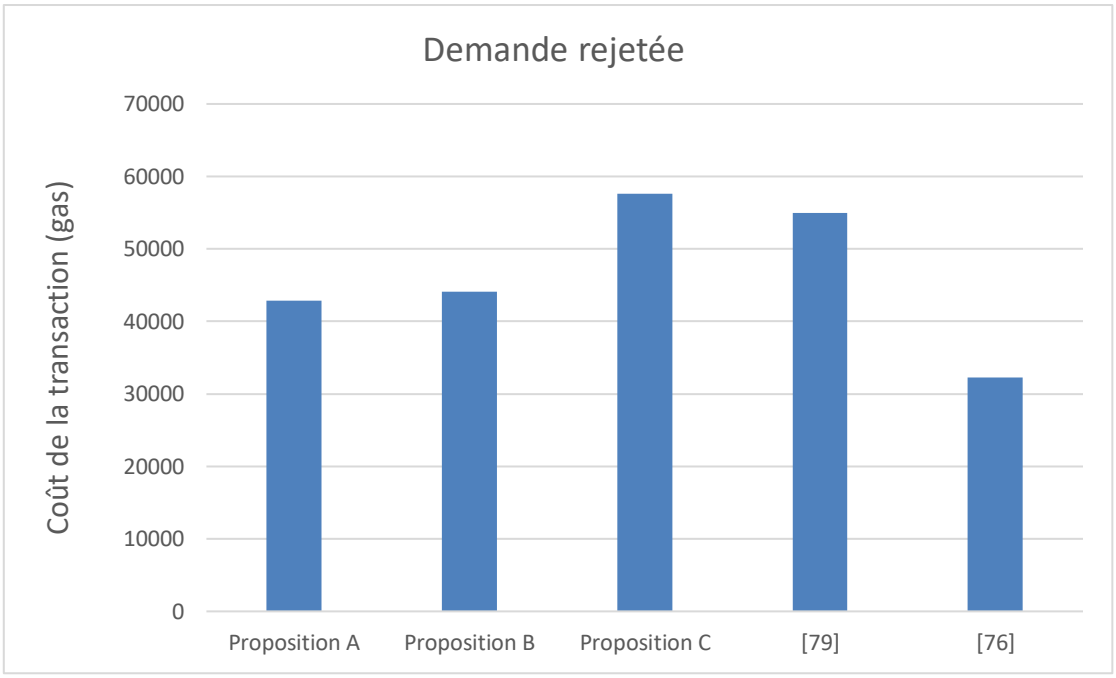

Figure IV.13. Comparaison du coût de transaction de la réponse « Demande rejetée ».

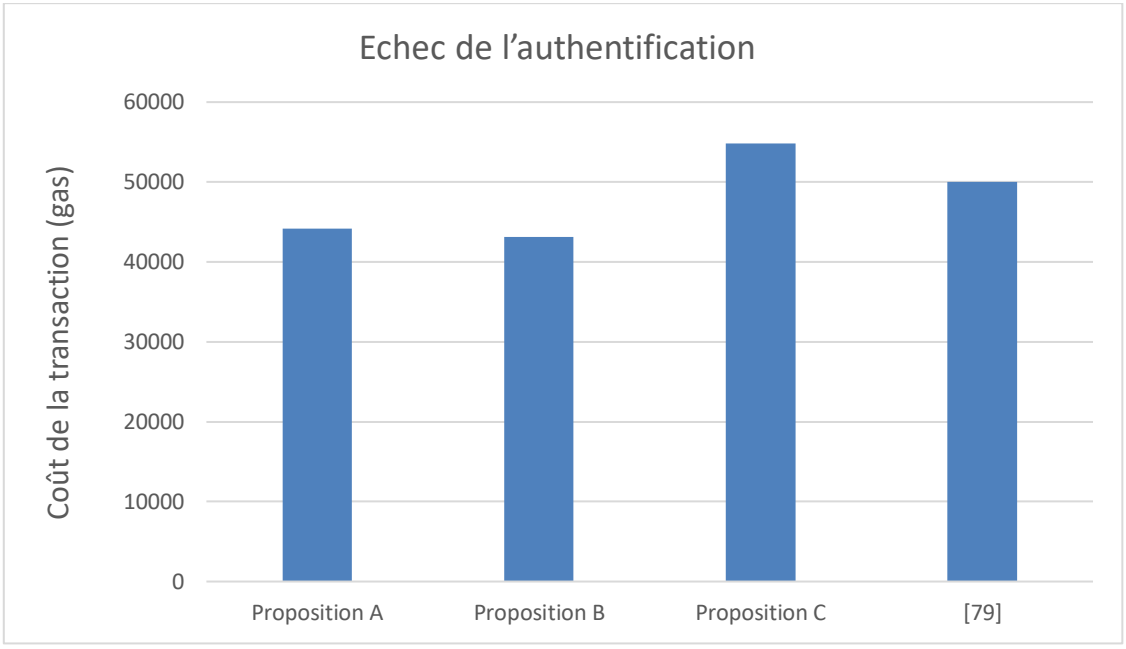

Figure IV.14. Comparaison du coût de transaction de la réponse « Echec de l'authentification ».

IV.5.5 Surcharge de communication entre les contrats

Dans la proposition A, lorsque le sujet envoie une demande d'accès, la méthode AccessControl () est exécutée et le résultat est envoyé au sujet, de cette manière nous consommons deux transactions, de même pour la proposition B. Tandis que dans la proposition C, lorsque l'ACC reçoit une requête d'accès, il déclenche le RC puis le MJC, ce qui signifie que la proposition C

consomme quatre transactions, ce qui reste mieux que la proposition [79] qui consomme cinq transactions, et la proposition [74] qui consomme six transactions comme illustré dans la figure IV.15.

