# Sécurité et vie privée centrées sur l'utilisateur dans l'IoT

Tidiane Sylla

université de BORDEAUX

U.S.T.T-B

THÈSE EN COTUTELLE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

**DOCTEUR DE**

**L'UNIVERSITÉ DE BORDEAUX**

**ET DE L'UNIVERSITÉ DES SCIENCES, DES TECHNIQUES ET DES TECHNOLOGIES DE BAMAKO**

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE (EDMI)

ÉCOLE DOCTORALE DES SCIENCES ET TECHNOLOGIES DU MALI (EDSTM)

SPÉCIALITÉ INFORMATIQUE

Par Tidiane SYLLA

# Sécurité et vie privée centrées sur l'utilisateur dans l'IoT

Sous la direction de Francine KRIEF et de Karim SAMAKÉ
Co-encadrant : Mohamed Aymen CHALOUF

Soutenue le 07/12/2021

Membres du jury :

| | | |
|---|---|---|
| M. CHAUMETTE Serge | Professeur, Université de Bordeaux | Président |
| M. BOUABDALLAH Abdelmadjid | Professeur, Université de Technologie de Compiègne | Rapporteur |
| M. Hassine MOUNGLA | Maître de Conférences, HDR Université Paris Descartes | Rapporteur |
| Mme KRIEF Francine | Professeur, Bordeaux INP | Directrice |
| M. SAMAKE Karim | Maître de Conférences, HDR Université des Sciences, des Techniques et des Technologies de Bamako | Co-Directeur |
| M. CHALOUF Mohamed Aymen | Maître de Conférences, Université de Rennes 1 | Co-Encadrant |

**Titre** Sécurité et vie privée centrées sur l'utilisateur dans l'IoT

**Résumé** Aujourd'hui, les nombreuses applications de l'Internet des Objets (IoT : Internet of Things) peuvent significativement améliorer la vie quotidienne des utilisateurs. Grâce à ces applications, il est possible de commander à distance les différents appareils de la maison, surveiller les signes vitaux d'un patient et alerter automatiquement son médecin en cas de problème. Cependant, les problèmes de sécurité et de protection de la vie privée empêchent les utilisateurs de faire pleinement confiance à ces applications, ce qui peut avoir pour effet de ralentir l'adoption globale de ces technologies et leur large déploiement. Pour résoudre ces problèmes de sécurité et de protection de la vie privée, plusieurs solutions ont été proposées. Cependant, plusieurs défis restent encore à relever pour permettre une large adoption de ces applications. L'approche centrée sur l'utilisateur semble être très pertinente pour relever un grand nombre de ces défis. Pour offrir une sécurité et une protection de la vie privée centrées sur l'utilisateur et permettre la prise en charge de nombreuses applications IoT, les travaux de cette thèse proposent d'adapter la mise en œuvre des mécanismes de sécurité et de protection de la vie privée en fonction du contexte de l'utilisateur.

Dans un premier temps, cette thèse présente l'architecture CASPaaS (*Context-Aware Security and Privacy as a Service*). Cette architecture de sécurité et de protection de la vie privée sensibles au contexte pour l'IoT est basée sur l'approche 'as a service'. Elle garantit l'adaptation dynamique et personnalisée des services de sécurité et de protection de la vie privée en fonction du contexte de l'utilisateur. Grâce à la conception 'as a service', cette architecture se caractérise par une grande flexibilité qui lui permet de prendre en charge de nombreuses applications IoT. Dans un second temps, cette thèse présente un système permettant de gérer la sécurité et la fiabilité de l'architecture CASPaaS elle-même. Ce système, appelé SETUCOM (*SEcure and TrUstworthy COntext Management*) gère la sécurité des données contextuelles échangées au sein de l'architecture CASPaaS ainsi que la confiance des sources de données. Ceci permet de pallier un grand nombre d'attaques pouvant conduire au disfonctionnement de notre architecture CASPaaS. Dans un troisième temps, cette thèse présente un nouveau système de gestion décentralisée des autorisations sensibles au contexte pour l'IoT. Ce système, basé également sur l'approche 'as a service', offre à l'utilisateur une gestion dynamique, décentralisée et simple des autorisations. Dans un dernier temps, cette thèse s'intéresse au déploiement du service CASPaaS au plus proche des utilisateurs en se basant sur une infrastructure de type *Edge Computing*. Dans ce contexte, nous proposons une nouvelle stratégie de placement dynamique de ce service. Cette stratégie fait appel à des techniques de l'intelligence artificielle afin de garantir un placement efficace tout en optimisant les différentes performances (réseau, service, capacités des nœuds *Edge*, processus de placement lui-même, etc.).

**Mots clés** Internet des Objets, Sécurité, Confidentialité, Intégrité, Authentification, Vie privée, Contexte utilisateur, Contraintes d'environnement

**Title** User-centric security and privacy in IoT

**Abstract** Today, the Internet of Things (IoT) applications can significantly improve the daily life of users. Thanks to these applications, it is possible to control the various devices in the home remotely, monitor a patient's vital signs, and automatically alert his doctor when problems occur. However, security and privacy issues hinder the users to trust fully these applications, what may have as effect to slow down the overall adoption of these technologies and their widespread deployment. Several solutions have been proposed to address these security and privacy issues. Despite this, several challenges still need to be overcome to enable the global adoption of this type of application. The user-centric approach seems to be very relevant to address a large number of these challenges. To provide user-centric security and privacy protection and to enable the numerous IoT applications support, this thesis proposes to adapt the implementation of security and privacy protection mechanisms based on the user context and enable support for many IoT applications.

First, this thesis presents the CASPaaS (Context-Aware Security and Privacy as a Service) architecture. This context-aware security and privacy architecture for IoT is based on the 'as a service' approach. It ensures the dynamic, personalized adaptation of security and privacy services based on the user's context. Thanks to the 'as a service' design, this architecture is characterized by high flexibility that will make it able to support multiple IoT applications. In a second step, this thesis presents a system to manage the security and reliability of the CASPaaS architecture itself. This system, called SETUCOM (SEcure and TrUstworthy COntext Management), addresses the security of the contextual data exchanged within the CASPaaS architecture and the trustworthiness of these data sources. This allows mitigating a large number of attacks that can lead to the malfunctioning of our CASPaaS architecture. In a third step, this thesis presents a new decentralized context-aware authorization management system for the IoT. This system, also based on the 'as a service' approach, offers the user dynamic, decentralized, and simple authorization management. Finally, this thesis focuses on deploying the CASPaaS service closer to the users based on an Edge Computing infrastructure. In this context, we propose a new dynamic placement strategy for this service. This strategy uses artificial intelligence techniques to ensure an efficient placement while optimizing the different performances (network, service, nodes capacities, placement process, etc.).

**Keywords** Internet of Things, Security, Confidentiality, Integrity, Authentication, Privacy, User context, Environment constraints

---

**Unité de recherche**

LaBRI (Laboratoire Bordelais de Recherche en Informatique) UMR 5800 - 351 Cours de la Libération, 33405 Talence

# Table des matières

# Table des figures

# Liste des tableaux

*À mes grands parents, Kadidiatou dite Paye, que Dieu t'accepte dans son paradis éternel.*

*À mes chers parents, Moustapha et Néné, sans vous, rien n'aurait pu être possible, que Dieu vous protège et vous prête une longue vie pleine de santé et de prospérité.*

*À ma chère épouse, Assitan, merci de m'avoir soutenu tout au long de cette aventure, merci pour le sacrifice consenti, que Dieu te protège.*

*À mes chers enfants, à tous les membres de ma famille et à tous mes amis.*

# Remerciements

*Je voudrais tout d'abord exprimer ma profonde gratitude envers ma directrice de thèse, le professeur Francine Krief, qui m'a proposé cette thèse et m'a accompagné tout au long de ces trois années. Son aide, ses encouragements, sa disponibilité et son soutien sans faille ont été déterminants pour la réalisation de cette thèse.*

*Je voudrais particulièrement exprimer ma reconnaissance envers mon co-encadrant, Mohamed Aymen Chalouf, qui n'a ménagé aucun effort pour me guider et m'accompagner tout au long de cette thèse.*

*Je voudrais également remercier Karim Samaké, mon co-directeur pour les efforts consentis malgré son départ à la retraite.*

*Je voudrais aussi remercier le Service de Coopération et d'Action Culturelle de l'Ambassade de France au Mali pour le financement de mes séjours en France durant cette thèse. Je remercie également l'Université des Sciences, des Techniques et des Technologies de Bamako pour son accompagnement financier.*

*Je tiens à remercier tous les membres de l'équipe PROGRESS et tout le personnel Administratif du LaBRI. Aussi, je tiens à remercier tous ceux qui de près ou de loin ont contribué à l'aboutissement de ces recherches que ce soit par un soutien moral, technique, scientifique, financier, amical ou familial.*

# Liste des publications

## A. Chapitres de livres

□ **Tidiane Sylla**, Mohamed-Aymen Chalouf, et Francine Krief. "*Adaptation du niveau de sécurité des applications IoT*" Chapitre 10, Ouvrage **La gestion et le contrôle intelligents de la sécurité dans l'IoT**, Mohamed-Aymen Chafouf, Collection ISTE (Déc. 2021)

## B. Journaux internationaux

□ **Tidiane Sylla**, Mohamed-Aymen Chalouf, Francine Krief and Karim Samaké. "*Context-aware security in the internet of things : a survey*" **International journal of autonomous and adaptive communications systems** 14.3 (2021) : 231-263, Indersciences.

□ **Tidiane Sylla**, Mohamed-Aymen Chalouf, Francine Krief and Karim Samaké. "*SE-TUCOM : Secure and Trustworthy Context Management for Context-Aware Security and Privacy in the Internet of Things*" **Security and Communication Networks** (2021), Hindawi.

□ **Tidiane Sylla**, Leo Mendiboure, Mohamed-Aymen Chalouf, and Francine Krief. "*Blockchain-Based Context-Aware Authorization Management as a Service in IoT*" **Sensors** 21(22), (2021), MDPI.

## C. Conférences internationales

□ **Tidiane Sylla**, Mohamed Aymen Chalouf, Francine Krief and Karim Samaké. "*Towards a context-aware security and privacy as a service in the internet of things*" **The 13$^{th}$ IFIP International Conference on Information Security Theory and Practice.** Springer, Cham, 2019.

# Introduction Générale

## Contexte et Motivations

L'émergence des technologies de l'électronique et de l'informatique embarquée et des communications sans fil a permis le développement massif de l'informatique ubiquitaire. L'informatique ubiquitaire consiste à rendre l'informatique omniprésente dans tous les compartiments de la vie quotidienne des utilisateurs. Concrètement, des systèmes sur puces électroniques, des actionneurs et des moyens de communication sans fil sont intégrés aux objets du quotidien. Ces objets évolués peuvent communiquer et participer à un réseau local ou global d'informations tel que l'Internet.

L'Internet des Objets (*Internet of Things* - IoT en anglais), s'inscrit dans ce cadre. L'IoT est une infrastructure globale de dispositifs interconnectés à l'aide des technologies de l'information et de la communication. Il a pour objectif de connecter le monde physique au monde virtuel afin de fournir des services intelligents. Les données collectées par les objets sur le monde physique sont traitées et analysées dans le Cloud par des algorithmes évolués, par exemple d'intelligence artificielle. Plusieurs décisions peuvent être prises : l'envoi de notifications aux utilisateurs via leurs smartphones et le déclenchement d'une série d'actionneurs.

Les applications de l'IoT couvrent de nombreux domaines. Elles comprennent la ville intelligente (e-santé, maisons intelligentes, mobilité intelligente, bâtiments intelligents, etc.), l'agriculture intelligente, l'industrie 4.0, la surveillance de l'environnement, etc. Par exemple, grâce à la e-santé, les médecins peuvent télé-consulter leurs patients en ayant accès à leurs signes vitaux sans les faire se déplacer. Par ailleurs, la grande majorité des utilisateurs des applications IoT vivent dans les villes. Selon l'Agence des affaires économiques et sociales des Nations Unies (UN DESA), 68% de la population mondiale vivra en ville à l'horizon 2050.

La ville intelligente intègre différents types d'applications IoT. Cependant, la mise en œuvre de ces applications n'est pas sans conséquence sur la sécurité des données et la vie privée des utilisateurs. En effet, les applications de l'IoT sont confrontées à plusieurs

problèmes de sécurité hérités des protocoles de communication utilisés, par exemple TCP/IP. Les risques de sécurité dans l'IoT proviennent également de la vulnérabilité des objets, contraints en ressources (énergie et capacité de calcul). Les risques pour la vie privée des utilisateurs proviennent de la grande quantité de données collectées, parfois sans leur consentement et utiliser dans de nombreuses applications.

Les solutions proposées dans de nombreux travaux se sont focalisées sur la mise en œuvre de mécanismes pour l'IoT permettant d'assurer un service de sécurité ou une combinaison de services de sécurité ou de protection de la vie privée. Cependant, les risques concernent tous les services de sécurité ainsi que la protection de la vie privée. En effet, les vulnérabilités et les menaces de sécurité sont présentes à tous les niveaux : dispositifs, communications réseaux et applications web/mobiles. En outre, ces solutions sont proposées pour une application. Or, un seul utilisateur peut disposer d'une dizaine d'applications IoT. Ainsi, proposer une solution permettant d'assurer l'ensemble des services de sécurité et la protection de la vie privée adaptée à l'IoT et prenant en charge différents types d'application IoT est nécessaire.

Par ailleurs, la grande majorité des solutions de sécurité proposée ne prend pas en considération les caractéristiques des utilisateurs d'une part (fracture numérique, facilité d'utilisation et préférences), et des applications IoT d'autre part (dynamicité, flexibilité). L'approche de la sécurité et de la protection de la vie privée centrées sur l'utilisateur dans l'IoT apparaît comme une alternative sérieuse pour la prise en charge de ces caractéristiques. En effet, elle pourrait permettre d'assurer l'ensemble des services de sécurité et de protection de la vie privée des utilisateurs tout en considérant leurs caractéristiques.

La prise en charge de la mobilité dans les solutions de sécurité et de protection de la vie privée pour l'IoT est une nécessité. L'Edge Computing est une technologie prometteuse et peut être une solution intéressante dans la gestion de la mobilité des utilisateurs IoT. En ce sens, c'est une technologie qui permet de ramener les capacités de calcul et de stockage au plus près des utilisateurs, quel que soit leur localisation géographique.

## Problématique

Plusieurs travaux se sont intéressés à la sécurité et à la protection de la vie privée centrées sur l'utilisateur [BZWL19, GMSS23, IVH16, LXI17, MHB+17]. Cependant, plu-

sieurs des problématiques importantes doivent être résolues :

— **une spécification de l'approche de la sécurité et de la protection de la vie privée centrées sur l'utilisateur :** la considération des caractéristiques des utilisateurs, notamment, la fracture numérique, la facilité d'utilisation et des préférences de ces derniers est problématique. L'approche centrée sur l'utilisateur paraît pertinente pour relever ce défi. En effet, elle peut permettre une meilleure prise en charge des caractéristiques des utilisateurs dans la mise en œuvre de la sécurité et de la protection de vie privée dans l'IoT. Cela permettra par la même de favoriser l'adoption des applications IoT par les utilisateurs de la ville intelligente. La sécurité sensible au contexte est un moyen intéressant et efficace pour la mise en œuvre de l'approche centrée sur l'utilisateur dans l'IoT ;

— **une architecture de sécurité et de protection de la vie privée sensibles au contexte complète et globale :** la sécurité sensible au contexte dans l'IoT devra considérer l'ensemble des services de sécurité et de protection de la vie privée. Elle ne devra pas être verticale, c'est-à-dire, spécifique à une application, comme proposée dans les travaux existants. Elle devra être horizontale, autrement dit, permettre la prise en charge d'une diversité d'applications. Ainsi, une architecture capable de fournir la sécurité sensible au contexte complète et horizontale doit être proposée. La sécurité sensible au contexte fait face à un défi important, à savoir la gestion sécurisée et fiable de la sensibilité au contexte. En effet, des informations contextuelles falsifiées peuvent être utilisées pour induire le système de sécurité en erreur. L'utilisation des techniques d'intelligence artificielle pour déceler les informations fiables parmi les non fiables est une piste pertinente ;

— **la gestion de la mobilité et l'intégration dans les nouvelles architectures réseaux :** l'IoT étant une technologie ubiquitaire, la sécurité et la protection de la vie privée lors de la mobilité de l'utilisateur doit être assurées. L'intégration des solutions de sécurité et de protection de la vie privée pour l'IoT dans les nouvelles architectures réseaux telles que l'Edge Computing et la 5G est une approche très intéressante. En effet, cette intégration peut permettre de combler les nombreuses lacunes des solutions existantes telles que la prise en charge de la mobilité de l'utilisateur, des applications temps réel ou quasi-temps réel et le passage à l'échelle. De ce fait, la conception d'une architecture de sécurité et de protection de la vie

privée sensibles au contexte qui peut intégrer ces nouvelles architectures est nécessaire l'IoT.

## Contributions

Les travaux effectués dans cette thèse ont pour objectif de fournir une solution aux problèmes ci-dessus identifiés. Ainsi, pour une meilleure solution de sécurité et de protection de la vie privée centrées sur l'utilisateur dans l'IoT, nous avons réalisé les contributions suivantes :

— **la définition d'une nouvelle architecture de communication IoV :** comme nous l'avons mentionné, les solutions existantes pour la sécurité et la protection de la vie privée des utilisateurs dans l'IoT présentent plusieurs limites : l'absence de mécanismes pouvant assurer l'ensemble des services de sécurité et de protection de la vie privée, les solutions proposées sont spécifiques à une application et ne prennent pas en charge la mobilité de l'utilisateur. Pour résoudre ces problèmes, nous avons proposé une architecture de sécurité et de protection de la vie privée sensibles au contexte en tant que service. Cette architecture met en œuvre l'approche centrée sur l'utilisateur, considère les caractéristiques des applications IoT et prend en charge une diversité d'applications ;

— **la définition et l'implémentation d'un système de gestion sécurisée et fiable de la sensibilité au contexte basée sur des dispositifs de confiance :** comme nous l'avons noté, la grande majorité des solutions existantes n'ont pas considéré la sécurité et la confiance dans la gestion de la sensibilité au contexte. Pour surmonter cette limite, un système de gestion sécurisée et fiable de la sensibilité au contexte basée sur des dispositifs de confiance pour l'IoT est nécessaire. Ainsi, le système de sécurité sensible au contexte pourra prendre des décisions d'adaptation de niveau de sécurité fiable. C'est pourquoi, nous avons proposé un système pour la sécurité et la confiance de la gestion de la sensibilité au contexte en prenant en compte la fiabilité des informations contextuelles et le comportement des dispositifs de l'utilisateur. Les évaluations effectuées ont prouvé les avantages de notre solution en termes de détection d'informations contextuelles non fiables, de dispositifs malveillants et d'impact sur la consommation énergétique des dispositifs ;

— **la définition et l'implémentation d'un système de gestion décentralisée des autorisations sensibles au contexte en tant que service :** le contrôle d'accès est un service de sécurité important dans l'IoT. Le contrôle d'accès sensible au contexte permet de résoudre les problèmes liés à la nature dynamique des applications IoT. Les solutions existantes ont plusieurs limites : serveur d'autorisation central, prise en charge de la mobilité de l'utilisateur, passage à l'échelle, absence de dynamisme et de facilité d'utilisation, etc. La proposition d'un système de gestion d'autorisation dynamique et qui adapte le niveau des autorisations en fonction du contexte est nécessaire. C'est pourquoi nous avons proposé une solution de gestion décentralisée des autorisations sensibles au contexte en tant que service pour l'IoT. La solution composée de la blockchain et d'un service de sécurité sensible au contexte permet aux utilisateurs de gérer les autorisations contextuelles avec une grande aisance. Les évaluations ont prouvé l'efficience et la faible latence de notre proposition ;

— **la définition et l'implémentation d'une stratégie de placement de blocs de service dans une infrastructure Edge Computing :** comme nous l'avons mentionné, l'IoT étant une technologie ubiquitaire, la mise en œuvre de la sécurité et de la protection de la vie privée doit être autant ubiquitaire. Cela permettra d'adapter le niveau de sécurité des applications quel que soit la localisation de l'utilisateur et quel que soit le moment. Pour cette mise en œuvre omniprésente de la sécurité et de la protection de la vie privée, l'intégration et le déploiement dans les infrastructures d'Edge Computing nous paraît intéressant. Cela pourra être appuyé par les nouvelles architectures réseaux telles que les réseaux définis logiciellement. Dans ce sens, le placement dynamique de service dans l'Edge comprend de nombreuses limites : taux d'échec de placement élevé dû à la mobilité, problème de gestion des ressources, gestion énergétique, etc. Ainsi, nous avons proposé une stratégie de placement de blocs de service pour l'IoT. Les expérimentations réalisées ont montré l'efficacité de notre proposition en termes de taux d'échec lors du passage à l'échelle.