Ces valeurs ne sont pas énormes lorsqu'il s'agit d'une demande d'accès, mais dans un système hospitalier où il y a une centaine de demandes d'accès par jour, ces valeurs seront multipliées par 100 et la différence sera énorme. Ainsi, dans nos trois propositions, nous réduisons les frais généraux de communication entre les contrats, et en particulier dans les propositions A et B. Nous pouvons donc conclure que les propositions A et B sont les mieux adaptées aux systèmes de soins de santé en termes de surcharge de communication.

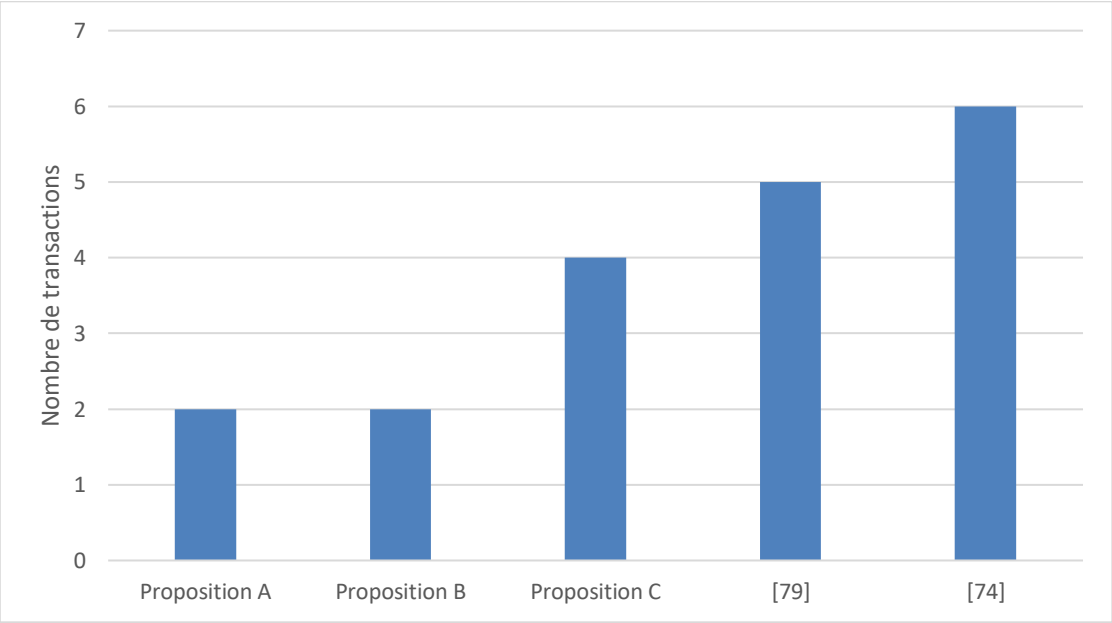

Figure IV.15. Coût de la demande d'accès en nombre de transactions.

IV.5.6 Latence de la réponse à la demande d'accès

Nous mesurons le temps de réponse en calculant la différence entre le temps de soumission d'une demande d'accès et le temps de réception de la réponse d'accès dans nos trois propositions. Nous calculons la moyenne de 100 mesures pour avoir la valeur finale, et nous supposons que 10 nouveaux utilisateurs rejoignent notre système chaque mois. Les résultats obtenus sont illustrés sur la figure IV.16. Nous observons que la latence augmente lorsque nous augmentons le nombre d'utilisateurs de 20 dans les propositions A et B, tandis que dans la proposition C, elle augmente lorsque nous augmentons le nombre d'utilisateurs de 10 seulement. Nous observons également que la proposition A est celle qui prend le moins de temps en termes de réponse à la demande d'accès, suivi par la proposition B qui reste meilleure que les résultats obtenus dans [71,79] (sachant que

dans [79], ils ont obtenu de meilleurs résultats que [71]). Nous pouvons donc conclure que la proposition A est la mieux adaptée aux systèmes de télésanté en termes de latence de réponse à la demande d'accès et qu'en termes de passage à l'échelle c'est la plus scalable.

Notez que ces valeurs pourraient être améliorées si l'hôpital utilisait des serveurs puissants pour faire le minage ou s'il choisit d'augmenter la limite de gas (payez plus pour une réponse plus rapide) ou s'il utilisait des consensus à faible coût tels que PoS ou DPoS.