## Organisation de la thèse

Ce manuscrit de thèse est organisé comme suit (Fig. 1) :

— **Chapitre 1 'Internet des Objets, sécurité et protection de la vie privée' :** ce

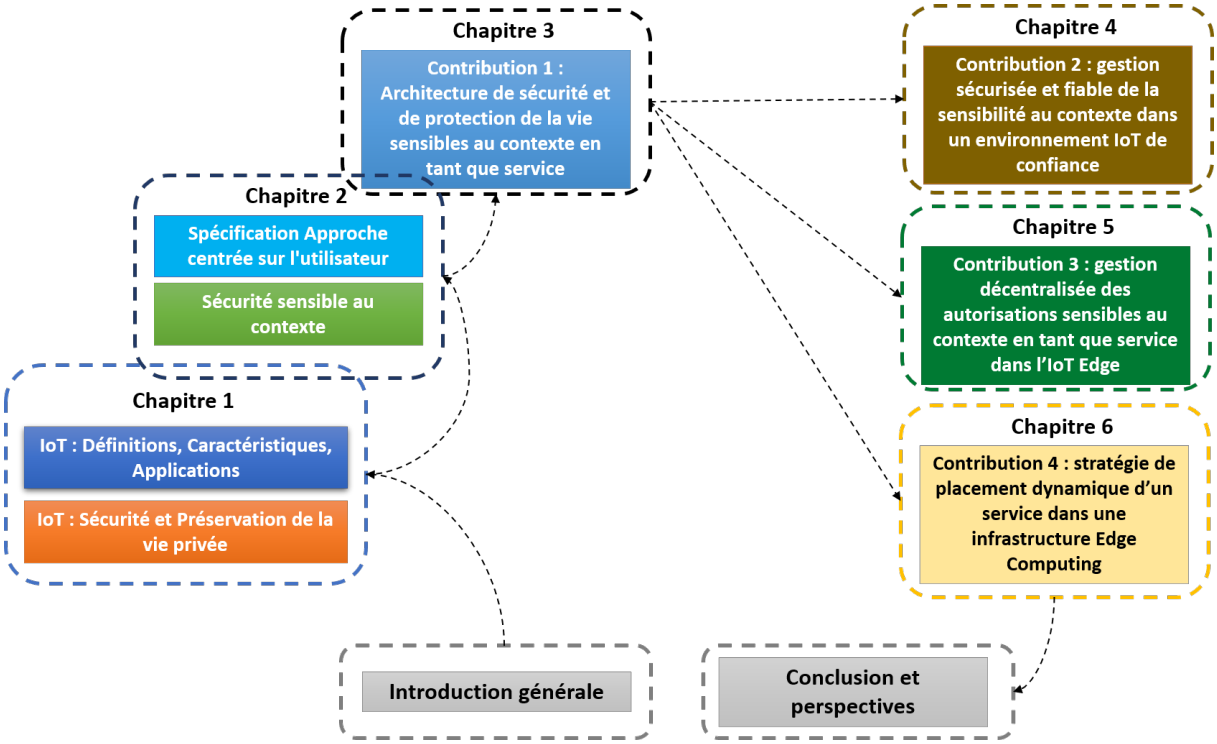



FIGURE 1 – Organisation du manuscrit de thèse

premier chapitre introduit le concept d'IoT, ses caractéristiques, son écosystème et ses applications. Ensuite, nous y dressons l'état des lieux de la sécurité et de la protection de la vie privée dans l'IoT. Pour ce faire, nous effectuons une revue systématique des problèmes de sécurité, notamment les menaces et les vulnérabilités auxquelles chaque couche de l'architecture IoT est exposée. Nous définissons également les services de sécurité et de protection de la vie privée nécessaires au bon fonctionnement des applications IoT ;

— **- Chapitre 2 'Sécurité Sensible au Contexte dans l'Internet des Objets' :** dans ce chapitre, nous effectuons une spécification de l'approche de la sécurité et de la protection de la vie privée centrées sur l'utilisateur. Puis, nous démontrons comment la sécurité sensible au contexte peut être un moyen efficace pour la mise en œuvre de cette approche. Par la suite, nous définissons l'informatique sensible au contexte ainsi que la sécurité sensible au contexte dans l'IoT. Ensuite, nous effectuons une revue comparée des travaux existants ayant traités la sécurité sensible au contexte dans l'IoT. Pour finir, nous discutons les points forts de ces différents travaux et nous proposons des pistes de recherche pour relever les défis restants ;

— **Chapitre 3 'Vers une architecture de sécurité et de protection de la vie privée sensibles au contexte en tant que service dans l'IoT' :** nous présentons dans ce chapitre notre première contribution : une architecture de sécurité et de protection de la vie privée sensibles au contexte en tant que service. Pour ce faire, nous identifions dans un premier temps les limites des solutions existantes de sécurité et de protection de la vie privée sensibles au contexte dans l'IoT. A partir des exigences non considérées, nous proposons une nouvelle architecture de sécurité et de protection de la vie privée pour l'IoT. Cette architecture se base sur un plan de connaissance pour gérer la perception du contexte et un plan de sécurité et de protection de la vie privée pour l'adaptation du niveau de sécurité. Nous présentons et discutons aussi de l'intégration de l'architecture proposée dans l'architecture de référence pour l'IoT de l'IUT-T ;

— **Chapitre 4 'Gestion sécurisée et fiable de la sensibilité au contexte dans un environnement IoT de confiance : Application à la Smart City' :** dans ce chapitre, nous introduisons notre deuxième contribution : gestion sécurisée et fiable de la sensibilité au contexte dans un environnement IoT de confiance. Pour ce faire, nous commençons par l'identification des limites en termes de sécurité et de confiance dans la gestion de la sensibilité au contexte dans l'IoT. Par la suite, nous identifions les menaces auxquelles un tel système est exposé. Pour y répondre, nous définissons un mécanisme de sécurisation des échanges d'information contextuelle léger et sécurisé. Ensuite, nous décrivons notre mécanisme de gestion de la confiance basé sur la réputation. Concrètement, ce mécanisme utilise deux techniques d'intelligence artificielle, les réseaux bayésiens et la logique floue, pour respectivement évaluer la fiabilité des informations contextuelles et le comportement des dispositifs de l'utilisateur. Enfin, nous réalisons les expérimentations pour comparer les mécanismes proposés à ceux des travaux existants et nous analysons aussi leur sécurité ;

— **Chapitre 5 'Gestion des autorisations sensibles au contexte en tant que service dans l'IoT Edge' :** nous introduisons dans ce chapitre une troisième contribution : gestion décentralisée des autorisations sensibles au contexte en tant que service dans l'IoT Edge. Premièrement, nous mettons en exergue les limites des solutions existantes sur la gestion dynamique des autorisations dans l'IoT. Par la suite, nous présentons les avantages d'un système d'autorisation sensible au

contexte basé sur des jetons d'accès contextuels. Nous soulignons également les apports de la décentralisation dans la gestion des autorisations, par exemple basée sur la blockchain. Troisièmement, nous introduisons notre architecture décentralisée et sensible au contexte et les menaces auxquelles elle est exposée. Ensuite, nous évaluons les mécanismes proposés, discutons les résultats et analysons leur sécurité ;

— **Chapitre 6 'Placement de l'architecture CASPaaS dans une infrastructure Edge computing' :** dans ce chapitre, nous présentons notre quatrième contribution dans cette thèse : stratégie de placement dynamique d'un service dans une infrastructure Edge Computing. Nous commençons par une introduction aux notions connexes au placement de services dans l'Edge Computing. Deuxièmement, nous faisons un état de l'art des travaux existants sur les stratégies de placement de services dans les infrastructures Edge Computing, puis nous identifions leurs limites. Pour répondre à ces limites, nous introduisons notre stratégie de placement dynamique de blocs de services dans une infrastructure Edge. Celle-ci s'appuie sur la prédiction de la mobilité de l'utilisateur en utilisant l'intelligence artificielle, l'optimisation multicritère avec les fonctions d'utilité et les décisions de placements basées sur de la logique floue. Ensuite, nous évaluons la faisabilité de notre proposition et comparons ses performances à celles des travaux existants ;

— **'Conclusion et perspectives' :** dans ce dernière partie, nous effectuons une conclusion générale des travaux réalisés et décrits dans ce manuscrit. Par la suite, nous présentons également les perspectives pour des travaux futurs dans la continuité de cette thèse.

# Chapitre 1

# Internet des Objets, sécurité et protection de la vie privée

## 1.1 Introduction

L'Internet des Objets ou *Internet of Things* (IoT) en anglais, est un paradigme qui consiste à connecter le monde virtuel de l'Internet au monde réel à travers des objets dits intelligents. Ces objets sont dotés de capacités de communication et d'échange de données sur Internet. Le concept a été évoqué pour la première fois en 1999 par K. Ashton du Massachusetts Institute of Technology (MIT) [BF16]. K. Ashton a été l'un des membres fondateurs de l'Auto-ID Centre, un groupe qui comprenait plusieurs entreprises et universités dont l'Université de Cambridge et le MIT. Le but visé par ce groupe était de permettre le suivi et le contrôle de chaque objet à distance, où qu'il soit dans le monde, au moyen d'étiquette RFID (Radio Frequency Identifier) [McF15]. Cela devait grandement améliorer les processus de fabrication, la logistique et la gestion de stocks. Ce groupe a dès lors posé les bases de l'Internet des objets que le monde connait aujourd'hui.

Au fil des années, l'IoT est devenu une extension de l'Internet aux objets de tous les jours pour permettre à ces derniers d'être intelligents, c'est-à-dire qu'ils aient la faculté de communiquer et d'échanger sur Internet en utilisant des technologies de communication sans fil pour la plupart. Parmi ces objets, nous avons des actionneurs de portes automatisées, des ampoules, des réfrigérateurs, des thermomètres, des pacemakers, des glucomètres, des vêtements, des chaussures, des montres, etc. La figure 1.1 illustre l'écosystème de l'IoT. Le nombre d'objets connectés à Internet a subi une augmentation exponentielle. Selon une étude publiée en 2016 par Ericsson, il y aura au moins autour de deux objets connectés par personne, soit un total de dix-huit milliards d'objets connectés en 2022 avec une population mondiale estimée à 7,6 milliards de personnes [Eri19].

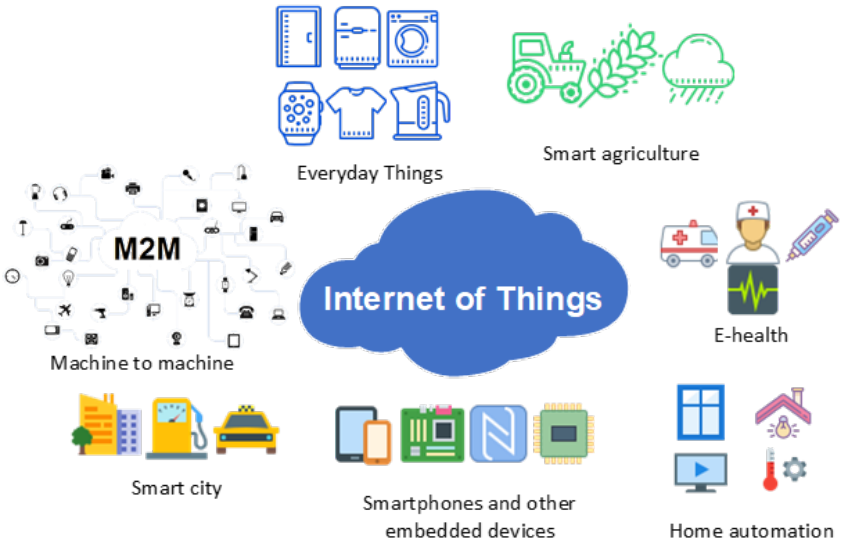



FIGURE 1.1 – Ecosystème de l'Internet des Objets

L'IoT est plébiscité ces dernières années par bon nombre d'acteurs comme étant la prochaine étape de l'évolution de l'Internet [Eva11, Pra17]. En effet, les nombreuses données collectées par les objets sont analysées dans le *Cloud* pour la prise de décision (Fig. 1.2). Les décisions prises sont ensuite notifiées à l'utilisateur à travers son application. C'est le cas, par exemple, dans l'agriculture, la protection de l'environnement, le suivi des activités sismiques, etc.

Dans ce chapitre, nous ferons un zoom sur le concept de l'Internet des Objets avant de présenter les défis sécuritaires auquel l'Internet des Objets est confronté. La suite de ce chapitre est organisée comme suit. Dans la section 1.2, nous donnerons les différentes définitions du concept IoT. Nous ferons une présentation des travaux et des efforts de normalisation en cours par les organismes de normalisation internationaux. Nous présenterons également les différentes architectures de référence et les différents protocoles de communication associés. Nous présenterons aussi quelques domaines d'applications de l'IoT et nous donnerons quelques caractéristiques inhérentes à l'IoT. Dans la section 1.3, nous discuterons des défis sécuritaires dans l'IoT. Nous présenterons également les menaces touchant l'IoT, puis nous définirons les différents services nécessaires à la sécurité et à la protection de la vie privée dans l'IoT. Dans la section 1.4, nous conclurons le chapitre et nous terminerons par une ouverture sur le prochain chapitre.

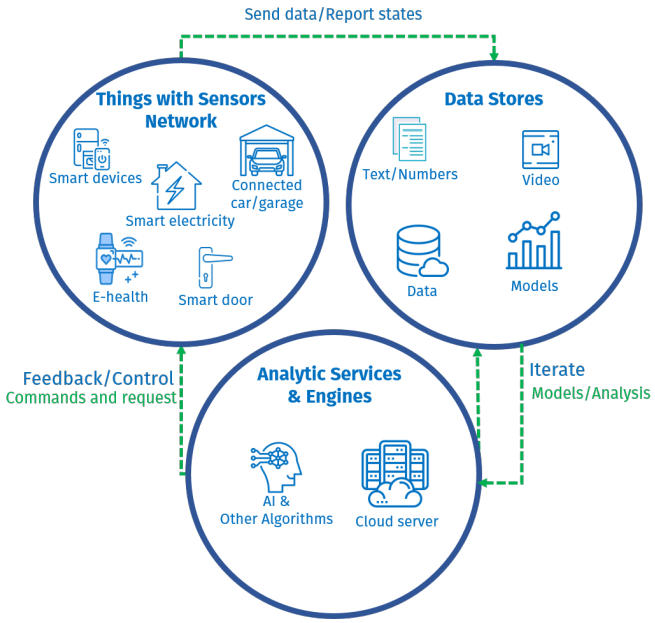



FIGURE 1.2 – Principales composantes de l'IoT [Pau14]

## 1.2 Concept de l'Internet des Objets

### 1.2.1 Définitions

Au cours de cette décennie, l'IoT a suscité un grand intérêt auprès des milieux universitaires et industriels. Cet intérêt s'explique par les différents avantages qu'apporte l'IoT dans tous les domaines (agriculture, santé, environnement, industrie, transport, énergie, etc.), mais également par la manne financière qu'il représente [AIM10, GBMP13, LL15]. La vision de l'IoT est très large et englobe plusieurs technologies tels que le RFID (Radio Frequency IDentifier), les réseaux de capteurs (WSN : Wireless Sensor Network), les communications machine à machine (M2M : Machine-To-Machine), etc. [SGFW10].

En considérant les différentes visions de l'IoT et les technologies qu'il englobe, il est très difficile de donner une définition standard de l'IoT. Cependant, plusieurs organismes internationaux ont cherché à donner une définition de l'IoT. En effet, les divergences entre les visions de l'IoT découlent du fait que les parties prenantes (alliances commerciales et industrielles, organismes de normalisation et de recherche) considèrent la question sous différents angles (orienté Internet ou orienté objet), ou en fonction de leurs intérêts spécifiques. L'IoT est sémantiquement défini comme : "*Un réseau mondial*

*d'objets interconnectés adressables de manière unique, basés sur des protocoles de communication standards* [BH08]". Les mêmes auteurs ont défini l'IoT en se focalisant sur les fonctionnalités offertes et la gestion des identités. Selon eux, l'IoT est défini comme suit : "*Les objets ont des identités et des personnalités virtuelles opérant dans des espaces intelligents utilisant des interfaces intelligentes pour se connecter et communiquer au sein de contextes sociaux, environnementaux et utilisateurs* [BH08]".

Le groupe de travail SG20 a été créé par l'UIT (Union Internationale des Télécommunications) au sein de sa branche UIT-T (chargée de la normalisation du secteur des télécommunications), afin d'entreprendre des travaux de normalisation sur les technologies IoT entrant dans le cadre des smart cities et des communautés intelligentes. Ce groupe de travail a défini l'IoT dans la recommandation ITU-T Y.2060 comme étant : "*une infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution* (ITU-T Y.2060, 2012)". Cette définition de l'IUT-T considère l'IoT sur deux visions : Internet et Objets. La première est orientée vers le réseau, car elle s'appuie sur le réseau Internet existant et la seconde vers l'intégration d'objets génériques aux réseaux afin d'offrir des services à valeurs ajoutées.

Quant à l'IETF (*Internet Engineering Task Force*), sa définition de l'IoT met l'accent sur les objets et les aspects de la connectivité des objets (adressage et identification uniques, protocoles de communication, etc.). Ainsi, selon l'IETF, l'IoT est un réseau d'objets physiques (chaque objet comportant des capteurs et des logiciels) qui peuvent échanger des données. L'ISO (*International Organization for Standardization*) et l'IEC (*International Electrotechnical Commission*) ont également défini l'IoT à travers le JTC1 (*Joint Technical Committee 1*), comité bipartie l'ISO/IEC chargé de la normalisation dans le secteur des technologies de l'information et de la communication. Selon l'ISO/IEC JTC1, l'IoT peut être défini comme une infrastructure d'entités (objets, personnes, systèmes) et de ressources d'informations interconnectées ainsi que des services intelligents leur permettant de traiter les informations du monde physique et virtuel et de réagir [ISO14].

À partir de cette multitude de définitions, nous pouvons définir l'IoT comme : "*Une infrastructure d'objets et de dispositifs interconnectés en utilisant les technologies de l'information et de la communication, et permettant de connecter le monde physique au monde virtuel dans le but d'offrir des services évolués et intelligents*".

### 1.2.2 Standardisation dans l'Internet des Objets

L'IoT a fait l'objet de travaux de standardisation depuis ses débuts, c'est-à-dire le milieu des années 2000. Ces travaux de standardisation permettent aux industries de créer des produits IoT interopérables. Elles permettent également de répondre à des problèmes de compatibilité. Plusieurs organismes et consortiums d'industries prennent part à ces travaux de standardisation. Parmi eux, nous avons : l'UIT, l'IETF, l'ISO/CEI, l'IEEE (*International Electrical and Electronics Engineers*), le 3GPP (*Third Generation Partnership Project*), oneM2M et l'alliance LoRA. La figure 1.3 illustre quelques normes phares parmi les normes existantes.

#### 1.2.2.1 UIT (ITU : International Telecommunication Union)

Le groupe de travail SG20 au sein de la branche de standardisation du secteur des technologies de l'Information et de la communication de l'UIT-T est chargé d'élaborer les normes internationales dans les technologies IoT, les communications machine à machine (M2M) et les réseaux de capteurs. Ses activités de standardisation visent les architectures IoT de bout-en-bout, les mécanismes d'interopérabilité des applications IoT, l'identification et la sécurité, etc.

Il travaille sur plusieurs séries de recommandations reparties en plusieurs thématiques spécifiques [1]. La série de recommandations concernant l'IoT et les villes intelligentes est la série Y.4000-Y4999. Les recommandations de cette série concernent spécifiquement les points suivants : les définitions et terminologies (série de recommandations Y.4050-Y.4099), les exigences et les cas d'utilisation (série de recommandations Y.4100-Y.4249), les infrastructures, la connectivité et les réseaux (série de recommandations Y.4250-Y.4399), les architectures, les cadres et les protocoles (série de recommandations Y.4400-Y.4549), les services, les applications, le calcul et le traitement des données (série de recommandations Y.4550-Y.4699), l'identification et la sécurité (série de recommandations Y.4800-Y.4899), etc.

#### 1.2.2.2 IETF (International Engineering Task Force

L'IETF est un organisme international à but non lucratif dont le rôle est d'élaborer et de promouvoir des standards Internet, notamment les standards concernant les

1. https://www.itu.int/ITU-T/recommendations/index$_sg.aspx?sg$=20

protocoles de communication. Ses standards sont généralement publiés sous forme de spécifications et édités dans des documents appelés *Request For Comments* (RFC).

L'IETF a réalisé plusieurs activités de standardisation afin de permettre l'intégration des objets contraints composant l'IoT dans Internet, et cela en utilisant des protocoles normalisés basés sur la technologie IP. C'est dans ce cadre que le protocole 6LoWPAN (*IPv6 for Low-power Wireless Personal Area Networks*) (RFC 4919) [KMS07] a été proposé pour permettre aux objets contraints, n'ayant pas la capacité de mettre en œuvre la pile protocolaire classique, de directement communiquer sur Internet en utilisant IPv6.

Par ailleurs, il a également élaboré à travers son groupe de travail ROLL (*Routing over low-power and Lossy networks*), un protocole de routage ouvert à vecteur de distance, RPL (*Routing Protocol for Low-power and lossy networks*) (RFC 6550) [WTB+12] adapté aux réseaux d'objets contraints. Le groupe de travail CoRE (*Constrained RESTful Environments*) a élaboré le protocole de niveau applicatif CoAP (*Constrained Application Protocol*) (RFC 7252) [SHB14], afin de permettre une intégration plus facile des objets au web. Le protocole CoAP est une alternative au protocole HTTP (*HyperText Transfer Protocol*). Il supporte les communications multidiffusion et possède un entête de paquet très léger.

#### 1.2.2.3 ISO/CEI (International Organization for Standards/International Electrotechnical Committee)

L'ISO et l'IEC sont des organismes de normalisation qui ne sont plus à présenter. Le comité bipartie JTC1 (Joint Technical Committee 1) de l'ISO et de l'IEC est depuis sa création, à la fin des années 1980, chargé de l'élaboration des normes dans le domaine des technologies de l'information et de la communication. Ce comité hébergé par l'IEC, comprend plusieurs sous-comités pour l'élaboration des normes dans les différents sous domaines des TIC (encodage des caractères, logiciels et ingénieries systèmes, techniques de sécurisation, télécommunication et échange d'informations entre les systèmes, intelligence artificielle, IoT et technologies relatives, etc.). Le sous-comité responsable de l'élaboration des normes dans l'IoT est le SC-41 (Sub committee – 41).

Ses activités de normalisation couvrent l'IoT et les technologies relatives à l'IoT. Ainsi, à ce jour, une vingtaine de normes ISO concernant l'IoT ont été publiées et d'autres sont en cours d'élaboration [2].

2. https://www.iso.org/committee/6483279/x/catalogue/p/1/u/0/w/0/d/0

Parmi les normes publiées, nous pouvons citer :

- ISO/IEC 20924 :2018 : Le vocabulaire pour l'IoT ;
- ISO/IEC 30141 :2018 : L'architecture de référence pour l'IoT ;
- ISO/IEC TR 22417 :2017 : Les cas d'utilisation de l'IoT ;
- ISO/IEC 21823-1 :2019 : L'interopérabilité pour les systèmes IoT ;
- ISO/IEC 30101 :2014 : Les réseaux de capteurs et leurs interfaces avec le système smartgrid ;
- ISO/IEC 30140-2 :2017 : L'architecture de référence pour les réseaux de capteurs subaquatiques.

#### 1.2.2.4 IEEE (Institute of Electrical and Electronics Engineers)

L'IEEE[3] est une organisation à but non lucratif qui a été créée dans le but de promouvoir la connaissance dans les domaines de l'ingénierie électrique, électronique et informatique. Elle compte plus de quatre cent milles membres. Elle comprend plusieurs sous associations. Elle fait la promotion du savoir en permettant aux chercheurs de publier dans ses revues scientifiques. Elle élabore des normes à travers l'*IEEE Standard Association* (IEEE-SA). Ainsi, de l'avènement de l'IoT à nos jours, l'IEEE-SA a élaboré plusieurs dizaines de normes, notamment sur les technologies de communication utilisées dans l'IoT[4]. Parmi ces normes, nous pouvons citer :

- IEEE 802.11ad – 2012 : Norme IEEE pour les réseaux locaux et métropolitains. Amélioration des spécifications des couches PHY et MAC pour le très haut débit dans la bande 60 GHz ;
- IEEE 802.15.4 et ses variantes : Norme IEEE pour les réseaux sans fil personnel à faible débit (LR-WPANs : Low Rate – Wireless Personal Area Networks) ;
- IEEE 802.15.6 – 2012 : Spécifications des couches PHY et MAC des réseaux sans fil personnel (WPAN) utilisés autour ou dans le corps ;
- IEEE 802.15.7 – 2011 : Communication optique sans fil à courte portée utilisant la lumière visible ;
- IEEE 802.22 - 2011 : Spécifications de contrôle d'accès au support (MAC) et de couche physique (PHY) du RAN (*Radio Access Network*) sans fil cognitif : Politiques et procédures relatives à l'utilisation des bandes de télévision.

3. https://www.ieee.org/
4. https://standards.ieee.org/initiatives/iot/stds.html

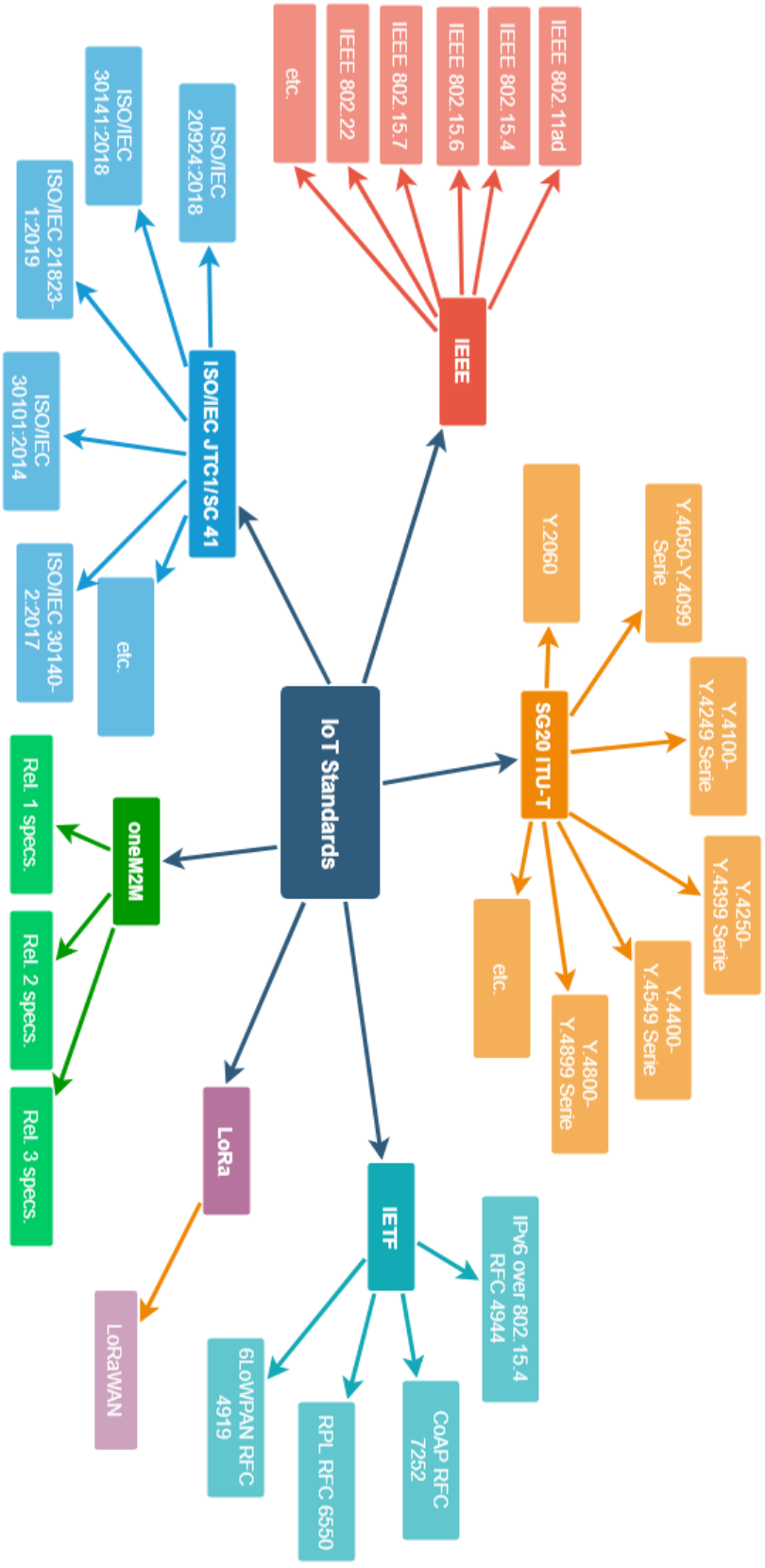



FIGURE 1.3 – Normes et organisations de normalisation dans l'IoT

#### 1.2.2.5 3GPP (Third Generation Partnership Project)

Le 3GPP est un projet de normalisation dans les télécommunications, plus précisément, les télécommunications mobiles. Il regroupe sept organisations de développement de normes de télécommunications (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC). Ses activités de normalisation sont centrées autour des technologies de communication cellulaire. Il élabore ainsi les spécifications complètes des systèmes de communication mobile, par exemple les réseaux 3G (UMTS, HSDPA/HSUPA, HSPA), 4G (LTE, LTE-A) et 5G. Parmi ces spécifications, nous avons les spécifications pour les accès radio, le cœur des réseaux mobiles, la qualité de service et la sécurité.

Le 3GPP a entrepris, à partir de 2014, des travaux de normalisation des technologies de communication mobile pour l'IoT. C'est ainsi que la norme NB-IoT (NarrowBand-IoT ou LTE Cat-NB1, LTE Cat-M2) est née [SWH17, ZM17]. Les travaux de normalisation de cette norme ont pris fin en 2016 et ont été publiés dans la release 13. C'est une technologie radio à bande étroite qui permet d'améliorer la couverture à l'intérieur des bâtiments (*indoor*), de prendre en charge un nombre important d'objets à faible débit et à faible latence, les débits dans les deux sens (montant/descendant) varient autour de 50 à 250 Kbps. Elle supporte les communications multicast, la mobilité et la continuité de service. Elle permet également aux objets qui en sont équipés de consommer moins d'énergie lors des communications.

Le 3GPP a également élaboré des normes de communication mobile pour les technologies relatives à l'IoT. Ainsi, il a publié dans ses releases 12 et 13, la norme LTE-M [5] (LTE catégorie M1 ou *Machine-Type Communication-eMTC*) pour les communications machine à machine. Cette norme permet aux objets d'atteindre un débit d'environ 1 Mbps dans les deux sens et d'avoir des sessions TCP/IP classiques, tout en consommant moins d'énergie.

La technologie 5G, cinquième génération des systèmes de communications mobiles, est une évolution majeure des standards de télécommunications mobiles. Les travaux de standardisation de la 5G ont été entrepris à partir de 2017. Ainsi, le 3GPP a publié la première série complète de standards dans sa release 15 en 2019 [3GP19]. Cette release officialise la phase 1 de la technologie 5G et constitue une étape importante dans l'atteinte des objectifs de l'IMT-2020 (*International Mobile Telecommunication*) 2020 [GMBC19].

5. http://www.3gpp.org/news-events/3gpp-news/1805-iot_r14

Elle comprend les nouveautés telles que la nouvelle interface radio (NR), l'intégration de l'Internet des Objets, la phase 2 des communications véhicules vers tout (*Vehicle-to-Everything Communication*) etc. La release 16 devra être complété en juin 2020 et devra comporter beaucoup d'améliorations, notamment sur l'interface radio et le système 5G [3GP20].

#### 1.2.2.6 oneM2M

oneM2M [6] est une organisation de normalisation internationale créée dans le but d'élaborer des normes, afin de permettre les communications machine à machine (M2M) et l'interopérabilité entre les services M2M et l'IoT à l'échelle mondiale. Elle compte parmi ses membres [7] huit organisations de normalisation internationales, à savoir : ETSI (*European Telecommunications Standards Institute*), ARIB (*Association of Radio Industries and Businesses, Japon*), TTC (*Telecommunication Technology Committee, Japon*), ATIS (*Alliance for Telecommunications Industry Solutions, USA*), TIA (*Telecommunications Industry Association, USA*), CCSA (*China Communications Standards Association*), TTA (*Telecommunications Technology Association, Corée*) et TSDSI (*Telecommunications Standards Development Society, India*). Ses travaux de normalisation répartis dans cinq groupes de travail concernent les exigences, les protocoles, la sécurité, etc.

Ses normes sont publiées sous forme d'ensembles de spécifications techniques nommés oneM2M Release specifications. Chaque Release specification contient plusieurs spécifications techniques. Ainsi, depuis sa création, elle a publié plusieurs ensembles de spécifications [8] [9] :

- oneM2M Release 1 specifications, 2014 ;
- oneM2M Release 2 specifications, 2016 ;
- oneM2M Release 2A specifications, 2018 ;
- oneM2M Release 3 specifications, 2018 ;
- oneM2M Release 4 specifications en cours d'élaboration.

La release *oneM2M Release 3 specifications* a été publiée en septembre 2018. Encore non finalisée, elle permet une interopérabilité avec les services offerts par les réseaux

6. http://www.onem2m.org/
7. http://onem2m.org/about-onem2m/partners
8. http://www.onem2m.org/technical/partner-transpositions
9. http://www.onem2m.org/technical/published-drafts

Tableau 1.1 – Classes de dispositifs selon le protocole LoRaWAN

| Classes | Descriptions |
|---|---|
| Classe A | Dispositifs finaux bidirectionnels à très faible consommation d'énergie |
| Classe B | Dispositifs finaux bidirectionnels avec latence déterministe sur la liaison descendante |
| Classe C | Dispositifs finaux bidirectionnels à latence minimale |

mobiles normalisés 3GPP et plus spécifiquement avec les réseaux NB-IoT et LTE-M. Cette interopérabilité représente une convergence très importante pour les réseaux IoT ouverts au niveau mondial car elle permet aux opérateurs IoT de déployer des réseaux IoT de plus en plus ouverts et hétérogènes sur le plan des fournisseurs de matériels et des services.

#### 1.2.2.7 Alliance LoRa

L'alliance LoRa [10] est une association à but non lucratif comptant plus de cinq cent membres. Elle a été créée en 2015 et son objectif est de promouvoir le protocole LoRaWAN à l'échelle mondiale comme la norme de communication sécurisée des réseaux étendus à faible puissance (LPWAN : *Low Power Wide Area Network*) pour l'IoT.

LoRaWAN [11] est un protocole de communication réseau à faible consommation d'énergie et sur une grande étendue géographique spécialement conçu pour connecter les objets contraints en énergie (alimentés par batterie par exemple) à Internet. C'est un protocole qui cible les besoins spécifiques de l'IoT, c'est-à-dire les communications bidirectionnelles, la mobilité, la sécurité de bout-en-bout, la localisation, etc. Il définit trois classes d'objets en fonction des besoins des applications et des caractéristiques des objets. Ces classes sont représentées dans le tableau ci-dessous. Les débits des communications varient de 0.3 à 50 Kbps. Le protocole intègre également des mécanismes cryptographiques pour sécuriser les échanges de données.

### 1.2.3 Architectures de l'Internet des Objets

L'IoT est aujourd'hui caractérisé par l'absence d'une architecture uniforme. Cela est dû aux différentes visions des différentes parties prenantes. Dans cette section, nous

10. https://lora-alliance.org/about-lora-alliance
11. https://lora-alliance.org/about-lorawan

présenterons quelques architectures proposées par certains organismes de normalisation et de recherche.

### 1.2.4 Architectures de référence

Plusieurs architectures de référence ont été proposées pour l'IoT afin de faciliter le développement de nouvelles applications. Nous décrirons deux de ces architectures, à savoir l'architecture de référence pour les systèmes industriels IoT (IIRA : *Industrial Internet Reference Architecture*) proposée par le consortium Internet Industriel (IIC : *Industrial Internet Consortium*) et l'architecture de référence proposée par l'Union Internationale des Télécommunications (UIT). En plus de ces deux architectures, nous présenterons d'autres architectures de référence, notamment celle proposée par l'ISO/IEC et celles proposées dans le cadre de projets européens et de forums.

#### 1.2.4.1 Architecture de référence de l'IIC

L'IIC (Industrial Internet Consortium) est un consortium d'entreprises composé de grands acteurs industriels évoluant dans le secteur des technologies de l'information et de la communication tels que Huawei, Intel, IBM, General Electric, Dell, Microsoft, Ericsson, etc., et de quelques universités telles que l'Université du New Hampshire, l'université de Pennsylvanie, l'université de Stuttgart, etc. Ce consortium propose, à travers son rapport technique IIRA (*Industrial Internet Reference Architecture*), une architecture de référence pour les systèmes industriels IoT [LMD+17]. Le but recherché par cette architecture de référence est de permettre aux architectes de systèmes industriels IoT de concevoir leurs systèmes IoT à partir d'un cadre, de concepts et de vocabulaires communs. Ainsi, l'architecture de référence proposée définit un système industriel IoT en termes de tiers, domaines, réseaux et flux de données (Figure 1.4). Il s'agit d'une architecture 3 tiers composée des parties suivantes : *edge tier, platform tier et enterprise tier*.