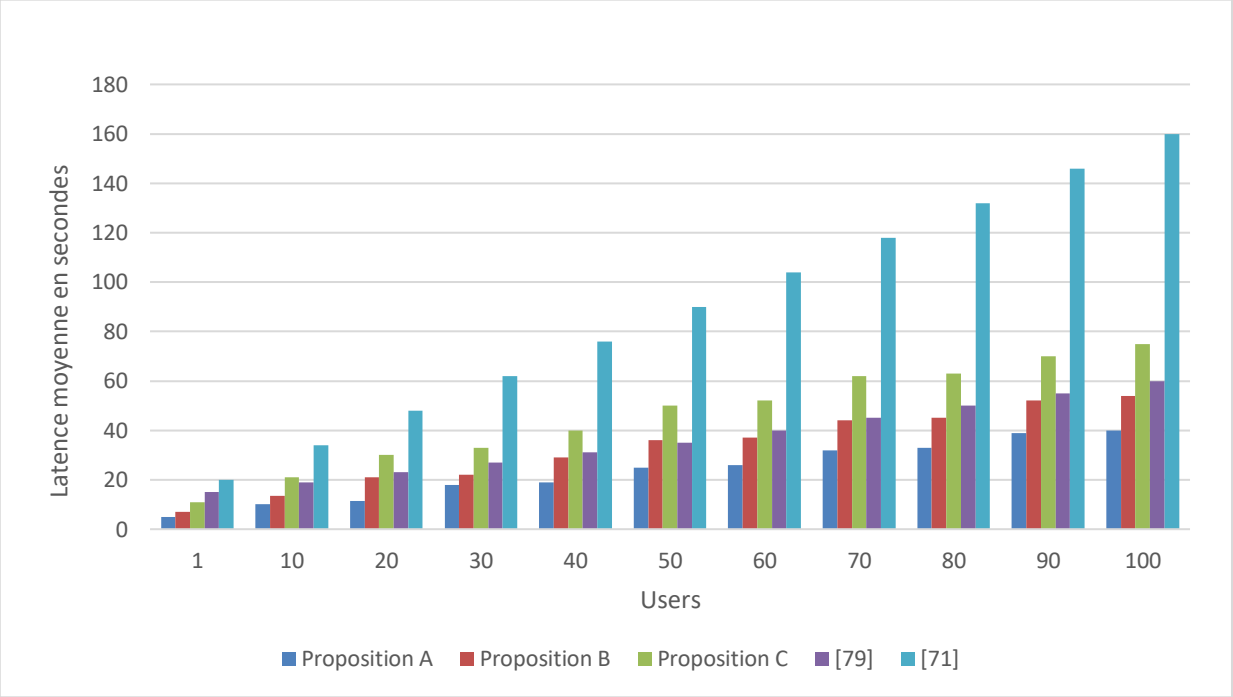

Figure IV.16. Latence de la réponse à la demande d'accès.

IV.6 Sécurité contre diverses attaques

IV.6.1 Attaque de contrefaçon de l'autorisation

Dans une attaque de contrefaçon de l'autorisation, l'attaquant tente de contrefaire les résultats de la demande d'accès. Dans nos quatre propositions, le contrôle d'accès se fait via des contrats intelligents et les demandes et réponses d'accès sont envoyées dans des transactions qui sont encapsulées dans des blocs, ainsi l'attaquant ne peut contrefaire les résultats à aucune étape du contrôle d'accès, à moins qu'il n'ait une puissance supérieure à 50% de la puissance de la BC [95], c'est pourquoi nous recommandons l'utilisation d'une blockchain publique dans les systèmes de soins de santé.

IV.6.2 Attaque par usurpation

Une attaque par usurpation se produit lorsqu'une personne ou un programme parvient à s'identifier comme un autre pour obtenir un avantage illégitime.

Dans notre système, toutes les entités sont doublement enregistrées dans le système. D'abord, l'hôpital fournit un numéro d'identification unique à toutes les entités, ensuite, chaque identifiant sera lié à une adresse dans la BC. Grâce à cette double identification et à la propriété d'immuabilité de la BC, l'attaquant ne peut pas usurper l'identité d'un utilisateur.

Dans le contrôle d'accès basé les NFT, chaque jeton est associé à une clé secrète (IdNft), de sorte que seuls les utilisateurs pouvant prouver qu'ils possèdent cette clé peuvent utiliser le jeton, ainsi l'attaque par usurpation devient impossible à réaliser.

IV.6.3 Attaque par inondation

L'attaque par inondation consiste à envoyer une succession de requêtes d'accès visant à réaliser un déni de service. Avec notre contrôle d'accès dynamique, l'attaquant sera pénalisé (bloqué) dès qu'il commencera à faire plusieurs accès fréquents, et cette pénalité augmentera avec le temps, ce qui rend impossible de réaliser cette attaque.

IV.7 Conclusion

Dans ce chapitre nous avons présenté les algorithmes que nous avons utilisés pour les quatre propositions de contrôle d'accès, ainsi que leurs implémentations. Les résultats et l'analyse des performances ont démontré que les trois propositions basées sur SC ont résolu les problèmes que nous reprochons aux travaux connexes. La première proposition a obtenu de meilleurs résultats en termes de coût de trasaction liés au déploiement des contrats, de coût de transaction liés aux différentes réponses, en termes de latence dans la réponse à la demande d'accès, et en termes de surcharge de communication. La deuxième proposition a obtenu de meilleurs résultats en termes de coût de transactions liés a l'exécution des fonctions et en termes de surcharge de communication, tandis que la troisième proposition a obtenu de bons résultats en termes de coût de déploiement des contrats juste après la première proposition. Nous concluons que la première proposition est celle qui convient aux systèmes de soins de santé. La proposition NFT quand à elle a obtenu de meilleurs résultats en termes de cout de création de jetons.

Conclusion générale et perspectives

Dans les systèmes de soins de santé basés sur l'IoT, le patient partage ses données personnelles et physiologiques avec plusieurs entités à distance, ce qui augmente le risque de violation de la confidentialité et de la vie privée du patient, pour cela le contrôle d'accès aux données du patient est primordial. Dans cette thèse, nous proposons trois schémas différents de contrôle d'accès basé sur les contrats intelligents de la technologie Blockchain et un schéma basé sur les NFT, tous centrés sur le patient, en d'autres termes, le patient est celui qui contrôle l'accès à ses données.

Dans le premier schéma de contrôle d'accès basé sur les contrats intelligents, chaque propriétaire d'objet crée un seul contrat intelligent pour la gestion des demandes d'accès de tous les sujets du système et y définit ses politiques d'accès. Tandis que dans le deuxième schéma, il crée un contrat pour chaque sujet du système, et dans le troisième schéma, il crée un contrat intelligent pour tous les sujets, tandis que l'administrateur crée deux autres contrats pour vérifier l'enregistrement et la conduite du sujet.