L'*edge tier* (partie bordure) comprend les nœuds de collecte de données et les différentes passerelles empruntées par ces nœuds pour l'envoi des données vers les réseaux d'accès. Il gère également les dispositifs, le contrôle de flux, etc. Le *platform tier* (partie plate-forme) a pour rôle de recevoir, traiter et transmettre les commandes de contrôle de la partie entreprise vers la partie bordure. Il est également responsable de l'analyse des flux de données en provenance de la partie bordure et fournit plusieurs fonctions

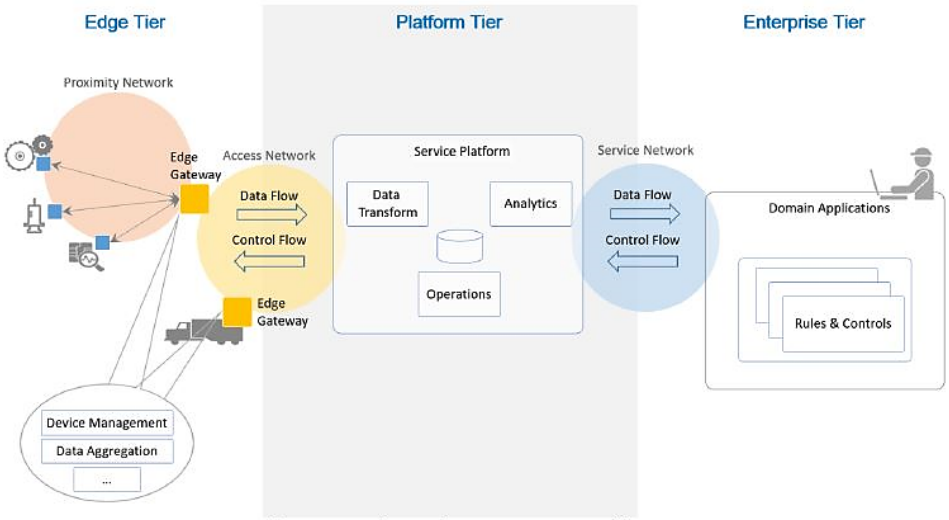



FIGURE 1.4 – Architecture de référence en 3-Tier proposée par l'IIC [LMD+17]

et services non spécifiques à un domaine particulier. L'*enterprise tier* (partie entreprise) est chargé de la mise en œuvre des applications. Il est également chargé de fournir les interfaces aux utilisateurs finaux ainsi qu'aux services d'exploitation et de maintenance des applications. Il exploite les données provenant des parties bordure et plateforme pour les applications. Il émet, à travers les interfaces, les commandes de contrôle vers les parties plate-forme et bordure.

#### 1.2.4.2 Architecture de référence de l'UIT-T

L'UIT-T a proposé à travers la recommandation Y.2060 une architecture de référence pour l'IoT [ITU12]. Cette architecture est constituée de quatre couches principales et deux couches transversales aux couches principales (Figure 1.5). Il s'agit de la couche Application, de la couche de prise en charge des services et des applications, autrement dit la couche service, de la couche réseau et de la couche dispositif, c'est-à-dire la couche des objets/détection. La couche dispositif est la couche la plus basse de l'architecture. Elle est composée des objets possédant des capacités de communication et capables de collecter et d'envoyer les données du monde physique. Elle gère également la phase d'éveil (actif/veille) des objets, cela permet aux objets de ne communiquer qu'en cas de besoin et ainsi d'économiser de l'énergie. En effet, la plupart des objets ont des contraintes énergétiques importantes. Elle permet également aux objets qui n'ont pas

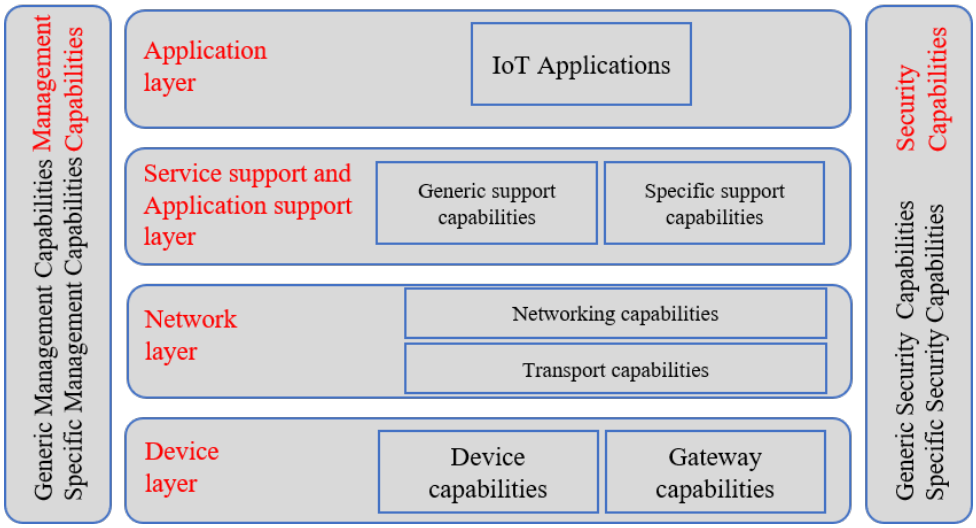



FIGURE 1.5 – Architecture de référence de l'UIT-T [ITU12]

la capacité de directement communiquer sur le réseau de communiquer à travers des passerelles. À cet effet, la passerelle réalise les conversions/adaptations de protocoles (par exemple Bluetooth/3G).

La couche réseau, située au-dessus de la couche dispositif, sert de relais entre les objets et les couches service et applications. Elle assure dans un premier temps la connectivité réseau des objets (contrôle d'accès, authentification, gestion de la mobilité, autorisation, etc.). Dans un second temps, elle assure le transport des données de chaque service/application ainsi que le transport des données de gestion et de contrôle du réseau. La couche prise en charge des services et des applications ou couche service, située au-dessus de la couche réseau, fait la liaison entre l'infrastructure et les applications. Elle gère les différents traitements et le stockage des données. En outre, elle permet de masquer les détails internes des applications et/ou de l'infrastructure ainsi que la mise en œuvre d'applications constituées d'objets et de technologies hétérogènes. La couche application est la couche supérieure de l'architecture. Elle est constituée des applications IoT, consommateurs finaux des données remontées par les objets.

Les deux couches transversales sont les couches capacités de sécurité et capacités de gestion. La couche capacités de sécurité est indépendante des applications et permet d'assurer les services de sécurité au niveau des différentes couches de l'architecture. Autrement dit, elle assure entre autres au niveau des différentes couches (dispositif, réseau, service et application) les services de sécurité suivants : authentification,

contrôle d'accès, autorisation, intégrité, confidentialité, non répudiation et disponibilité. La couche capacités de gestion permet d'assurer le bon fonctionnement du réseau IoT. Elle gère les objets (mise à jour, gestion du mode de fonctionnement, etc.), la topologie du réseau et la qualité de service (QoS) dans le réseau.

#### 1.2.4.3 Autres architectures de référence

Les architectures de référence pour l'IoT ne se limitent pas à celles abordées précédemment. Plusieurs autres architectures de référence ont été proposées par d'autres organismes. L'ISO/IEC a proposé à travers la norme ISO/IEC 30141 :2018 [12], une architecture de référence pour l'IoT. L'architecture proposée est découpée en plusieurs domaines, répartis entre quatre couches [DMRF+18]. Ce découpage en domaines permet de mieux décrire les services et les fonctionnalités qui doivent être offerts par une infrastructure IoT. L'architecture de Référence IoT-A a été proposée dans le cadre du projet Européen FP7 Lighthouse, dans le but d'apporter l'interopérabilité dans les solutions IoT, notamment au niveau de la communication et des services à travers différentes plateformes [BBDL+13]. Il s'agit d'une architecture abstraite à trois couches. Selon les auteurs, elle prend en charge les aspects commerciaux, les exigences des applications et les technologies actuelles. Une autre architecture de référence a été proposée par le forum mondial de l'IoT, organisé par Cisco et dont les participants sont entre autres AT&T, PHILIPS, VMWare, etc. [Cis14]. Il s'agit d'une architecture très granulée, c'est-à-dire constituée de sept couches.

### 1.2.5 Eléments d'une architecture IoT

Dans la suite de ce document, nous nous baserons sur l'architecture de référence proposée par l'UIT-T, c'est-à-dire une architecture IoT constituée de quatre couches, à savoir : la couche de détection, la couche réseau, la couche service et la couche application. Ce découpage nous semble très bien car il représente tous les aspects des problèmes de sécurité et de protection de la vie privée dans l'IoT. Techniquement, une architecture IoT est constituée d'une très large variété de dispositifs et de technologies de communication réseaux. La figure 1.6 illustre l'organisation en couches et les différents éléments de chaque couche. Dans les sous-sections suivantes, nous présente-

12. https://www.iso.org/standard/65695.html

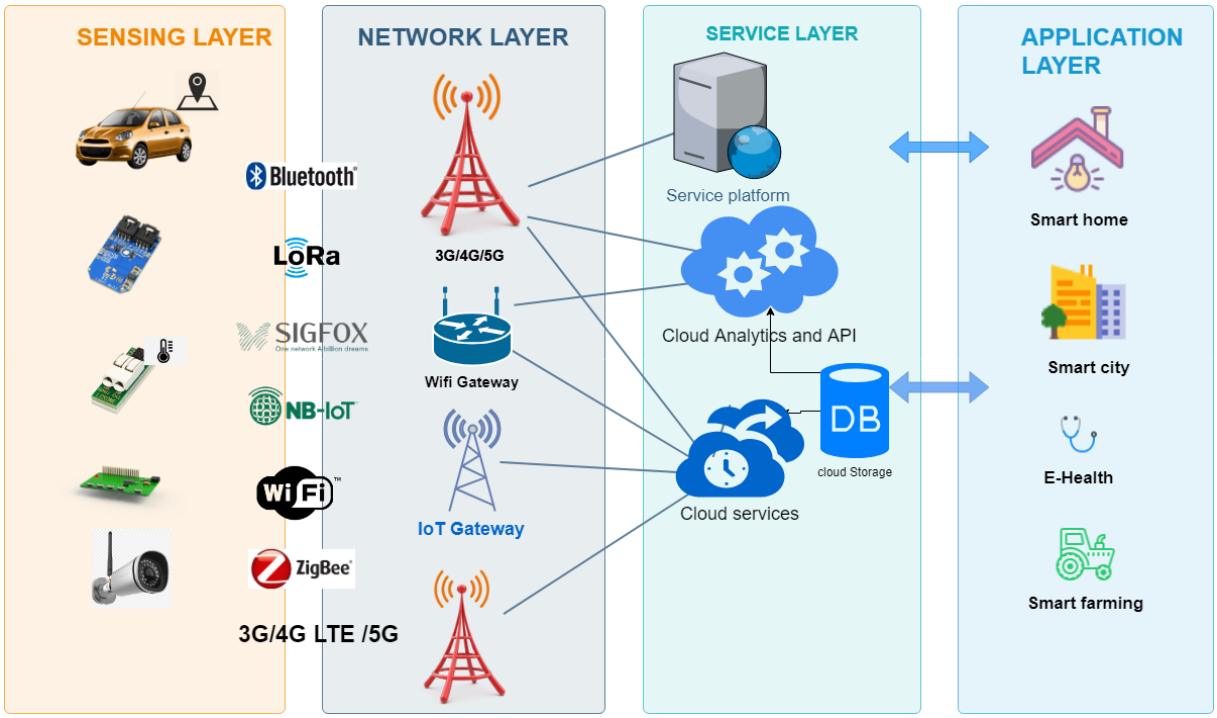



FIGURE 1.6 – Architecture IoT à quatre couches [ITU12]

rons les différents éléments et technologies généralement utilisées dans les différentes couches d'une architecture IoT.

#### 1.2.5.1 Couche de détection

La couche de détection ou sensing layer en anglais est la couche la plus basse de l'architecture. Les éléments de cette couche constituent les périphériques finaux de l'architecture. Il s'agit des objets dit "connectés et intelligents" et des capteurs. Leur rôle est de collecter des données de leur environnement au moyen de divers capteurs et de remonter via la couche réseau les données collectées à la couche service. Ces objets peuvent également agir sur leur environnement au moyen d'actionneurs. Chaque objet est globalement identifiable de manière unique. Suivant les applications, on distingue plusieurs types de capteurs : des accéléromètres, des gyroscopes, des capteurs de température, des capteurs de qualité des eaux, des capteurs du rythme cardiaque, des capteurs photo-électriques, des capteurs électro-chimiques, des capteurs à infrarouge, etc.

Suivant leurs capacités, les objets peuvent être classés en deux grandes catégories :

Tableau 1.2 – Classification des objets contraints selon leurs capacités [BEK14]

| Nom | Taille de la mémoire | Taille du code (mém. Flash) |
|---|---|---|
| Classe 0, C0 | $\ll$ 10 Ko | $\ll$ 100 Ko |
| Classe 1, C1 | $\sim$ 10 Ko | $\sim$ 100 Ko |
| Classe 2, C2 | $\sim$ 50 Ko | $\sim$ 250 Ko |

les objets contraints et les objets non contraints. Les objets contraints sont les objets possédant de faibles capacités de calcul, de stockage et d'énergie. Ils sont également limités par leurs tailles. Une sonde de température et d'hygrométrie, un Arduino nano, un bracelet connecté (smart band) sont des exemples d'objets contraints. L'IETF a dressé une classification des objets contraints dans le RFC 7228 [BEK14].

Les classes résultantes de cette classification, trois classes, sont représentées dans le tableau 1.2. Il faut également noter que ces objets, pour des questions de gestion énergétique, ne sont actifs que par intermittence, c'est-à-dire actifs que pendant le temps d'une communication. Les objets non contraints sont les objets possédant des capacités supérieures à celles des objets contraints, et non limités par leurs tailles. Les objets non contraints peuvent être directement connectés à Internet car ils possèdent des capacités leurs permettant de supporter plusieurs technologies de communication. Un Raspberry pi 3, une smart watch, un smartphone sont des objets IoT non contraints.

#### 1.2.5.2 Couche réseau

La couche réseau ou network layer en anglais, située au-dessus de la couche de détection, gère toutes les activités réseau d'une infrastructure IoT, c'est-à-dire la connectivité, le routage, l'interconnexion, l'acheminement de paquet, la qualité de service, la gestion de l'adressage et l'adaptation de protocole, etc. Elle héberge des passerelles (gateway) qui font office de passerelle vers Internet pour des objets contraints, en assurant l'adaptation de protocole et le routage des données.

Elle est également responsable du transport des données provenant de la couche de détection vers les services et applications, consommateurs de ces données. Une diversité de technologies de communication réseau est disponible dans cette couche. Ces technologies peuvent être subdivisées en trois catégories : réseaux étendus à faible puissance (LPWAN : *Low Power Wide Area Networks*), réseaux étendus (WAN : *Wide Area Network*) et réseaux locaux/personnels (LAN : *Local Area Networks* et WPAN : *Wireless Personal*

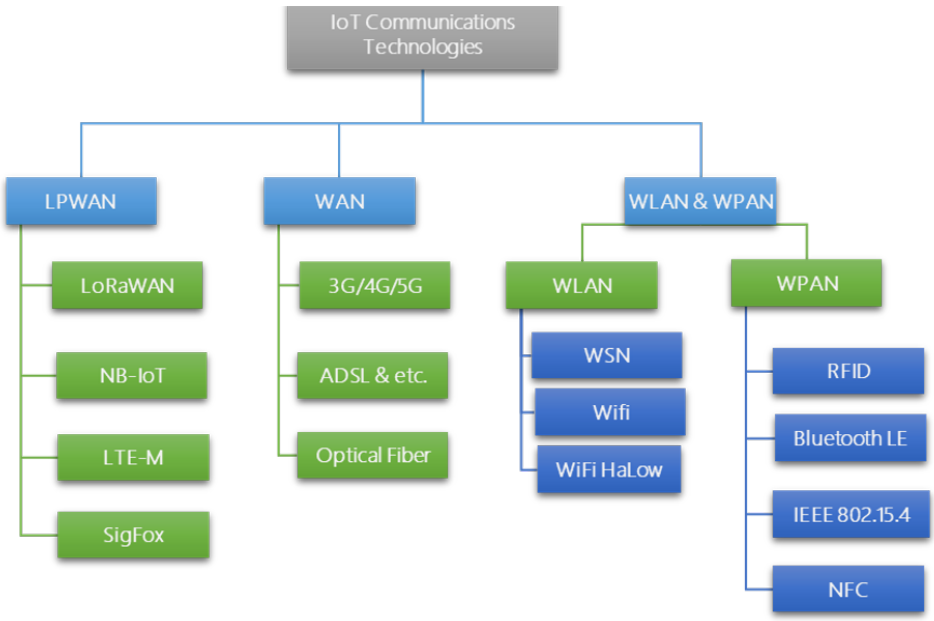



FIGURE 1.7 – Technologies de communication mises en œuvre dans l'IoT

*Area Networks*). La figure 1.7 illustre la répartition de ces différentes technologies de communication dans les différentes catégories.

Dans la catégorie des réseaux étendus à faible puissance, nous avons :

— LoraWAN (*Long Range Wide Area Networks*) de l'alliance LoRa ;

— SigFox de l'entreprise SigFox ;

— NB-IoT (*Narrowband Internet of Things*) du 3GPP [SWH17] ;

— LTE-M (*Long Term Evolution – M*) du 3GPP.

Dans la catégorie des réseaux étendus, nous avons :

— les réseaux mobiles 3G/4G/5G ;

— les réseaux haut débit filaires (ADSL, Fibre optique).

Dans la troisième et dernière catégorie, nous avons :

— les technologies de réseaux de capteurs (WSN : *Wireless Sensor Network*) [GBMP13] ;

— les technologies de communication pair à pair à courte portée :

  — BLE : *Low Energy Bluetooth* [GOP12] ;

  - IEEE 802.15.4 [IEE16] ;
- les technologies de communication à très courte portée :
  - RFID [SS17] ;
  - NFC (*Near Field Communication* [BRKK18]) ;
- et les technologies de communication à moyenne portée en réseau local :
  - WiFi HaLow IEEE 802.11ah (Low Power WiFi [Sun16]) ;
  - les variantes des normes sans fil WiFi.

Les protocoles réseaux mis en œuvre dans cette couche sont notamment le protocole 6LoWPAN (*IPv6 over Low-Power Wireless Personal Area Networks*) standardisé par l'IETF, le protocole RPL (R*outing Protocol for Low-power and Lossy Network*) développé par le groupe de travail ROLL (*Routing Over Low-power and Lossy-networks*) au sein de l'IETF [BM07, BRKK18, LXI17].

#### 1.2.5.3 Couche service

La couche service ou service layer, située au-dessus de la couche réseau, fournit et gère les services requis par les utilisateurs et les applications de l'architecture. A cet effet, elle héberge les services de stockage, de traitement et d'analyse d'informations des données remontées par la couche de détection. Elle est également responsable de la mise en œuvre des logiques métiers des applications. Cela nécessite l'extraction des informations nécessaires à partir de l'énorme quantité de données collectées par les objets. Les services que l'on retrouve à ce niveau sont généralement les services d'analyse des données Big Data, les services d'apprentissage automatique et les services d'interfaces de programmation d'application (API : *Application Programing Interface*).

Nous avons également, au niveau de cette couche, le service de support opérationnel, de configuration et de mise à jour des dispositifs (les objets), de reporting et de facturation. C'est également au niveau de cette couche que sont hébergés le service de gestion de la qualité de service et la plupart des opérations de sécurité (bien que la sécurité soit transversale à toutes les couches) tels que le contrôle d'accès, la gestion de l'identification et la gestion des processus métiers.

#### 1.2.5.4 Couche application

La couche application ou application layer est la couche la plus haute de l'architecture. Elle exploite les fonctionnalités offertes par la couche service. Elle comprend également les méthodes d'interaction du système avec les utilisateurs finaux. Ainsi, on y retrouve diverses applications basées sur les objets connectés. Parmi elles, nous avons les maisons intelligentes, la e-santé, l'industrie 4.0, le smart grid, etc.

### 1.2.6 Couches du modèle de communication

Les objets de l'IoT doivent communiquer avec les infrastructures existantes. Dans ces systèmes, la pile de protocoles TCP/IP est utilisée pour permettre la communication entre deux ou plusieurs entités sur Internet. Les contraintes de ressources sur la majorité des objets ont poussé le développement d'une pile de protocoles de communication adaptée aux objets connectés, leur permettant de communiquer avec de faibles ressources de calcul et d'énergie. Les efforts de développement de cette pile de protocoles ont été conduits par les organismes de normalisation et de recherches tels que l'IEEE et l'IETF [BRKK18, GMSS23]. La figure 1.8a illustre cette pile de protocoles et la figure 1.8b les protocoles que l'on retrouve au niveau de chaque couche de la pile. Elle comprend six couches :

— la couche physique ;
— la couche MAC ;
— la couche d'adaptation,
— la couche réseau ;
— la couche transport ;
— et la couche application.

#### 1.2.6.1 Couche physique et couche MAC

Le norme 802.15.4 a été conçue pour permettre des communications consommant une très faible quantité d'énergie [IEE20a], sa dernière version normalisée étant le 802.15.4z-2020 [IEE20b]. A cet effet, elle définit les spécifications techniques de la couche physique PHY et de la sous-couche MAC (*Medium Access Control*) pour la connectivité sans fil, à faible débit et à très faible consommation d'énergie des objets contraints

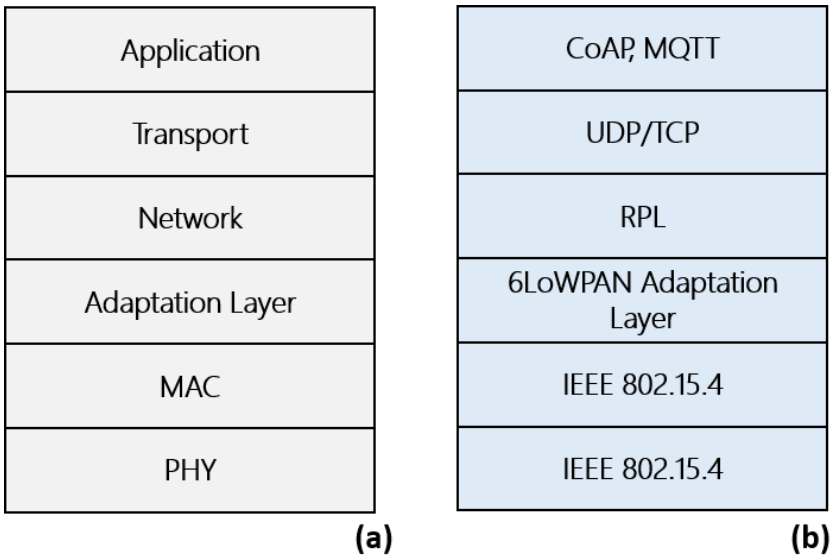



FIGURE 1.8 – Pile protocolaire et protocoles de communication dans l'IoT

énergétiquement (batterie limitée) et à coûts très bas. En effet, dans les environnements contraints, les protocoles possédant une faible bande passante, une puissance de transmission limitée et une faible consommation d'énergie sont requis.

Les nœuds peuvent communiquer suivant deux topologies principales : la topologie égale à égale ou peer-to-peer et la topologie en étoile. La taille des paquets est limitée à 127 octets avec un débit de transmission maximum de 250 Kbit/s [IEE16]. Cela est dû à la portée limitée des communications. La norme spécifie également des mécanismes de transmission robuste permettant la retransmission des paquets perdus.

#### 1.2.6.2 Couche d'adaptation

Les communications IEEE 802.15.4 laisse une possibilité de charge utile de 102 octets pour la transmission des données des protocoles de couches supérieures. Cependant, cette valeur est très loin du MTU (*Maximum Transmission Unit*) d'un paquet IPv6, c'est-à-dire la taille maximale d'un paquet IPv6 qui vaut 1280 octets [GMSS23].

L'objectif principal de la couche d'adaptation 6LoWPAN est de permettre la transmission de paquets IPv6 sur l'IEEE 802.15.4. IPv6 est aujourd'hui un protocole Internet plus stable, plus riche en fonctionnalités et qui fournit un très grand espace d'adressage capable de supporter l'IoT. Spécifiquement, la couche d'adaptation 6LoWPAN permet de compresser les entêtes de paquets IPv6, de fragmenter et de réassem-

bler les paquets IPv6 en des paquets transmissibles par IEEE 802.15.4 et de permettre la communication entre les nœuds d'un réseau 6LoWPAN (adressage, routage, etc.). [BRKK18, GMSS23, KM07]. Les objets IoT utilisant 6LoWPAN peuvent directement communiquer sur Internet moyennant l'utilisation d'une passerelle IPv4/IPv6.

#### 1.2.6.3 Couche réseau

La couche réseau gère l'acheminement des données à travers le réseau en assurant le routage des paquets de données entre les nœuds du réseau. Le protocole de routage mis en œuvre pour les réseaux IoT exploitant 6LoWPAN est le protocole RPL. RPL est un protocole de routage ouvert à vecteur de distance qui a été développé par le groupe ROLL de l'IETF [VAH+11]. Il a été spécialement conçu pour les réseaux à perte et à faible puissance (LLN : *Low power and Lossy Networks*). Il spécifie comment construite le graphe de routage DODAG (*Destination Oriented Directed Acyclic Graph*) en utilisant un ensemble de fonctions et de métriques [VAH+11]. Ce graphe permet d'atteindre les différentes destinations à partir des différents nœuds. En plus du routage, il fournit un cadre adaptable aux exigences des applications [GMSS23]. Par exemple, plusieurs graphes peuvent être établis pour une destination en fonction des exigences telles que la sécurité des liens, la latence, etc.

#### 1.2.6.4 Couche transport

La couche transport assure un service de transport de bout en bout transparent pour les applications. Dans la couche transport de la pile de protocoles TCP/IP, deux protocoles sont habituellement utilisés : TCP (*Transmission Control Protocol*) et UDP (*User Datagram Protocol*). TCP n'est pas recommandé pour les objets contraints parce qu'il comporte beaucoup de surcharge relative au contrôle de flux et du fait que c'est un protocole orienté connexion, mais il peut être utilisé. En revanche, UDP est préféré parce qu'il est sans connexion et qu'il ne comporte pas de contrôle de flux.

#### 1.2.6.5 Couche application

La couche application est chargée de formater et de présenter les messages provenant du réseau. Elle est également à l'origine des messages envoyés sur le réseau. Les

protocoles applicatifs classiques que l'on retrouve dans l'Internet tels que HTTP (*Hyper-Text Transfer Protocol*) ne sont pas adéquats pour la plupart des objets connectés. Ainsi, plusieurs protocoles applicatifs adaptés aux contraintes de capacités des objets connectés ont été développés. On distingue : CoAP (*Constraint Application Protocol*)[SHB14] et MQTT (*Message Queue Telemetry Transport*)[Mos14].

CoAP a été proposé par l'IETF afin de permettre l'interopérabilité au niveau de la couche application [SHB14, GMSS23]. Il est basé sur UDP et fonctionne suivant l'architecture web REST (REpresentational State Transfer). Les données sont échangées dans un format binaire et très léger, c'est-à-dire EXI (Efficient XML Interchanges) [BRKK18]. Ainsi, les objets exécutant CoAP opèrent comme des serveurs de ressources et les applications consommateurs des données comme des clients demandeurs de ressources. Le protocole MQTT, développé à l'origine par IBM, existe en plusieurs versions (propriétaires et libres). C'est un protocole client/serveur basé sur le protocole TCP. Les clients opèrent comme des publieurs/abonnés (pub/sub) et le serveur fait office de courtier (broker). Le serveur distribue aux abonnés les messages auxquels ils ont préalablement souscrit. Les messages envoyés aux abonnés proviennent des publieurs. Compte-tenu des capacités restreintes des objets contraints, une version plus allégée de MQTT, MQTT-S/MQTT-SN a été proposée [BRKK18, HTS08].

### 1.2.7 Domaines d'applications de l'IoT

Les avantages offerts par l'Internet des objets couvrent un grand nombre d'applications. Ces applications permettent d'améliorer la vie des personnes : à la maison, au travail, en déplacement, à l'hôpital, etc. La figure 1.9 illustre quelques domaines d'applications phares de l'IoT.

#### 1.2.7.1 E-santé

Les applications de l'IoT sont particulièrement bénéfiques pour le suivi de l'état de santé d'une personne. Elles sont constituées d'un certain nombre d'objets comprenant des capteurs de tout genre permettant d'avoir en tout temps, en tout lieu, les éléments comme : le taux de sucre dans le sang, la tension artérielle, le rythme cardiaque, la qualité du sommeil, les activités physiques, la qualité du sang, le niveau de respiration, etc. La e-santé est l'intersection de la santé, de l'informatique médicale et des appareillages électroniques qui permet d'utiliser les informations fournies pour améliorer la

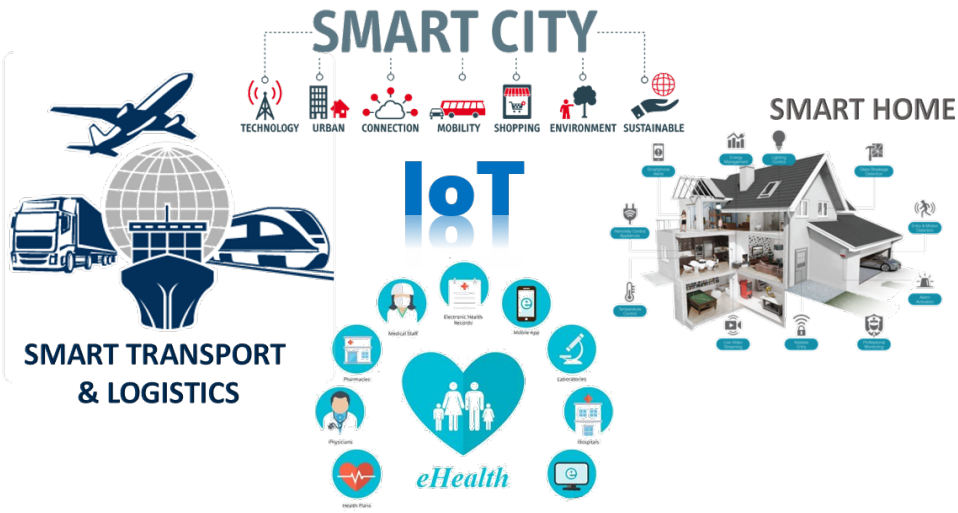



FIGURE 1.9 – Quelques domaines d'applications de l'IoT
*Crédits de l'image : Psi Matrix, Medium.com, openPR.com, Clever.fr*

santé des personnes [Eys01]. Elle permet de mieux prendre soin des bébés, des personnes âgées, des diabétiques, des personnes souffrant d'hypertension artérielle, etc. Elle permet également de réaliser un suivi régulier d'un patient en ayant son dossier médical électronique à jour à tout instant [KJ17, SS17]. Par exemple, Yang et al. ont dans [YZL+16] proposé un système de suivi de l'électrocardiogramme d'un patient. À travers cette application, il est possible de détecter une crise de tachycardie ou une crise cardiaque et d'envoyer une équipe de secours sur place le plus rapidement possible.

En fitness, les applications de suivi des activités physiques journalières d'une personne sont disponibles. Les données des activités physiques journalières d'une personne sont collectées à travers un bracelet intelligent, équipé d'accéléromètre, de capteur d'ECG, etc. Ces applications analysent les données collectées par le bracelet par le moyen d'algorithmes complexes et donnent des conseils et des recommandations aux porteurs. Les données collectées par les objets de plusieurs patients peuvent également être analysées dans le *Cloud* à travers les algorithmes Big Data : fouille de données, apprentissage automatique et intelligence artificielle [JJB12].

#### 1.2.7.2 Maisons intelligentes ou *smart homes*

Les maisons intelligentes ou *smart homes* en anglais, sont l'une des applications phares de l'IoT. Une maison intelligente est une maison dont les systèmes d'éclairage, de

chauffage et d'appareils électroniques et de sécurité peuvent être commandés à distance par un smartphone ou un ordinateur. Les applications smart home permettent d'offrir des services automatisés à l'utilisateur. Pour ce faire, plusieurs capteurs de divers types sont déployés dans la maison. Ce sont, entre autres, les détecteurs de fumées, les thermostats, les caméras, les détecteurs de mouvement, les détecteurs d'intrusion connectés. Ils aident à économiser de l'énergie en éteignant les lumières, les écrans et autres produits électroniques d'usage domestique de manière automatique dès qu'ils ne sont plus utilisés [KAA+18].

Certains capteurs et objets peuvent être utilisés pour assurer la sécurité de l'utilisateur en mettant en œuvre des alarmes anti-intrusion dans la maison. Les maisons intelligentes permettent également d'automatiser certaines tâches de l'utilisateur comme l'arrosage du jardin, le fonctionnement des stores, etc. Cela permet de faciliter la vie de l'utilisateur. La maison intelligente peut être très bénéfique pour les personnes âgées. Lorsqu'elle est combinée aux capteurs de la e-santé, elle peut permettre un suivi plus rapproché des personnes âgées, telles que la détection des chutes, des crises cardiaques, etc. [RNWK14]. Par exemple, les secours seront immédiatement alertés si la personne chute ou si elle a une attaque.

#### 1.2.7.3 Villes intelligentes

Selon le département des affaires économiques et sociales des Nations Unies (UN DESA : United Nation Department of Economic and Social Affairs), approximativement 68% de la population mondiale vivra dans les villes en 2050 [13]. Cet accroissement rapide de la population urbaine exerce une pression importante sur les infrastructures existantes. La situation ne fera qu'empirer dans les années à venir. Les innovations qu'apportent les applications IoT pour les villes permettront de répondre à plusieurs défis de taille pour les villes et cela de manière durable, rendant ces villes "intelligentes", car elles pourront s'adapter en fonction des situations. Une ville est intelligente si elle utilise les ressources informatiques et technologiques pour améliorer le service public et le cadre de vie, informer ses citoyens et accroitre son efficacité. Les applications IoT pour les villes intelligentes permettent de mieux gérer l'eau et de réaliser des économies d'énergie dans les immeubles et bâtiments. Par exemple, les compteurs d'eau intelligents

13. https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html

peuvent détecter les fuites et alerter les responsables de l'eau de la ville.

Les systèmes de gestion énergétique intelligents des immeubles utilisent des objets interconnectés pour coordonner le fonctionnement des systèmes de climatisation, d'éclairage et de sécurité incendie afin de mieux utiliser l'énergie disponible et réduire le gaspillage de l'énergie. Ces systèmes sont également interconnectés avec d'autres systèmes IoT qui permettront une intervention rapide du personnel de secours en cas de problèmes. L'IoT permet aux villes d'améliorer la sécurité des citoyens. En effet, à travers un grand nombre de caméras de surveillance intelligentes, les vidéos seront analysées en temps réel et les incidents (accident, braquage, vol, etc.) seront détectés et les alertes automatiquement envoyées aux responsables concernés par ces incidents. Cela est possible notamment grâce aux supports de l'intelligence artificielle et de l'apprentissage automatique. L'IoT permettra également aux villes d'améliorer le système de collecte des déchets avec l'usage des poubelles et de dépôt de transit connectés.

#### 1.2.7.4 Transport et logistique

Les applications de l'IoT sont nombreuses dans le domaine du transport et de la logistique [LL15]. Les applications de transport intelligents (ITS : *Intelligent Transport System*) permettent aux véhicules de se déplacer sur les routes en toute sécurité. Les véhicules sont de plus en plus équipés de technologies IoT (IoV : *Internet of Vehicles*), leur permettant d'échanger entre eux plusieurs informations telles que leurs positions, leurs vitesses, leurs directions, etc. [SS17]. Ainsi, grâce aux informations échangées dans ces réseaux véhiculaires, les véhicules sont capables de détecter les accidents et d'éviter les collisions [KS13, SS17].

L'un des défis majeurs posé à une ville est la gestion de la congestion du trafic. Les feux de circulation intelligents peuvent ajuster leur cadence pour s'adapter aux déplacements des voitures en fonction de la période (par exemple aux heures de pointe) et de l'afflux de voitures. Les responsables de la gestion des métropoles peuvent collecter et regrouper des données provenant de caméras de surveillance du trafic, de véhicules et d'autres capteurs pour la surveillance en temps réel des incidents de la circulation. Les conducteurs peuvent être avertis des accidents et dirigés vers des itinéraires moins encombrés.

Dans la logistique, les véhicules de transport de marchandises sont équipés de tech-

nologies de géolocalisation, GPS (*Global Positioning System*) ou Galileo, permettant leur suivi en temps réel et d'agir sur leurs itinéraires en cas de problème. Les marchandises sont équipées de technologies RFID. Cela permet leur identification rapide et également d'établir leur traçabilité. L'IoT permet également aux entreprises de facilement gérer leurs flottes de véhicules de transports de marchandises à travers des puces GPS ou Galileo. Il faut également noter que la logistique a été l'une des idées précurseurs de l'IoT au moment où le concept a été évoqué par K. Ashton [McF15].

#### 1.2.7.5 Industrie 4.0

L'industrie 4.0 ou smart factory ou encore système de fabrication intelligent est un système intégré et collaboratif qui répond aux demandes et aux conditions variantes de l'usine, du système d'approvisionnement et des besoins des clients, et cela en temps réel. Cela est possible grâce aux capteurs et actionneurs sophistiqués et connectés en utilisant les technologies IoT dans le processus de fabrication mais également au pilotage des activités de production par l'intelligence artificielle. De plus, avec l'interconnexion aux systèmes extérieurs, les machines peuvent directement entamer la production dès qu'un bon de commande est reçu par l'usine. De par son importance, ce domaine d'application de l'IoT a fait l'objet de plusieurs travaux de normalisation [TTHG+17].

### 1.2.8 Quelques caractéristiques de l'Internet des Objets

Dans les sous-sections suivantes, nous explorerons quelques caractéristiques propres à l'IoT, à savoir : l'intelligence, les données sensibles, la sensibilité au contexte, l'hétérogénéité, l'adaptation, la forte densité d'objets, les ressources faibles et le temps réel.