L'analyse des performances a démontré que les trois propositions basées sur les contrats intelligents ont apporté de meilleurs résultats que les travaux connexes en termes de coût de transaction liés au déploiement des contrats, aux différentes réponses et à l'exécution des fonctions, ainsi qu'en termes de surcharge de communication et de latence dans la réponse à la demande d'accès. Cependant, le premier schéma de contrôle d'accès basé sur les contrats intelligent est celui qui a apporté de meilleurs résultats par rapport aux deux autres propositions, ainsi avec ce schéma, nous avons optimisé les systèmes de soins de santé qui se base sur les contrats intelligents pour contrôler l'accès aux données en réduisant les coûts de transactions ce qui rend cette architecture plus adoptable par rapports aux architectures trouvées dans la littérature et en réduisant aussi le temps d'accès aux données qui un critère primordiale dans le contexte des soins de santé. A noter également, qu'avec la proposition NFT, nous avons réduit le coût de la création des jetons par rapport aux propositions trouvées dans la littérature.

Comme perspective à notre travail réalisé dans le cadre de cette thèse nous prévoyons de sécuriser la communication entre les appareils IoT et la couche Fog pour pouvoir proposer une architecture de soins de santé plus sécurisée. Nous prévoyons aussi d'optimiser le schéma de contrôle d'accès basé sur les NFT en termes de coûts de déploiement et de latence d'accès, et enfin nous prévoyons d'utiliser le fog computing pour effectuer une analyse et un filtrage de données.

Publications (5)

Articles de revues internationales (1)

- Hideyat Zerga, Asma Amraoui and Badr Benmammar. "Distributed, dynamic and trustworthy access control for telehealth systems". *Concurrency and Computation: Practice and Experience, Wiley InterScience Edition*, 2022; Volume: 34 issue: 28, e7352. DOI: 10.1002/cpe.7352. First published: 01 October 2022. Revue de classe A (Impact Factor = 1.831).

Articles d'actes de conférences internationales (1)

- H. Zerga, A. Amraoui and B. Benmammar, "Blockchain based access control for home hospitalization during covid-19," *2022 19th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2022, pp. 692-697, doi: 10.1109/SSD54932.2022.9955649.

Articles d'actes de conférences nationales (3)

- Hideyat Zerga, Asma Amraoui and Badr Benmammar. "Fog computing and Blockchain in healthcare IoT systems". *Conférence Nationale sur les Télécommunications et ses Applications (CNTA 2021)*. Ain-Témouchent, Algeria, December 20-21, 2021.
- Hideyat Zerga, Asma Amraoui and Badr Benmammar. "Smart contract-based access control for IoT healthcare systems". *Conférence Nationale sur les Télécommunications et ses Applications (CNTA 2021)*. Ain-Témouchent, Algeria, December 20-21, 2021.
- Hideyat Zerga, Badr Benmammar et Asma Amraoui. "Optimisation des systèmes de e-santé à base de l'internet des objets". Ateliers SACONET. (SACONET'19). 21-23 Décembre 2019. Université Oran 1. Oran. Algérie.

Membre d'un projet de recherche (1)

- Projet PRFU : IoT et E-santé : optimisation et innovation. Spécialité : réseaux et systèmes informatiques distribués. 2019-2022.

Références

[1] T.A. Jaffe, E. Hayden, L. Uscher-Pines, et al. Telehealth use in emergency care during coronavirus disease 2019: a systematic review. *J Am Coll Emerg Physicians Open*, no. 2(3):e12443, 2021.

[2] H. Ben Hassen, N. Ayari, B. Hamdi. A home hospitalization system based on the Internet of things, Fog computing and cloud computing. *Informatics in Medicine Unlocked*. No. 20, 2020.

[3] N. Mani, A. Singh, S. L. Nimmagadda. An IoT Guided Healthcare Monitoring System for Managing Real-Time Notifications by Fog Computing Services. in Procedia Computer Science (Elsevier B.V), pp. 850–859, 2020.

[4] P. Verma, S. K. Sood. Fog assisted-IoT enabled patient health monitoring in smart homes. *IEEE Internet of Things Journal*, no. 5, pp. 1789–1796, 2018.

[5] V. Dhillon, D. Metcalf, M. Hooper. Blockchain in Healthcare. *In: Blockchain Enabled Applications. Apress, Berkeley, CA*, pp. 201–220, 2021.

[6] I. Yaqoob, K. Salah, R. Jayaraman, et al. Blockchain for healthcare data management: opportunities, challenges, and future recommendations. *Neural Comput & Applic*, no. 34, pp.11475–11490, 2022.

[7] N. Sohrabi, X. Yi, Z. Tari, and I. Khalil. BACC: Blockchain-Based Access Control For Cloud Data. *In Proceedings of the Australasian Computer Science Week Multiconference (ACSW '20)*, Article 10, pp. 1–10, 2020.

[8] S. Sun, R. Du, S. Chen and W. Li. Blockchain-Based IoT Access Control System: Towards Security, Lightweight, and Cross-Domain. In *IEEE Access*, vol. 9, pp. 36868-36878, 2021.

[9] N. Tariq, A. Qamar, M. Asim, F. A. Khan. Blockchain and Smart Healthcare Security: A Survey. *Procedia Computer Science*, vol. 175, pp. 615-620, 2020.

[10] S. K. Rana, K. Nisar, A. A. Ag Ibrahim, A. K. Rana, N. Goyal, P. Chawla. Blockchain Technology and Artificial Intelligence Based Decentralized Access Control Model to Enable Secure Interoperability for Healthcare. *Sustainability*, no. 14(15):9471, 2022.

[11] Statista Research Department, nombre d'utilisateur d'internet dans le monde, 2022. Accédé le 20/09/2022. Disponible en ligne : https://fr.statista.com/statistiques/571074/nombre-d-utilisateurs-d-internet-dans-le-monde-2005--/#:~:text=Cette%20statistique%20pr%C3%A9sente%20le%20nombre,%C3%A0%204%2C9%20milliards%20environ

[12] N. Shahid and S. Aneja. Internet of Things: Vision, application areas and research challenges. *Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017*, vol. 10, no. 7, pp. 583–587, 2017.

[13] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[14] C. Sarkar, A. U. Nambi, V. Prasad and A. Rahim. A scalable distributed architecture towards unifying IoT applications. *IEEE*, pp. 508-513, 2014.

[15] R. wirleess World, IoT architecture basics | IoT hardware, software architecture, 2012. Accédé le 07/05/2022. Disponible en ligne: https://www.rfwireless-world.com/IoT/IoT-architecture.html.

[16] Oracle, qu'est ce que l'IoT, Accédé le 20/09/2022 . Disponible en ligne : https://www.oracle.com/fr/internet-of-things/what-is-iot/

[17] J. Gubbi, R. Buyya, and S. Marusic, Smart Socket for Electricity Control in Home Environment, Procedia Computer Science,vol. 157, Pages 465-472, 2019.

[18] A. A. Lisbon. A Study on Cloud and Fog Computing Security Issues and Solutions. *International Journal of Innovative Research in Advanced Engineering (IJIRAE).*, vol. 03, pp. 17–23, 2017.

[19] G. Oliver and S. Knight. Storage is a strategic issue: Digital preservation in the cloud. *D-Lib Mag.*, vol. 21, no. 3–4, 2015.

[20] M. Firdhous, O. Ghazali, and S. Hassan. Fog Computing: Will it be the Future of Cloud Computing?. *Third Int. Conf. Informatics Appl.*, no. October, pp. 8–15, 2014.

[21] X. Q. Pham and E. N. Huh. Towards task scheduling in a cloud-fog computing system. *18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1-4, 2016.

[22] R. Roman, J. Lopez, and M. Mambo. Mobile Edge Computing, Fog et al .: A Survey and Analysis of Security Threats and Challenges. Future Generation Computer Systems. vol. 78, pp. 680–698, 2018.

[23] K. P. Saharan and K. Anuj. Fog in Comparison to Cloud: A Survey. International Journal of Computer Applications. vol. 122, no. 3, pp. 10–12, 2015.

[24] Medium, the architecture of fog network – A bridge between Cloud and IoT, 2019, Accédé le 03/01/2022. Disponible en ligne : https://hindujab.medium.com/the-architecture-of-fog-network-a-bridge-between-cloud-and-iot-part-2-a45612145a0b

[25] P. Hu, S. Dhelim, H. Ning, and T. Qiu. Survey on fog computing : architecture , key technologies , applications and open issues. *J. Netw. Comput. Appl.*, vol. 98, no. April, pp. 27–42, 2017.

[26] P. Zhang, M. Zhou, and G. Fortino. Security and trust issues in Fog computing : A survey Security and trust issues in Fog computing : a survey. *Futur. Gener. Comput. Syst.*,vol. 88, pp. 16-27, 2018.

[27] W. Shi and S. Dustdar. The Promise of Edge Computing. *Computer*, vol. 49, no. 5, pp. 78-81, 2016.

[28] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge Computing: Vision and Challenges. *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, 2016.

[29] R. Mohanan, What Is Edge Computing? Components, Examples, and Best Practices, *Spiceworks*, 2022. Accédé le 25/09/2022. Disponible en ligne: https://www.spiceworks.com/tech/edge-computing/articles/what-is-edge-computing/

[30] K. Ismail, edge computing vs fog computing whats the difference. *csm wire*, 2018. Accédé le 03/01/2022. Disponible en ligne : https://www.cmswire.com/information-management/edge-computing-vs-fog-computing-whats-the-difference/

[31] République Francaise, Qu'est ce qu'un système de santé, *Vie publique*, 2022. Accédé le 15/09/2022. Disponible en ligne : https://www.vie-publique.fr/fiches/37853-definition-et-acteurs-du-systeme-de-sante-francais

[32] A. Kumari, S. Tanwar, S. Tyagi, and N. Kumar. Fog computing for Healthcare 4 . 0 environment : Opportunities. *Comput. Electr. Eng.*, vol. 72, pp. 1–13, 2018.

[33] F. A. Kraemer, A. E. Braten, N. Tamkittikhun, and D. Palma. Fog Computing in Healthcare — A Review and Discussion. *IEEE Access*, vol. 3536, no. 2169, pp. 1–16, 2017.

[34] A. Calihlman, architecturs in the IoT civilization, NetBurner, 2019. Accédé le 05/08/2022. Disponible en ligne : https://www.netburner.com/learn/architectural-frameworks-in-the-iot-civilization/.

[35] M. Maksimović. Implementation of Fog computing in IoT-based healthcare system. *JITA - J. Inf. Technol. Appl. (Banja Luka) - APEIRON*, vol. 14, no. 2, pp. 100–107, 2018.

[36] T. Aladwani. Scheduling IoT Healthcare Tasks in Fog Computing Based on their Importance. Procedia Computer Science. 163, pp.560-569, 2019.

[37] G. Manogaran, C. Thota, and D. Lopez. Big Data Security Intelligence for Healthcare Industry 4 . 0. *Thames, L., Schaefer, D. (eds) Cybersecurity for Industry 4.0. Springer Series in Advanced Manufacturing. Springer, Cham.* 2017.