#### 1.2.8.1 Intelligence

L'intelligence signifie l'application de la connaissance dans l'IoT. Elle est généralement mise en œuvre par une association d'algorithmes, de techniques de traitement de données évolués (apprentissage automatique, raisonnement) et de capacité de calcul. Cette intelligence est présente au niveau de la couche service d'une architecture IoT.

#### 1.2.8.2 Données sensibles

Selon le Règlement Général sur la Protection des Données (RPGD) de l'Union Européenne, une donnée sensible est définie comme suit : "*une donnée est considérée comme sensible si elle peut révéler l'origine ethnique ou raciale d'une personne, sa croyance religieuse et ses opinions politiques. Est également considérée comme donnée sensible, le code génétique d'une personne, des données biométriques servant à l'identifier, ses données de santé et sa vie sexuelle ou ses orientations sexuelles* [Cou16]". En plus de ces données personnelles, d'autres données collectées sur la personne peuvent également être considérées comme sensibles, comme par exemple, sa localisation, ses préférences, etc.

Les données collectées par les capteurs et les dispositifs IoT sur les utilisateurs sont alors pour la plupart des données sensibles. Par exemple, dans la e-santé, les capteurs ECG collectent les données sur le rythme cardiaque des utilisateurs. Les tensiomètres et les glucomètres collectent les données sur la tension artérielle et sur le taux de sucre dans le sang des utilisateurs. Parmi les données sensibles, nous avons la localisation géographique. Dans les applications de géolocalisation, les capteurs GPS collectent et transmettent en continu les données sur la localisation des utilisateurs. A partir des données de localisation, il est possible de traquer une personne, de savoir les lieux de cultes fréquentés, par la même son orientation religieuse. Certaines applications pour des fins de fiabilité nécessitent la fourniture des informations telles que le pays d'origine, la couleur de la peau, etc. Toutes ces données sont alors des données sensibles. La collecte, le transfert, le traitement et le stockage de ces données sensibles requièrent une protection afin qu'elles ne tombent pas entre de mauvaises mains.

#### 1.2.8.3 Sensibilité au contexte

L'IoT est principalement animé par la collecte de données des capteurs des objets connectés. Cette collecte reflète les changements dynamiques qui ont lieu dans l'environnement des objets. Les objets doivent également faire face aux changements liés à leur contexte (changement de localisation géographique, etc.), au nombre d'objets présents (ajout ou retrait d'objets à la volée) et à la configuration réseau (par exemple changement fréquent de réseau d'accès (Wifi, 4G, etc.), d'adresse IP, etc. L'état des objets change également dynamiquement, avec des temps d'activité et de connectivité sporadiques les faisant passer de l'état veille à l'état actif ou encore de l'état connecté à

l'état déconnecté.

#### 1.2.8.4 Hétérogénéité

L'IoT est caractérisé par des objets et des systèmes basés sur des configurations matérielles, logicielles et réseaux très différentes. Ces objets peuvent interagir entre eux ou avec différentes plateformes de services et cela via des réseaux hétérogènes. Les applications et systèmes à mettre en œuvre devront prendre en charge cette hétérogénéité de l'IoT, c'est-à-dire permettre l'interopérabilité et la modularité.

#### 1.2.8.5 Adaptation

Les systèmes IoT doivent cohabiter avec les systèmes et infrastructures informatiques environnants. Ainsi, un système IoT requiert de la flexibilité pour pouvoir s'intégrer à ces systèmes et infrastructures existants sans besoin de changements dans ces derniers.

#### 1.2.8.6 Densité d'objets élevée

L'IoT est caractérisé par le nombre important d'objets qui seront connectés. Selon les prévisions, environ dix-huit milliards d'objets seront connectés d'ici 2022 [Eri19]. Les applications et infrastructures IoT à mettre en place devront permettre le passage à l'échelle.

#### 1.2.8.7 Ressources restreintes

Les objets sont les éléments essentiels d'une architecture IoT. Ils collectent et remontent les données du monde physique vers les autres parties de l'infrastructure. Le faible coût et les contraintes des environnements dans lesquels ces objets doivent être déployés jouent sur les capacités de ces objets. Cela se traduit par des ressources (processeur, mémoire vive) restreintes au niveau de la majorité des objets [BEK14].

#### 1.2.8.8 Temps réel

Certaines applications IoT ont besoin de traiter les données collectées en temps réel, d'autres non, telles que les données se rapportant à la croissance des plantes ou permettant l'analyse des sols en agriculture. Les applications dont l'objectif est de permettre

une nouvelle expérience utilisateur, un suivi plus précis ou d'avoir un système critique plus réactif sont celles qui nécessitent un traitement en temps réel des données. Cela va de l'analyse de flux de données en continu pour le monitoring à l'adaptation des commandes et contrôles, etc. Par exemple, les dispositifs de e-santé qui suivent les signes vitaux d'un patient permettent aux médecins de détecter en temps réel les changements dans l'état de santé du patient, de réagir rapidement en adaptant le traitement ou en cas d'urgence d'envoyer une ambulance chez le patient. Dans une ville intelligente, l'analyse en temps réel des données sur la consommation d'énergie envoyées par les compteurs intelligents permet aux services publics d'adapter la fourniture d'énergie afin de minimiser le gaspillage, de réduire l'impact environnemental et de permettre aux citoyens d'économiser de l'argent.

Le suivi en temps réel des mouvements de produits permet aux entreprises de surveiller le stock et de déclencher automatiquement la procédure de réapprovisionner du stock avant une rupture. Les dispositifs IoT intégrés dans les camions frigorifiques transportant des denrées périssables (poisson, fruits, etc.) peuvent aider à surveiller la chaîne du froid en mesurant continuellement la température et l'humidité à l'intérieur des camions et en cas d'augmentation anormale des températures avertissent le conducteur et l'entreprise concernée. Ensuite, grâce aux dispositifs IoT de géolocalisation, les clients peuvent savoir la position du camion en temps réel. Les exemples cités ci-dessus illustrent les besoins temps réel de certaines applications IoT.

## 1.3 Sécurité dans l'Internet des Objets

Malgré les nombreux avantages de l'IoT, les défis qui freinent son développement sont nombreux. Dans les sous-sections suivantes, nous présenterons les défis sécuritaires dans l'IoT. Nous décrirons également les vulnérabilités et les attaques menaçant la vie privée et le bon fonctionnement des applications IoT.

### 1.3.1 Défis sécuritaires dans l'Internet des Objets

La sécurité et la protection de la vie privée sont parmi les défis les plus importants à relever dans l'Internet des Objets (Figure 1.10) [ODO17]. Selon la littérature, plusieurs risques de sécurité dans l'IoT proviennent principalement de la vulnérabilité des objets [Woo16]. Ces vulnérabilités sont essentiellement dues aux contraintes liées à leurs performances restreintes et à la non prise en charge des questions de sécurité

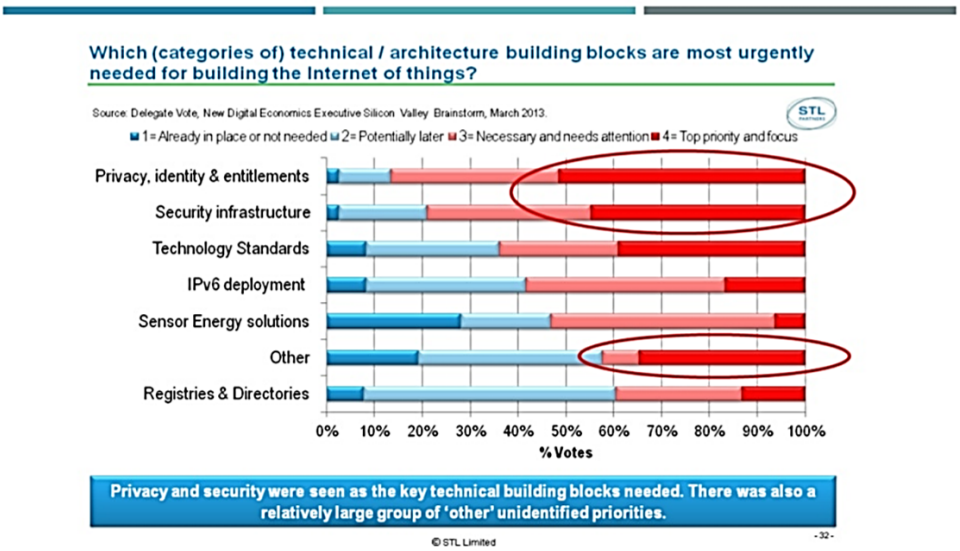



FIGURE 1.10 – Défis à relever dans l'Internet des Objets [Pip14]

dès leur conception. Ces vulnérabilités sont exacerbées par leur omniprésence. Ils facilitent les accès non autorisés aux informations personnelles et l'accès à d'autres parties des réseaux. Autrement dit, ils constituent une grande surface d'attaque. Par exemple, un objet IoT compromis peut permettre à un pirate de lancer une attaque de déni de service. Le déni de service peut devenir distribué lorsque l'attaquant prend le contrôle de plusieurs milliers d'objets. Ce fut le cas par exemple en 2016, lorsqu'un pirate a pris le contrôle de plusieurs centaines de milliers d'objets IoT infectés par un malware du nom de Mirai [Woo16]. L'attaque avait mis plusieurs serveurs de l'opérateur Dyn hors service pendant plusieurs heures, causant une coupure de l'Internet sur toute une partie des Etats-Unis. Un autre risque lié aux vulnérabilités des objets connectés est le risque d'attaque personnelle, certains objets jouant un rôle vital chez leurs porteurs. Par exemple, un régulateur cardiaque connecté pourrait être la cible d'un attaquant visant l'assassinat de son porteur.

Pour les risques liés à la protection de la vie privée, les objets connectés collectent d'énormes quantités de données. Selon un rapport de la Federal Trade Commission, 10 000 ménages utilisant des objets connectés de la maison intelligente peuvent générer jusqu'à 150 millions de points de données discrets par jour [FTC15]. Ces données contiennent souvent des informations sensibles. Plus il y a de données, plus il est possible d'inférer et de lier les données entre elles pour établir un profil. Pour sensibiliser

et avoir le consentement des utilisateurs, les entreprises proposent de longues clauses de confidentialité. Généralement, les utilisateurs ne font que défiler les clauses de confidentialité pour finir par en accepter tous les termes. Les compagnies peuvent utiliser les données ainsi collectées, sans que les utilisateurs aient pris réellement connaissance de la clause les autorisant à utiliser leurs données comme bon leur semble. Par exemple, une compagnie d'assurance santé peut utiliser les données des utilisateurs pour évaluer leurs primes d'assurance. Un autre problème auquel la vie privée est confrontée dans l'IoT est la possibilité pour les pirates ou les gouvernements de pratiquer du eavesdropping, c'est-à-dire, de virtuellement envahir la maison d'une personne, suivre ses mouvements et faits et gestes, voire prédire ou il devra se rendre, en un mot "espionner".

### 1.3.2 Menaces dans les différentes couches d'une architecture IoT

Une vulnérabilité est une faiblesse ou une lacune dans un système. C'est également un manque dans les efforts de protection du système. Une menace est toute possibilité d'exploitation d'une vulnérabilité par un individu, et cela de manière intentionnelle ou accidentelle, dans le but d'obtenir, d'endommager ou de détruire un bien. Un risque exprime la possibilité de perte, de dommage ou de destruction d'un bien (information sensible, une réputation, etc.) à la suite de l'exploitation d'une vulnérabilité, par un individu, et cela de manière accidentelle ou intentionnelle.

Toute couche d'une architecture IoT (section 1.2.7) comporte plusieurs failles de sécurité. Les attaques exploitant ces failles peuvent viser les différents protocoles de communication (liaison de données, réseau, etc.), ainsi que les applications et les services IoT. Ces menaces proviennent généralement de vulnérabilités connues, de failles zéro-jour (ou Zero-day en anglais) et profitent de la capacité restreinte des objets, de leur emplacement physique, etc. Une faille zéro-jour est une vulnérabilité découverte mais n'ayant fait l'objet d'aucune publication ou n'ayant pas de correctif connu. Une attaque exploitant cette vulnérabilité est appelée exploit zéro-jour.

Par ailleurs, les outils de découverte automatique de failles et de vulnérabilités des objets IoT sont disponibles sur Internet et sont facilement accessibles. Ces outils donnent la possibilité à des personnes même peu qualifiées de détecter les failles et les vulnérabilités des objets IoT, où que l'objet se trouve dans le monde. Par conséquent, ils augmentent les risques d'attaques envers la sécurité des applications IoT et la vie privée des utilisateurs. Parmi ces outils, nous avons Shodan , un moteur de recherche d'ob-

jets vulnérables connectés à Internet. Plus récemment, un outil du nom d'Autosploit a été publié sur Internet [Sas18]. Autosploit utilise le moteur de recherche Shodan pour trouver les objets vulnérables et permet par la suite de facilement mener des attaques d'exploitation de masse des objets trouvés sans que l'attaquant ne sache la localisation des objets. Dans [GKS18, LXI17], les auteurs ont dressé une liste des menaces les plus courantes pour l'IoT. Dans [MS18], OWASP (*Open Web Application Security Project*) a également dressé une liste des vulnérabilités connues de l'IoT. Nous décrirons ces différentes menaces dans les sous-sections suivantes.

#### 1.3.2.1 Couche de détection

Les objets présents sont pour la plupart conçus pour une faible consommation d'énergie avec des ressources contraintes. Au niveau de cette couche, nous avons plusieurs vulnérabilités héritées des systèmes de communication mais également des vulnérabilités dues aux spécificités des objets connectés. Les protocoles de communication exposés aux attaques sont les protocoles de liaison de données (par exemple IEEE 802.15.4, etc.). Ces attaques peuvent être réparties en deux catégories : les attaques pouvant compromettre les objets ou leurs disponibilités et les attaques pouvant compromettre les communications, c'est-à-dire les attaques auxquelles la collecte et la remontée des informations sont exposées.

(i) **Attaques contre les objets** : Ces attaques ne sont réalisables que lorsque l'attaquant a un accès physique aux objets. Elles sont généralement difficiles à réaliser.

— **Attaque contre la disponibilité (capture physique)** : Certains objets sont déployés dans des lieux non protégés. Un attaquant peut rendre un objet inaccessible en capturant l'objet physiquement ou en le détruisant. Ces attaques pouvant compromettre la disponibilité des objets sont également appelées attaques contre l'intégrité physique.

— **Accès non autorisé et extraction de données sensibles** : Un attaquant peut physiquement accéder à un objet non protégé. Il peut alors extraire des informations sensibles stockées sur l'objet. Les attaques auxquelles les objets sont exposés sont les attaques par canal latéral ou *side channel attacks*. Parmi les attaques par canal latéral, nous avons les attaques par analyse de la consommation électrique, et l'attaque par injection de faute. Une attaque par analyse

de la consommation est une attaque par laquelle un attaquant cherche à retrouver des informations sensibles (par exemple une clé privée) à partir de la consommation de courant lors des opérations cryptographiques. Une attaque par injection de faute consiste à causer une perturbation dans l'exécution d'un algorithme, dans le but d'extraire des informations, cela en utilisant des moyens tels que la surchauffe du processeur, l'émission de laser, etc.

— **Micrologiciel vulnérable** : Les objets connectés reposent sur des micrologiciels. Un micrologiciel (firmware en anglais) est un programme intégré dans la mémoire d'un objet afin d'assurer son bon fonctionnement. Certains micrologiciels peuvent comporter des erreurs d'exécution ou des failles au niveau de leurs implémentations. Les vulnérabilités dues à ces bugs plus ou moins critiques exposent les objets à plusieurs menaces, par exemple à l'attaque par débordement de mémoire tampon (*buffer overflow*). L'application des correctifs de ces bugs de firmwares à travers les mises à jour OTA (*On The Air*), via des canaux non sécurisés expose d'avantage les objets à des menaces de clonage. En effet, un attaquant peut capturer une copie du firmware, pour ensuite reproduire un objet possédant les mêmes fonctionnalités que l'objet auquel le firmware était destiné. De plus, il peut ajouter des portes dérobées à l'objet cloné et remplacer un objet légitime d'une application IoT par cet objet cloné.

— **Clonage et Usurpation d'objet** : Il est possible que certains fabricants peu scrupuleux lors du processus de fabrication clonent les caractéristiques d'objets authentiques. Ils peuvent ensuite insérer une porte dérobée aux fonctionnalités de l'objet cloné. L'objet cloné fonctionne normalement comme les objets authentiques mais peut en plus envoyer clandestinement les informations collectées sur l'utilisateur au fabricant ou à un gouvernement à des fins d'espionnage. Il est également possible lors de l'installation, que l'attaquant puisse se faire passer pour l'installateur légitime. Il peut dans ce cas de figure remplacer l'objet authentique par un objet compromis possédant exactement les mêmes caractéristiques que l'objet authentique. Cet objet compromis peut falsifier les informations de l'utilisateur, il peut également retransmettre à l'attaquant toutes les informations sensibles de l'utilisateur.

(ii) **Attaques visant les communications** : Ces attaques peuvent être menées dès que

l'attaquant a une connectivité avec un ou plusieurs objets.

— **Exécution de code malveillant** : Les objets de la couche de détection sont exposés à la menace d'exécution de codes malveillants en leur sein. Le code malveillant peut être un programme malveillant (malware), un virus ou cheval de Troie. Cette attaque peut avoir lieu dès qu'un attaquant a un accès non autorisé aux objets concernés. Généralement l'objectif de ces attaques est le contrôle à distance des objets. Par exemple, Mirai a été utilisé en 2016 pour contrôler plusieurs milliers d'objets IoT afin de lancer une attaque de déni de service distribué (DDoS).

— **Déni de service** : La faible quantité de mémoire et d'énergie dont disposent les objets rend ces derniers vulnérables aux attaques de déni de service. Ces attaques peuvent compromettre la disponibilité des objets. En effet, un attaquant peu rendre les ressources d'un objet inutilisables pour ses utilisateurs légitimes. Pour ce faire, il peut continuellement envoyer un nombre important de messages à l'objet jusqu'à l'épuisement des ressources de l'objet (drainage de la batterie, débordement de la mémoire, etc.).

— **Interception des informations brutes** : La transmission des données au niveau de la couche détection est exposée à l'attaque de l'interception des données transmises. Les données collectées et transmises sont généralement des données sensibles sur la vie privée des utilisateurs ou sur les infrastructures critiques. Un attaquant peut intercepter les données transmises et en extraire les informations.

#### 1.3.2.2 Couche réseau

La couche réseau de l'architecture IoT interconnecte plusieurs types de réseau, cela peut être problématique sur le plan de la sécurisation de la couche. En effet, certains réseaux plus sécurisés peuvent être exposés aux vulnérabilités provenant d'autres réseaux moins sécurisés. Les menaces de cette couche de l'architecture concernent généralement la vie privée, la confidentialité et l'intégrité des communications et la disponibilité du réseau. Les protocoles de communication concernés sont les protocoles d'acheminement de données (6LoWPAN), de routage (RPL) et les protocoles de transport de données de bout en bout (TCP/UDP) [GKS18, LXI17].