[38] E. A. Oladimeji, L. Chung, H. T. Jung, and J. Kim. Managing security and privacy in ubiquitous ehealth information interchange. *Proc. 5th Int. Conf. Ubiquitous Inf. Manage. Commun. (ICUIMC)*, New York, NY, USA, pp. 26:1–26:10. 2011.

[39] S. Al-janabi, I. Al-shourbaji, M. Shojafar, and S. Shamshirband. Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications. *Egypt. Informatics J.*, vol. 18, no. 2, pp. 113–122, 2017.

[40] C. S. Nandyala and H. Kim. From Cloud to Fog and IoT-Based Real-Time U-Healthcare Monitoring for Smart Homes and Hospitals. *International Journal of Smart Home*, vol. 10, no. 2, pp. 187–196, 2016.

[41] L. M. Dang, J. Piran, D. Han, K. Min, and H. Moon. A Survey on Internet of Things and Cloud Computing for Healthcare. Electronics, pp. 1–49, 2019.

[42] K. Abouelmehdi, A. B. Hessane, and H. Khaloufi. Big healthcare data : preserving security and privacy. *J. Big Data*, pp. 1–18, 2018.

[43] K. Abouelmehdi, A. Beni-hssane, H. Khaloufi, and E. Nationale. ScienceDirect ScienceDirect Big data security and privacy in healthcare : A Review. Procedia Comput. Sci., vol. 113, pp. 73–80, 2017.

[44] D. Elangovan, C.S. Long, F.S. Bakrin, C.S. Tan, K.W. Goh,S.F. Yeoh, et al. The Use of Blockchain. Technology in the Health Care Sector: Systematic Review. JMIR Med Inform, vol. 10(1), 2022.

[45] C.C. Agbo,Q.H. Mahmoud QH, J.M. Eklund. Blockchain Technology in Healthcare: A Systematic Review. *Healthcare*, vol. 7(2), 56, 2019.

[46] T. Kumar, V. Ramani, I. Ahmad, A. Braeken, E. Harjula and M. Ylianttila. Blockchain Utilization in Healthcare: Key Requirements and Challenges. *IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pp. 1-7, 2018.

[47] S. Jang, J. Guejong, and J. Jeong. Fog computing architecture based blockchain for industrial IoT. *in Computer Science(), Springer, Cham*, vol. 11538, pp. 1–14, 2019.

[48] Hölbl M, Kompara M, Kamišalić A, Nemec Zlatolas L. A Systematic Review of the Use of Blockchain in Healthcare. *Symmetry*, vol.10(10), no.470, 2018.

[49] A. Gayte, Halving bitcoin : définition, *Numerama*, 2022. Accédé le 27/11/2022. Disponible en ligne: https://www.numerama.com/tech/720617-quest-ce-que-le-halving-du-bitcoin-et-pourquoi-est-ce-si-important.html#:~:text=Cette%20situation%20a%20depuis%20chang%C3%A9,bitcoins%20pour%20chaque%20nouveau%20bloc.

[50] T. Mcghin, K. R. Choo, C. Zhechao, and D. He. Journal of Network and Computer Applications Blockchain in healthcare applications : Research challenges and opportunities. *J. Netw. Comput. Appl.*, vol. 135, pp. 62–75, 2019.

[51] Tariq N, Asim M, Al-Obeidat F, Zubair Farooqi M, Baker T, Hammoudeh M, Ghafir I. The Security of Big Data in Fog-Enabled IoT Applications Including Blockchain: A Survey. *Sensors*, vol. 19(8), no.1788, 2019.

[52] M. Sookhak, M.R. Jabbarpour, N.S. Safa, F.R. Yu. Blockchain and smart contract for access control in healthcare: A survey, issues and challenges, and open issues. Journal of Network and Computer Applications, vol. 178, no.102950, 2021.

[53] Sghaier Omar, A., Basir, O. Capability-Based Non-fungible Tokens Approach for a Decentralized AAA Framework in IoT. *In: Blockchain Cybersecurity, Trust and Privacy. Advances in Information Security*, Springer, Cham, vol. 79, pp. 6-31, 2020.

[54] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Disponible sur: https://bitcoin.org/bitcoin.pdf

[55] H. zerga, A. Amraoui, B. Benmammar. Fog computing and healthcare in healthcare IoT systems. Conférence national sur les télécommunications et ses applications. 2021.

[56] D. F. Maesa, P. Mori. Blockchain 3.0 applications survey. *Journal of Parallel and Distributed Computing*,vol. 138, pp. 99-114, 2020.

[57] C. Pirtle, J. Ehrenfeld. Blockchain for Healthcare: The Next Generation of Medical Records?. *J Med Syst*, vol. *42*, no. 172, 2018.

[58] Khatoon, A. A blockchain-based smart contract system for healthcare management. *Electronics*, 9(1), 94. 2020.

[59] H. R. Hasan, K. Salah, R. Jayaraman *et al*. Blockchain-Based Solution for COVID-19 Digital Medical Passports and Immunity Certificates. in *IEEE Access*, vol. 8, pp. 222093-222108, 2020.

[60] H. Mhamdi, M. Ayadi, A. Ksibi, A. Al-Rasheed, B.O. Soufiene,S. Hedi. SEMRAchain: A SecureElectronic Medical Record Based onBlockchain Technology. *Electronics*, 11, 3617, 2022.

[61] A. Ouaddah. A.A. Elkalam, A.A. Ouahman. Towards a Novel Privacy-Preserving Access Control Model Based on Blockchain Technology in IoT, *Europe and MENA Cooperation Advances in Information and Communication Technologies. Advances in Intelligent Systems and Computing, Springer* vol. 520, no. February, 2018.

[62] Y. Zhu, X. Wu, Z. Hu. Fine Grained Access Control Based on Smart Contract for Edge Computing. *Electronics*, vol.11(1), no.167, 2022.

[63] A. Muthanna, A.A. Ateya, A. Khaimov,I. Gudkova,A. Abuarqoub,K. Samouylov, A. Koucheryavy. Secure and Reliable IoT Networks Using Fog Computing with Software-Defined Networking and Blockchain. *Journal of Sensor and Actuator Networks*, vol. 8(1), no.15, 2019.

[64] G. Vincent, “Modèles Principaux de contrôle d’accès,” 2017. Accédé le 02/09/2022. Disponible en ligne: https://slideplayer.fr/slide/11843058/.

[65] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo. Attribute-Based Access Control. in *Computer*, vol. 48, no. 2, pp. 85-88, 2015.

[66] H. S. G. Pussewalage and V. A. Oleshchuk. An Attribute Based Access Control Scheme for Secure Sharing of Electronic Health Records. *IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pp. 1-6, 2016.

[67] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-Based Access Control Models y z 1 INTRODUCTION. *IEEE Computer*, vol. 29, no. 2, pp. 38-47, 1996.

[68] A. Outchakoucht, H. Es-samaali, A. Abou, E. Kalam, and S. Benhadou. Apprentissage par Renforcement et Blockchain : Nouvelle approche pour sécuriser l’IoT. OpenScience, no. October, pp. 1–20, 2018.

[69] D.F. Maesa, P. Mori, and L. Ricci. Blockchain based access control . *IFIP International Conference on Distributed Applications and Interoperable Systems*. *Springer*, Cham, p. 206-220, 2017.

[70] Y. Zhu, Y. Qin, G. Gan, Y. Shuai and W. C. Chu. TBAC: Transaction-Based Access Control on Blockchain for Resource Sharing with Cryptographically Decentralized Authorization. *IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, pp. 535-544, 2018.

[71] J. Xu *et al*. Healthchain: A Blockchain-Based Privacy Preserving Scheme for Large-Scale Health Data, in *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8770-8781, 2019.

[72] O. Novo. Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT.*IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184-1195, 2018.

[73] Kim, J, Park, N. Role-based Access Control Video Surveillance Mechanism Modeling in Smart Contract Environment. Trans Emerging Tel Tech; vol. 33, no. 4. 2022 : e4227.

[74] H. Guo, E. Meamari, and C.C. Shen. Multi-Authority Attribute-Based Access Control with Smart Contract. *In Proceedings of the 2019 International Conference on Blockchain Technology (ICBCT 2019)*. Association for Computing Machinery, New York, NY, USA, pp. 6–11, 2019.

[75] Y. Zhang, M. Yutaka, M. Sasabe and S. Kasahara. Attribute-Based Access Control for Smart Cities: A Smart-Contract-Driven Framework. in *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6372-6384, 15 April15, 2021.

[76] D. Di Francesco Maesa, P. Mori and L. Ricci. Blockchain Based Access Control Services. *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. 1379-1386, 2018.

[77] J. Wang, P. Gong, H. Wang, W. Zhang, C. Sun, and B. Zhao. A Right Transfer Access Control Model of Internet of Things Based on Smart Contract. Secur. Commun. Networks, vol. 2022, p. 3682952, 2022.

[78] Y. Zhang, S. Kasahara, Y. Shen, X. and Jiang. Smart Contract-Based Access Control for the Internet of Things. *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594-1605, 2019.

[79] A. Saini, Q. Zhu, N. Singh, Y. Xiang, L. Gao, and Y. Zhang. A Smart-Contract-Based Access Control Framework for Cloud Smart Healthcare System, *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5914-5925, 2021.

[80] A. Outchakoucht, E.S. Hamza, and J.P. Leroy. Dynamic Access Control Policy based on Blockchain and Machine Learning for the Internet of Things. *International Journal Of Advanced Computer Science and Applications*, vol. 8, no 7, p. 417-424, 2017.

[81] G.G. Dagher, J. Mohler, M. Milojkovic, and P.B. Marella. Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. *Sustain. Cities Soc*., vol. 39, pp. 283–297, 2018.

[82] V. Hossein, K. Esmaeili, M.E. Dargahi, T. Khonsari, et al. A. Blockchain-based privacy-preserving healthcare architecture. *IEEE Canadian Conference of Electrical and Computer Engineering*. vol. 180, pp. 31-47, 2019.

[83] A. Ismail, Q. Wu, M. Toohey, Y. C. Lee, Z. Dong and A. Y. Zomaya. TRABAC: A Tokenized Role-Attribute Based Access Control using Smart Contract for Supply Chain Applications. *IEEE International Conference on Blockchain (Blockchain)*, pp. 584-589, 2021.

[84] N. Fotiou, I. Pittaras, V.A. Siris, S. Voulgaris, G.C. Polyzos. OAuth 2.0 authorization using blockchain-based tokens, 2020.

[85] A. Sghaier Omar, O. Basir. Capability-Based Non-fungible Tokens Approach for a Decentralized AAA Framework in IoT. *In: Choo, KK., Dehghantanha, A., Parizi, R. (eds) Blockchain Cybersecurity, Trust and Privacy. Advances in Information Security*, vol 79. Springer, Cham. pp. 7-31, 2020.

[86] B. Pradhan, S. Bhattacharyya, and K. Pa. IoT-Based Applications in Healthcare Devices. *J. Healthc. Eng*., vol. 2021, Article ID 6632599, 2021.