— **Attaque contre le routage** : Les protocoles de routage des paquets de données dans les réseaux IoT sont exposés à plusieurs attaques. Parmi ces attaques, nous avons la possibilité de modification ou d'usurpation des chemins de routage. Cette attaque, une fois réalisée, permet aux attaquants de créer des boucles de routage, de paralyser le réseau IoT ou de capturer les informations échangées.

— **Attaque de l'homme du milieu (Man-in-the-middle)** : Les réseaux IoT sont très disparates. Ainsi, plusieurs menaces pèsent sur la capture des communications entre deux ou plusieurs nœuds IoT. Il est possible pour une personne malveillante de s'interposer entre deux nœuds communiquant dans un réseau. La personne peut alors intercepter les messages échangés et peut même se faire passer pour un interlocuteur légitime. Dans le cas où aucun service de sécurité n'est mis en place, cette attaque peut être menée à tout moment. Si le service de confidentialité est mis en œuvre, l'attaque est réalisable lors de l'établissement de la connexion sécurisée où les échanges de clés de chiffrement et de session peuvent se faire sur un canal en clair. Dans l'un des deux cas, l'attaquant peut capturer les données sensibles échangées, perturber la communication en rejouant les données déjà échangées, etc. Par ricochet, l'attaque de l'homme du milieu concerne également la vie privée.

— **Ecoute clandestine et menaces sur la transmission des données** : Il est possible que la communication entre les objets et le reste de l'architecture se fasse sur un canal non sécurisé. Dans ce cas de figure, la communication est particulièrement vulnérable à l'attaque d'écoute clandestine (eavesdropping), car cette attaque est particulièrement facile à réaliser dans ces conditions. Les attaquants peuvent également intercepter les messages et exploiter les données que ces messages transportent sans difficulté. Dans le cas où la communication se fait sur un canal sécurisé et que les échanges de configuration de la communication sécurisée se font sur un canal non sécurisé, un attaquant peut capturer les échanges et retrouver les clés secrètes et les clés de session. Il pourra, par la suite, utiliser ces clés pour déchiffrer les informations échangées sur le canal sécurisé établi.

#### 1.3.2.3 Couche service

La couche service fournit à l'architecture IoT plusieurs services et activités comme les API, le traitement des événements, les analyses de données, etc. Les services de cette couche sont exposés à plusieurs attaques. Ces attaques concernent l'exploitation des vulnérabilités des mécanismes d'authentification, le déni de service, la confidentialité des informations, l'intégrité des données, la non répudiation et la disponibilité des services [LXI17]. Ce sont entre autres :

— **Accès aux informations privées** : Dans la couche service, les informations sont pour la plupart extraites, traitées et stockées. Si les services de confidentialité ne sont pas correctement assurés, il est possible pour un attaquant d'accéder à des informations sensibles (position GPS, rythme cardiaque, etc.) et de les exposer.

— **Accès non autorisé au service** : Lorsqu'un mécanisme d'authentification faible est mis en place (par exemple un service utilisant un simple login/mot de passe dont les mots de passes sont faibles), il est possible pour un utilisateur non autorisé d'accéder à des services réservés aux utilisateurs légitimes.

— **Déni de service** : L'attaque de déni de service existe également à ce niveau de l'architecture. Elle peut exploiter les bugs, les erreurs d'implémentation, etc. L'exploitation de ces failles peut permettre à un attaquant de rendre le service indisponible et inaccessible, par la même occasion de perturber ou de mettre l'infrastructure IoT hors service. La réalisation d'une telle attaque est possible lorsque l'attaquant compromet le système hébergeant le et les services ciblés.

#### 1.3.2.4 Couche application

La couche application héberge les applications. Au niveau de cette couche de l'architecture, plusieurs menaces et vulnérabilités sont également présentes. Les protocoles de communication concernés au niveau de cette couche sont les protocoles CoAP, HTTP, MQTT, etc. Il faut également noter que c'est à partir de cette couche que les utilisateurs interagissent généralement avec les applications IoT. En règle générale, l'utilisateur en lui-même, à cause de ses négligences et de sa non sensibilisation, est une porte d'entrée parfaite pour un attaquant, au système, donc constitue une vulnérabilité. Ce sont entre autres :

— **Authentification** : La plupart des utilisateurs utilisent le couple nom d'utilisateur/mot de passe pour se connecter aux applications. L'utilisateur d'un mot de passe exposent les applications aux attaques par force brute si l'attaquant possède une puissance de calcul assez élevée. De plus, plusieurs techniques sont utilisées par les attaquants pour trouver le mot de passe des utilisateurs, notamment avec l'utilisation d'enregistreur de frappes.

— **Vie privée** : Lorsque les données sont exploitées en clair par les applications, elles sont exposées à plusieurs attaques. En effet, un attaquant peut traquer une personne à son insu, vendre ou divulguer ses informations personnelles.

— **Commandes et contrôles** : Les attaques qui peuvent compromettre une application exploitent généralement des failles des mécanismes d'authentification. Lorsqu'elles sont exploitées, un attaquant peut avoir un accès non autorisé à l'application afin d'envoyer des commandes aux objets, par exemple : déverrouiller la porte de la maison, mettre tous les feux au vert en cas d'attaques terroristes, suspendre le système de refroidissement du bâtiment, piloter la voiture sans que son conducteur ne le sache, etc. Par exemple, en 2015, deux chercheurs ont réussi à prendre le contrôle à distance d'une jeep cherokee. Ils avaient alors désactivé le système de freinage, bloqué l'accélérateur, etc.

— **Falsification des données** : Les attaques pouvant compromettre l'intégrité des données existent au niveau de la couche application. En ce sens, il est possible pour un attaquant qui a un accès aux données des applications, de modifier ces données pour de mauvaises intentions.

Au vu de tout ce qui précède, nous remarquons que des failles de sécurité sont présentes dans toutes les couches d'une architecture IoT. Cela est dû aux caractéristiques spécifiques et hétérogènes des objets, à l'attractivité de ces derniers par les pirates, aux protocoles et technologies de communication, aux services et applications composant les différentes couches de l'architecture. Par conséquent, les risques pour la sécurité et la protection de la vie privée sont très élevés. Il est alors très important de prendre en compte les exigences de sécurité et de protection de la vie privée des personnes depuis la conception des systèmes jusqu'au déploiement de ces systèmes. Il est également important de sensibiliser les utilisateurs concernant les enjeux de sécurité et de protection de la vie privée dans leur utilisation de l'IoT.

### 1.3.3 Services de sécurité, protection de la vie privée et confiance dans l'IoT

La mise en place de systèmes et d'applications IoT nécessite de prendre en charge les exigences de sécurité et de protection de la vie privée dès la conception. Dans les sous-sections suivantes, nous définirons et décrirons les services exigés pour pouvoir assurer la sécurité dans l'IoT, il s'agit de : l'authentification, le contrôle d'accès, la confidentialité, la disponibilité et la non répudiation. Nous définirons et décrirons également la protection de la vie privée et la gestion de la confiance, complémentaires à la sécurité.

#### 1.3.3.1 Protection de la vie privée

La vie privée concerne les informations qui représentent l'identité d'une personne, sa santé, ses activités journalières (finances, déplacements, etc.) ou toute autre information personnelle. La protection de la vie privée consiste pour une personne à avoir le contrôle sur ces informations collectées dans le cadre d'une application spécifique, par exemple une application de fitness. Elle consiste également à garantir pour un utilisateur que seules les entités autorisées, c'est-à-dire ayant les autorisations nécessaires pourront avoir accès à ses données personnelles.

La vie privée dans l'IoT peut être menacée à tous les niveaux de l'architecture IoT. Au niveau de la couche détection, les données sensibles des utilisateurs qui sont collectées par les capteurs doivent être protégées des " regards indiscrets ". Dans la couche réseau, la vie privée peut également être attaquée. Par exemple, un attaquant pourrait intercepter les communications entre un capteur ECG (Electrocardiogramme) et le serveur applicatif du système e-santé, pour ainsi recevoir les informations sensibles de l'utilisateur. La vie privée peut également faire l'objet d'attaque au niveau de la couche application, où les applications accèdent aux données stockées dans le *Cloud*. A ce niveau, les données l'externalisation des données avec des applications tierces parties posent un risque pour la divulgation de la vie privée.

#### 1.3.3.2 Gestion de la confiance

La confiance permet de s'assurer que les entités IoT évoluent dans un environnement sûr et que ses interlocuteurs sont les interlocuteurs légitimes. Les tâches nécessaires à la

gestion de la confiance sont l'établissement de la confiance, le maintien de la confiance et la révocation de la confiance. L'établissement et la gestion de la confiance sont des tâches essentielles et complémentaires à la sécurité. Le contexte hétérogène et dynamique des entités de l'IoT requiert la définition de différentes techniques d'évaluation de la confiance [VKK+18]. La spécificité des objets connectés fait que la gestion de la confiance doit se faire de manière dynamique. Cela est assez complexe car les capacités énergétiques des objets et les puissances de calcul des objets sont limitées. Généralement, les mécanismes basés sur les réputations sont utilisés pour gérer la confiance dans l'IoT.

La mise en œuvre de mécanismes de protection de la vie privée et de gestion de la confiance sont insuffisants pour pallier les problèmes de sécurité et de protection de la vie privée. Elles sont complémentaires aux services de sécurité. Dans les sous-sections suivantes, nous allons définir les services de sécurité.

#### 1.3.3.3 Services de sécurité

##### 1.3.3.3.1 Identification, Authentification et Contrôle d'accès

**1.3.3.3.1.1 Identification et Authentification** L'authentification est un service qui consiste à s'assurer de l'identité d'une entité (utilisateur, objet, application) afin d'autoriser cette entité à accéder à des ressources protégées. Il suit l'identification qui consiste à identifier une entité en utilisant un élément d'identification. Un élément d'identification est une information qui permet de formellement identifier de manière unique une entité.

On distingue plusieurs méthodes d'authentification selon le niveau de sécurité demandé, et cela moyennant l'utilisation de différents éléments d'identification et d'informations appelés facteurs d'authentification. Il existe cinq facteurs d'authentification : ce que l'entité connaît, ce que l'entité détient, ce que l'entité est, ce que l'entité sait faire ou fait et où l'entité se situe. Les facteurs que l'entité connaît sont les mots de passe, les codes PIN, etc. Les facteurs que l'entité détient sont les cartes à puce, les cartes magnétiques, smart band (bracelet connecté), smartphones, etc. Les facteurs qui déterminent l'entité sont entre autres l'empreinte digitale, l'empreinte rétinienne, les caractéristiques du visage, etc. Les facteurs que l'entité sait faire ou fait sont une signature manuscrite,

une biométrie du comportement, etc. Le dernier type de facteur détermine où l'entité se situe lors de son authentification, par exemple une localisation géographique. Un autre élément important de l'authentification est la préservation de la vie privée. Les mécanismes d'authentification doivent permettre la préservation de la vie privée lors du processus d'authentification en protégeant les identités des entités qui s'authentifient.

Dans l'IoT, l'authentification concerne les entités suivantes : les utilisateurs, les objets et les messages échangés. Les utilisateurs doivent s'authentifier auprès des applications IoT afin d'accéder aux données présentées par celles-ci ou commander des objets connectés comme les actionneurs. Le second niveau d'authentification est celui des objets dans un système IoT. À ce niveau, le but de l'authentification est de s'assurer que le dialogue s'effectue entre les objets connus. Le troisième niveau est l'authentification des messages échangés entre les objets (M2M), ou entre les objets et l'application.

**1.3.3.3.1.2 Contrôle d'accès** Le contrôle d'accès est un mécanisme qui permet de s'assurer que seules les entités possédant les autorisations nécessaires accèdent à une ressource protégée. En d'autres termes, c'est un mécanisme qui permet de protéger l'accès à une ressource/donnée/information sensible en ne permettant qu'aux entités ayant les droits nécessaires d'y accéder. Par exemple, permettre à un médecin d'accéder aux données d'un patient dans le cadre d'un service e-santé. Ce contrôle d'accès peut contribuer, entre autres, à la protection de la vie privée des utilisateurs. En outre il limite les actions qu'une entité peut entreprendre dans un système, en lui attribuant seulement les autorisations nécessaires. Ce service est assuré grâce à un serveur qui reçoit les requêtes d'accès afin d'obtenir l'autorisation ou le refus de l'accès.

Dans l'IoT, avoir l'autorisation pour une entité signifie la possibilité pour cette entité d'accéder aux informations d'un capteur, de commander un actionneur, etc. [KIRM18]. Pour ce faire, il existe plusieurs techniques de contrôle d'accès : contrôle d'accès basé sur une sorte de liste de contrôle d'accès ou ACL (Access Control List), contrôle d'accès basé sur les rôles, les capacités, les attributs, etc.

**1.3.3.3.2 Confidentialité** La confidentialité des données doit être assurée depuis la collecte, durant la transmission et jusqu'au stockage. La nature des informations échangées dans l'IoT rend la confidentialité des données lors des échanges primordiale. Elle assure que seules les entités impliquées dans une communication ont accès aux données échangées. Le chiffrement est le mécanisme principal utilisé pour la mise en œuvre de

ce service de sécurité. Les données sont chiffrées à l'aide d'un algorithme cryptographique. Une clé est utilisée pour le chiffrement et une autre clé pour le déchiffrement. Ainsi, même si les données sont capturées durant la communication, elles restent inintelligibles pour les entités non autorisées, c'est-à-dire ne possédant pas les clés.

Selon le principe de Kerckhoffs [Ker83], la force d'un système de chiffrement réside dans le secret des clés. On distingue deux types de système cryptographique : les systèmes à cryptographie symétrique et les systèmes à cryptographie asymétrique. Un système à cryptographie symétrique utilise la même clé pour le chiffrement et le déchiffrement des données. Un système à cryptographie asymétrique utilise des clés différentes pour le chiffrement et le déchiffrement des données. Pour ce faire, chaque entité possède une paire de clés : une clé dite publique et une dite clé privée.

**1.3.3.3.3 Intégrité** L'intégrité des données permet de s'assurer que les données n'ont subi aucune altération volontaire ou accidentelle durant leur transmission, leur traitement ou leur stockage. Elle est primordiale pour la sécurité des transmissions car elle permet de s'assurer que les informations reçues par une ou plusieurs destinations sont bien celles qui ont été transmises par la source. Le mécanisme utilisé pour contrôler l'intégrité des données est le hachage. Le hachage consiste à générer une empreinte unique à partir des données à transmettre. Cette empreinte est généralement transmise en même temps que ces données. À la réception une nouvelle empreinte est générée en appliquant le même algorithme. Cette empreinte sera comparée à celle générée avant l'envoi. Si les deux empreintes sont identiques, alors nous avons une preuve que les données n'ont pas été altérées.

**1.3.3.3.4 Authentification de l'origine** L'authentification de l'origine d'un message permet d'assurer que lors d'une communication, les données reçues proviennent effectivement de l'un des interlocuteurs légitimes. Le *Message Authentication Code* (MAC) et la signature électronique sont les mécanismes les plus utilisés pour à la fois authentifier l'origine d'un message et vérifier son intégrité. Le MAC utilise une fonction de hachage qui génère un bloc sur la base du résultat de la concaténation des données échangées et d'une clé symétrique. Ce bloc servira à authentifier l'origine des données en même temps que de vérifier leur intégrité. La signature électronique utilise également une fonction de hachage pour générer le hachage des données échangées avant d'utiliser la crypto-

graphie asymétrique pour chiffrer le résultat du hachage. Ceci permettra d'authentifier l'origine des données ainsi que leur intégrité.

**1.3.3.3.5 Non répudiation** La non répudiation est un des principes de la sécurité informatique qui consiste à assurer qu'un expéditeur ne peut nier avoir envoyé un message et qu'un destinataire ne peut nier avoir reçu un message. Généralement, la signature électronique est le moyen utilisé pour assurer la non répudiation.

**1.3.3.3.6 Disponibilité** La disponibilité est le service de la sécurité qui consiste à assurer que les services et les ressources offerts par le système sont disponibles à tout moment, et accessibles par les utilisateurs authentifiés et autorisés. C'est un service essentiel au bon fonctionnement d'une application IoT. Les mécanismes utilisés pour la mise en œuvre de la disponibilité sont les moyens de protection contre les attaques de déni de service (DoS : *Denial of Service*) et les attaques de déni de service distribué (DDoS : *Distributed Denial of Service*).

## 1.4 Conclusion

Dans ce chapitre, nous avons commencé par introduire le concept d'Internet des Objets et nous avons présenté quelques architectures de référence utilisées dans l'IoT. L'IoT a fait l'objet de plusieurs définitions et plusieurs travaux de normalisation ont été entrepris. Nous avons également présenté les principales architectures de référence proposées pour l'IoT. Cette multitude de définitions, de travaux de normalisation et d'architectures de référence est due aux différentes visions que les parties prenantes ont de l'IoT. Ainsi, nous avons proposé une définition de l'IoT et nous avons détaillé en particulier l'architecture de référence à quatre couches de l'UIT, que nous avons retenu dans le cadre de nos travaux de recherche. Par la suite, nous avons décrit les caractéristiques de l'IoT. Ensuite, nous avons identifié les problèmes de sécurité et de protection de la vie privée dans l'IoT. Enfin, nous avons présenté les services de sécurité et de protection de la vie privée permettant d'assurer la sécurité et la protection de la vie privée dans l'IoT.

L'IoT recouvrant de nombreux domaines, nous avons choisi de nous intéresser en particulier à la ville intelligente. En effet, les applications IoT que l'on retrouve dans une ville intelligente sont les applications qui ont un plus grand impact sur la vie privée des

citoyens (*smart home*, e-santé, *smart building*, *smart grid*, *smart mobility*, *smart water management*, etc.). De plus, c'est un domaine en forte expansion, avec une hausse de 80% de l'urbanisation mondiale prévue pour 2050. C'est pourquoi, dans ce contexte, il nous semble également opportun de nous intéresser à la sécurité et à la protection de la vie privée centrées sur l'utilisateur. Ainsi, dans le chapitre suivant, nous nous intéresserons à l'approche de la sécurité et de la protection de la vie privée centrées sur l'utilisateur dans la ville intelligente.

# Chapitre 2

# Sécurité Sensible au contexte dans l'Internet des Objets

## 2.1 Introduction

Dans le premier chapitre, nous avons présenté l'Internet des Objets. Nous avons également identifié les principaux problèmes de sécurité et de protection de la vie privée dans l'IoT. Nous avons, par la suite, défini les services à mettre en œuvre afin d'assurer la sécurité et la protection de la vie privée des utilisateurs dans les applications IoT. Plusieurs travaux de recherche ont été effectués afin d'assurer la sécurité et la protection de la vie privée dans l'IoT. Cependant, la grande majorité des solutions proposées aujourd'hui ne fournit pas de mécanismes permettant aux utilisateurs de contrôler la sécurité et la protection de leur vie privée. Cela est certainement dû à la conception des systèmes IoT, basés sur un modèle dans lequel l'utilisateur doit implicitement faire confiance aux fournisseurs de services. Par conséquent, au vu des problèmes identifiés, ce modèle est insuffisant d'autant plus que les utilisateurs sont de plus en plus soucieux de la sécurité de leurs dispositifs et de la protection de leur vie privée.

Par ailleurs, l'IoT recouvre de nombreux domaines. Dans ce chapitre, nous nous intéresserons à la ville intelligente, car les applications IoT d'une ville intelligente ont un impact significatif sur la vie privée des citoyens. De plus, ce domaine est en forte expansion, avec approximativement 68% de la population mondiale qui vivra dans les villes en 2050 [1]. Compte tenu de l'importance des problèmes de sécurité identifiés, nous pensons que l'approche de sécurité et de protection de la vie privée centrées sur l'utilisateur permettra d'apporter des solutions idoines à ces problèmes.