[87] M. Hajvali, S. Adabi, A. Rezaee, et al. Software architecture for IoT-based health-care systems with cloud/fog service model. *Cluster Comput*, vol. 25, pp. 91–118, 2022.

[88] H. Ben Hassen, N. Ayari, B. Hamdi. A home hospitalization system based on the Internet of things, Fog computing and cloud computing. *Informatics in Medicine Unlocked*. No. 20, 2020.

[89] H. Zerga, A. Amraoui and B. Benmammar. Blockchain based access control for home hospitalization during covid-19. *19th International Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 692-697, 2022.

[90] H. Zerga, A. Amraoui and B. Benmammar. Distributed, dynamic and trustworthy access control for telehealth systems. *Concurrency and Computation: Practice and Experience, Wiley InterScience Edition*, vol. 34, n. 28, 2022.

[91] Remix- ide for smart contract deployment provided by Ethereum. Disponible en ligne :https://remix.ethereum.org

[92] Web3 javascript api to interact with ethreum nodes. Disponible en ligne :https://github.com/ethereum/wiki/wiki/JavaScript-API

[93] Hideyat Zerga. Access control. GitHub repository. Disponible sur :https://github.com/hidou716/Access-control

[94] Open Zeppelin. Open Zeppelin contracts. GitHub repository. Disponible en ligne: https://github.com/OpenZeppelin/openzeppelincontracts/blob/master/contracts/token/ERC721/ERC721.sol

[95] D. Yermack. Corporate Governance and Blockchains. *Review of Finance*, vol. 21(1), pp. 7–31,2017.

Résumé

Avec sa capacité à "minimiser l'intervention humaine lors de la génération, de l'échange et de la consommation de données", l'Internet des objets (IoT) se déploie de plus en plus dans tous les secteurs, en particulier dans le secteur de la santé. L'IoT dans les soins de santé permet de garder les patients connectés avec des appareils portables et d'autres outils de surveillance des patients à distance afin d'aider les praticiens à travailler plus efficacement. Cependant, cette innovation implique que les patients partagent à distance leurs données personnelles et physiologiques avec le personnel hospitalier, ce qui peut mettre en danger la vie privée du patient. Ainsi, la mise en place d'un contrôle d'accès est obligatoire. Par conséquent, l'objectif de cette thèse est de parvenir à un contrôle d'accès distribué et fiable pour les systèmes de soins de santé en utilisant la technologie de la Blockchain. Pour ce faire, nous avons proposé trois approches différentes de contrôle d'accès basées sur des contrats intelligents et une approche basée sur les jetons non fongibles. Nos propositions ont été comparées avec des travaux connexes en termes de latence de réponse à la demande d'accès et de consommation de gaz liée au déploiement du contrat, à l'exécution des fonctions et aux différentes réponses. Les résultats obtenus sont très satisfaisants.
Mots clés : IoT, soins de santé, contrôle d'accès, blockchain, contrat intelligent, jetons non fongibles.

Abstract

With its ability to "minimize human intervention when generating, exchanging and consuming data", the Internet of Things (IoT) is increasingly being deployed in all sectors, particularly in the health sector. IoT in healthcare keeps patients connected with wearable devices and other remote patient monitoring tools to help practitioners work more efficiently. However, this innovation involves patients sharing their personal and physiological data remotely with the hospital staff, which may endanger patient privacy. Thus, the implementation of an access control is mandatory. Therefore, the objective of this thesis is to achieve distributed and reliable access control for healthcare systems using Blockchain technology. To do so, we proposed three different approaches to access control based on smart contracts and one approach based on non-fungible tokens (NFT). Our proposals were compared with related works in terms of access request response latency and gas consumption related to contract deployment, function execution and different responses. The results obtained are very satisfactory.
Keywords: IoT, healthcare, access control, blockchain, smart contract, NFT.

ملخص

بفضل قدرتها على "تقليل التدخل البشري عند إنشاء البيانات وتبادلها واستهلاكها"، يتم استخدام إنترنت الأشياء بشكل متزايد في جميع القطاعات، لا سيما في قطاع الصحة. تحافظ إنترنت الأشياء في مجال الرعاية الصحية على اتصال المرضى بالأجهزة القابلة للارتداء وغيرها من أدوات مراقبة المريض عن بُعد لمساعدة ممارسين المهنة على العمل بكفاءة أكبر. ومع ذلك، فإن هذا الابتكار ينطوي على مشاركة المرضى عن بعد لبياناتهم الشخصية والفسيولوجية مع موظفي المستشفى، مما قد يعرض خصوصية المريض للخطر. وبالتالي، فإن تنفيذ التحكم في الوصول إلزامي. لذلك، فإن الهدف من هذه الأطروحة هو تحقيق التحكم في الوصول الموزع والموثوق لأنظمة الرعاية الصحية باستخدام تقنية سلسلة الكتل. للقيام بذلك، اقترحنا ثلاث طرق مختلفة للتحكم في الوصول استنادًا على العقود الذكية وطريقة واحدة تعتمد على الرموز الغير القابلة للاستبدال. تمت مقارنة مقترحاتنا بالأعمال ذات الصلة من حيث زمن استجابة طلب الوصول واستهلاك الغاز المرتبط بنشر العقود وتنفيذ الوظيفة والاستجابات المختلفة. النتائج التي تم الحصول عليها مرضية للغاية.
الكلمات الرئيسية: إنترنت الأشياء، الرعاية الصحية، التحكم في الوصول، سلسلة الكتل، العقد الذكي، الرموز غير القابلة للاستبدال.