Dans ce chapitre, nous commencerons par détailler le concept de sécurité et de protection de la vie privée centrées sur l'utilisateur dans les applications IoT des villes

1. https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html

intelligentes. Nous présenterons également les limites des solutions classiques (c'est-à-dire non centrées sur l'utilisateur) et les apports de la sécurité et de la protection de la vie privée centrées sur l'utilisateur dans ce domaine applicatif. Ensuite, nous définirons les concepts de sensibilité au contexte et sécurité sensible au contexte dans l'IoT (cf. section 2.3.2). Cette sécurité sensible au contexte nous permettra plus tard, non seulement de mettre en place une sécurité et une protection de la vie privée centrées sur l'utilisateur, mais aussi d'avoir une sécurité optimale et auto-adaptative aux différentes situations et contextes. Dans un deuxième temps, nous présenterons des travaux qui se sont intéressés à la sécurité sensible au contexte dans certaines applications de l'IoT. Nous effectuerons une revue critique de ces travaux en termes de protection de la vie privée et de types de services de sécurité sensibles au contexte proposés, puis nous dégagerons leurs apports et limites pour une sécurité et une protection de la vie privée centrées sur l'utilisateur dans les applications de la smart city.

## 2.2 Sécurité et protection de la vie privée centrées sur l'utilisateur dans la smart city

### 2.2.1 Importance de l'utilisateur

Les applications IoT ne concernent pas seulement l'amélioration des processus commerciaux et industriels, elles ont aussi un impact significatif sur la vie quotidienne des personnes. Les applications IoT des villes intelligentes comprennent les applications de maisons intelligentes (smart home), de e-santé (e-health), de réseaux électriques intelligents (smart grid), de gestion de l'eau (smart water management), de gestion des déchets intelligents (smart waste management), de gestion des transports intelligents (intelligent transport system), etc. Ces applications représentent la grande majorité des applications IoT qui ont un impact significatif sur la vie quotidienne de leurs utilisateurs [LL15]. Par conséquent, elles devraient attirer un grand nombre d'utilisateurs.

Cependant, plusieurs préoccupations relatives à la protection de la vie privée des utilisateurs et à la sécurité de leurs biens et de leurs objets figurent parmi les obstacles à l'adoption de ces applications par les utilisateurs. En effet, force est de constater que les réseaux et les dispositifs IoT sont exposés à plusieurs menaces de sécurité (cf. chap. 1, section 1.3.2). La vie privée des utilisateurs est également largement exposée à des risques de divulgation, d'espionnage, de suivi en continu, etc. Dans ce contexte, les

utilisateurs risquent de ne plus avoir de vie privée. Ces risques sont relatifs aux objets connectés, aux données qu'ils collectent sans l'aval des utilisateurs et à l'absence de mécanismes de protection appropriés.

Plusieurs travaux ont été effectués afin de répondre à ces problèmes. Plusieurs approches de sécurité et de protection de la vie privée ont ainsi été proposées. La grande majorité de ces approches se focalise sur un service de sécurité en particulier, ou sur la protection de la vie privée indépendamment des services de sécurité, sans la prise en charge des caractéristiques spécifiques des utilisateurs (mobilité, fracture numérique et facilité d'utilisation, préférences, habitudes d'utilisation, etc.) et de l'IoT (cf. chap. 1, section 1.2.8). Or, la mise en œuvre des services de sécurité et la protection de la vie privée sont complémentaires.

Supposons que Bob est une personne diabétique et propriétaire d'une maison intelligente. La maison de Bob contient divers appareils d'e-santé comme un glucomètre, un tensiomètre, un capteur ECG, un injecteur d'insuline. Dans sa maison, il a également des appareils électroménagers tels qu'un réfrigérateur, un compteur d'eau, un compteur d'électricité, un téléviseur connecté à Internet, etc. Bob porte également un bracelet de fitness qui relève en continu ses activités journalières et effectue le suivi de son sommeil. Ses dispositifs génèrent des données, en grande quantité, qui peuvent être exploitées par diverses entités à diverses fins. Par exemple, le système de suivi d'e-santé de l'hôpital traite les informations provenant des appareils d'e-santé et du réfrigérateur pour améliorer l'état de santé de Bob et son alimentation. Les fournisseurs de contenus pour TV collectent les données de Bob afin de connaitre ses préférences et lui proposer des contenus personnalisés. Le réseau smartgrid collecte les données de son compteur électrique afin d'adapter la distribution de courant électrique. Le réseau de fourniture d'eau collecte les données de son compteur d'eau afin d'adapter la distribution de l'eau et de mieux maitriser la gestion de l'eau.

L'usage inapproprié des données susmentionnées pourrait porter atteinte à la vie privée de Bob. En effet, même si Bob souhaite avoir un très bon suivi médical et une alimentation saine afin de mieux vivre avec son diabète, il se peut qu'il ne souhaite pas que l'on sache qu'il est diabétique. De ce fait, la divulgation intentionnelle ou accidentelle des données sur son état de santé ou sur son alimentation par les entités concernées peut définitivement affecter sa vie sociale et professionnelle. De plus, il se peut que Bob ne souhaite pas que l'on divulgue son style de vie, ses trajets, les lieux qu'il fréquente,

ses habitudes (sport, programmes TV, sommeil, etc.), ses appareils électriques et les utilisations qu'il fait de l'eau. Si ces informations étaient divulguées, Bob pourrait être la cible de cambrioleur ou être espionné jour et nuit dans le cas il prendrait part à une organisation politique, etc. Bob porte également sur lui certains de ses dispositifs d'e-santé lors de ses déplacements en ville. Ainsi, lors de ses déplacements, des mécanismes devront également être mis en œuvre pour préserver sa vie privée et la sécurité de ses applications. Dans la suite de ce chapitre, nous nous baserons sur le cas de Bob.

Dans la suite de cette section, nous présenterons la sécurité et la protection de la vie privée centrées sur l'utilisateur, dans laquelle les utilisateurs des systèmes IoT peuvent avoir la capacité de jouer un rôle central et efficace dans la protection de leur vie privée.

L'approche sécurité et vie privée centrées sur l'utilisateur est une approche qui semble bien adaptée pour assurer la sécurité et la protection de la vie privée des utilisateurs dans diverses applications, aussi bien que dans l'IoT comme en témoigne la définition donnée par la norme ISO/IEC 29100. La norme ISO/IEC 29100 version 2011 amendée en 2018 définit en effet un cadre pour la protection de la vie privée dans les technologies de l'information et de la communication. Selon cette norme, la sécurité et la protection de la vie privée sont centrées sur l'utilisateur si les parties consommatrices des données de l'utilisateur (applications, services, etc.) utilisent les préférences de protection de la vie privée définies par l'utilisateur pour exercer des contrôles à toutes les étapes du traitement de l'information, de la collecte et du stockage à l'utilisation, au transfert et à la suppression [Sve19]. En outre, les lois sur la protection de la vie privée et des données des utilisateurs mettent en avant une plus grande implication des utilisateurs dans la protection de leurs données en leur permettant d'exercer plus de contrôle sur la collecte de leurs données [Cou16, BPGB18]. Ainsi, l'approche de la sécurité et de la protection de la vie privée centrées sur l'utilisateur permet à l'utilisateur d'avoir un véritable rôle à jouer dans la protection de sa vie privée (Barhamgi et al., 2018).

Afin de pouvoir apporter une solution appropriée à la problématique de l'approche centrée sur l'utilisateur, il est important de situer les différentes interactions d'un utilisateur avec les applications IoT, c'est-à-dire, expliciter quand un utilisateur est actif et quand il ne l'est pas. Il existe différents moyens à travers lesquels un utilisateur peut interagir avec une application IoT. Ces moyens d'interaction sont : le contrôle à distance, la surveillance active, les notifications automatiques et l'automatisation des actions. Le choix du moyen d'interaction dépend en général des cas d'utilisation de l'application.

Le contrôle à distance met une interface graphique à la disposition de l'utilisateur afin qu'il puisse contrôler le système à distance. Cette interface graphique peut être une application web ou une application mobile accessible à travers les appareils tels qu'un smartphone, une tablette ou un ordinateur. Par exemple, un utilisateur peut ajuster le niveau de la température de l'air conditionné ou permettre occasionnellement à un livreur d'accéder à sa maison en son absence.

La surveillance active, ou monitoring en anglais, permet à l'utilisateur d'être informé par l'application IoT en temps réel. Ce mode d'interaction utilise également les applications web ou mobiles accessibles à travers les appareils susmentionnés. Par exemple, un utilisateur peut suivre en continu l'évolution de son rythme cardiaque, ou suivre en temps réel sa voiture qu'il a prêtée à un ami. Les notifications automatiques permettent à l'utilisateur de recevoir des notifications ou d'être alerté quand les évènements inhabituels se produisent. Ces notifications peuvent être envoyées sous formes de simples notifications sur les appareils de l'utilisateur, de messages SMS, de courriels électroniques ou voire même d'appels téléphoniques dans le cas d'une urgence.

Par exemple, l'utilisateur voudrait être informé lorsque la température interne du réfrigérateur excède un seuil, ou lorsque sa tension artérielle est anormalement basse. L'automatisation dans l'IoT est le mode d'interaction dans lequel les objets et les dispositifs effectuent des actions en fonction des différentes situations de l'utilisateur sans forcément son intervention. Ce mode d'interaction est basé sur certaines caractéristiques de l'IoT telles que la sensibilité au contexte et l'intelligence. Par exemple, en fonction de la température du thermostat, le climatiseur se met automatiquement en route, ou tous les matins l'arrosage de la pelouse se déclenche automatique, la machine à café se met en marche dès que la montre de l'utilisateur signale qu'il est réveillé, etc.

### 2.2.2 Limites des méthodes de sécurité classiques dans les applications de la smart city

Depuis l'avènement de l'IoT, beaucoup de travaux ont été effectués afin de résoudre les problèmes de sécurité des objets, des communications mais également les problèmes concernant la protection de la vie privée des utilisateurs. Les approches adoptées dans ces travaux ciblent généralement la proposition ou l'amélioration de mécanismes d'un service de sécurité particulier ou une combinaison de quelques services, mais pas l'ensemble des services [ASEJY18, AAC+18, BPGB18, Bor18, DK17, GMS14, GMSS23,

JQM+18, MK18, MAPP12, Nas17, SDK18, TKSA16, WM17].

Les mécanismes sont ainsi proposés pour répondre à un ou plusieurs modèles de menace de sécurité auxquels le système est exposé. Un modèle de menace de sécurité, ou threat model en anglais, est un ensemble de menaces potentielles auxquelles un système est exposé dans une situation donnée. Il permet de répondre aux questions suivantes par rapport à un service de sécurité ou à une situation donnée : quelle est la surface d'attaque ? de quoi/de qui le service doit-il être protégé ? quelles sont les menaces à prendre en compte ? quels sont les risques encourus en cas d'exploitation de ces menaces et quelles sont les mesures de sécurité à mettre en œuvre ?

Par ailleurs, ces mécanismes ne considèrent pas certaines caractéristiques des utilisateurs des applications susmentionnées. En outre, les mécanismes de protection de la vie privée existants ne sont pas conformes aux exigences des récentes lois sur la protection de la vie privée [Cou16]. Parmi ces caractéristiques, nous avons la mobilité, les préférences vis-à-vis de la vie privée, la fracture numérique et les habitudes d'utilisation. La mobilité des utilisateurs entraîne un changement fréquent de la situation de l'utilisateur (réseau d'accès, position géographique, dispositifs portés, etc.).

Chaque situation d'un utilisateur présente son modèle de menace. Par exemple, lorsque Bob a une crise chez lui et qu'il tombe accidentellement, les secours sont automatiquement contactés et envoyés à son adresse. Les services médicaux doivent dans l'urgence accéder à l'adresse de la maison et aux données de Bob sans son avis préalable car Bob est inconscient. Dans cette situation, les données sont exposées à un risque de divulgation (de la part des services médicaux d'urgence). En effet, les agents sur place peuvent par inattention divulguer l'état de santé de Bob, par exemple, en discutant avec les voisins ou des journalistes si Bob est une personne importante. Une autre situation est lorsque Bob est dans un restaurant et que ces dispositifs utilisent une connexion Internet non sécurisée. Dans cette situation, Bob est exposé à plusieurs attaques pouvant viser sa vie privée, la sécurité de ces dispositifs, etc. (cf. section 1.3.2). Ces dispositifs peuvent être attaqués, ses informations sensibles interceptées. Les risques alors changent continuellement avec la mobilité de l'utilisateur. Très peu de mécanismes proposés considèrent également les préférences vis-à-vis de la vie privée des utilisateurs. Or, chaque utilisateur a ses préférences par rapport au niveau de sécurité qu'il souhaiterait avoir. Ainsi, certains utilisateurs préfèrent un niveau sécurité élevé au détriment de la simplicité alors que d'autres y sont peu sensibles.

La fracture numérique est le fait qu'un grand nombre d'utilisateurs de services numériques et informatiques maîtrisent peu ou pas du tout les applications et outils offerts par ces services, l'IoT y compris. Le manque de simplicité dans la mise en œuvre des mécanismes proposés dans les approches classiques de sécurité pousse un certain nombre d'utilisateurs à se passer de la sécurité. Une autre résultante de la fracture numérique est la non sensibilisation des utilisateurs aux enjeux de la protection de la vie privée. En effet, les utilisateurs ne sont généralement pas conscients des risques qu'implique la divulgation de leurs données à une tierce partie (applications, services, etc.).

En outre, l'intelligence et la sensibilité au contexte de l'IoT ne sont également pas considérées par les mécanismes de sécurité classiques. L'intelligence et la sensibilité au contexte permettront de dynamiquement et intelligemment évaluer les risques de sécurité et de vie privée, et de mettre en œuvre des mesures de sécurité et de protection de la vie privée adaptées et en temps réel. Par exemple, malgré les menaces dans certaines situations, il existe peu de risques, mais les mécanismes de sécurité classiques maintiendront le même niveau de sécurité au détriment de l'aisance de l'utilisateur. In fine, les approches classiques de sécurité et de protection de la vie privée des utilisateurs dans l'IoT montrent leurs limites en termes d'efficacité, de capacité d'adaptation et de simplicité d'utilisation.

### 2.2.3 Sécurité et protection de la vie privée centrées sur l'utilisateur dans les applications de la smart city : cas du smart home et de la e-santé

Les dispositifs des applications de villes intelligentes susmentionnées collectent systématiquement et automatiquement une grande quantité de données sensibles des utilisateurs (section 1.3.1). Les données collectées sont envoyées vers la couche service pour être traitées. Les applications exposent alors les résultats des traitements et donnent parfois la possibilité aux utilisateurs d'interagir avec les différents dispositifs. Dans une application de maison intelligente, l'utilisateur peut interagir avec les dispositifs, ce qui n'est pas le cas dans la smartgrid ou la gestion intelligente de l'eau par exemple. Ainsi, en plus des menaces liées à ces dispositifs, les informations sensibles des utilisateurs peuvent être révélées lors de leur envoi, mais également au niveau du stockage. Par exemple, en 2017, Google a admis que ses dispositifs d'assistance domestique dans le cadre des maisons intelligentes, Google Home Mini, espionnaient leurs utilisateurs en

permanence, et cela en leur insu, en enregistrant systématiquement leurs conversations [Bur17]. De plus, avec les avancées actuelles, l'usage des algorithmes et des techniques d'apprentissage automatique peut permettre, en disposant de suffisamment de données, un profilage (portrait numérique) et un suivi des utilisateurs. Par conséquent, l'accès aux informations de l'utilisateur ne doit être accordé qu'aux utilisateurs, services et applications préalablement autorisés.

Dans ce contexte, les utilisateurs doivent jouer un rôle central, c'est-à-dire qu'ils doivent être habilités à gérer la divulgation de leurs données en toute sécurité avec un niveau de granularité élevé. Ainsi, dans un premier temps, l'utilisateur doit comprendre les enjeux et les risques qu'il encourt lorsqu'il divulgue une donnée à une entité quelconque. Ensuite, il doit pouvoir comparer les risques et les avantages liés à la divulgation de cette donnée afin de pouvoir prendre une décision. En effet, du fait que leurs conséquences ne sont pas directement palpables, les décisions concernant la vie privée sont difficiles à prendre. Ensuite, dans un deuxième temps, il doit avoir la possibilité de définir Qui accède à Quelle information, à Quel moment, à Quel endroit et dans Quelle condition on peut accéder à cette information.

Revenons à notre cas d'étude (cf. section 2.2.1). Bien que Bob soit très soucieux de sa vie privée et conscient des risques potentiels qu'encourt sa vie privée en cas d'éventuels partages de données avec certaines entités, il peut tout de même souhaiter échanger certaines informations de sa vie privée avec ces entités. Par exemple, il peut accepter de communiquer à la structure de gouvernance sanitaire de la ville ou son hôpital, ses relevés de taux de glucose et les informations par rapport à ses injections d'insuline pour des fins épidémiologiques, mais il souhaite garder l'anonymat parce qu'il ne veut pas que l'on sache qu'il est diabétique. Il peut également accepter d'utiliser les assistants domestiques intelligents, mais ne veut pas que ses conversations soient enregistrées en permanence. Il peut aussi souhaiter définir ses préférences de programme TV, mais ne veut pas que l'on sache les chaines TV ni les émissions qu'il regarde.

Par ailleurs, il peut aussi accepter de fournir aux services des eaux de la ville quelques relevés de sa consommation d'eau afin de permettre à la ville d'améliorer la distribution d'eau et de faire des économies. Il peut aussi accepter d'utiliser certaines applications de géolocalisation pour améliorer ses trajets, mais ne veut pas être géolocalisé en permanence et en temps réel. Dans ces exemples avant de prendre une décision, Bob devra dans un premier temps évaluer le niveau de confiance de l'entité consommatrice de ses

données. Ensuite, dans un second temps, il devra évaluer à quelles fins ses données seront exploitées et si cela lui convient en matière de protection de sa vie privée.

L'approche sécurité et protection de la vie privée centrées sur l'utilisateur comme définie plus haut est la méthode adéquate pour trouver des solutions idoines à cette problématique. Compte tenu des caractéristiques des applications IoT susmentionnées, il nous semble opportun d'utiliser la sécurité sensible au contexte comme outil dans la mise en œuvre de cette approche. En effet, les applications de la smart city peuvent être sensibles au contexte de l'utilisateur, c'est-à-dire qu'elles sont capables de réagir différemment selon les différentes situations de ce dernier. En ce sens, l'approche sécurité et protection de la vie privée centrées sur l'utilisateur permet de répondre à ces questions en considérant principalement les préférences de l'utilisateur, la réglementation en vigueur (par exemple le RGPD) et les caractéristiques spécifiques des applications IoT, en l'occurrence dans le cadre de la smart city. En plus de donner la main à l'utilisateur en matière de contrôle de ses données, elle permet, de mettre l'accent sur la facilité d'utilisation depuis la conception des objets et des applications. En effet, c'est une méthode qui permet une mise en œuvre efficace de la protection de la vie privée par conception, ou *privacy by design* en anglais [BPGB18]. En ce sens, cette approche, en plus de la simplicité d'utilisation, permet de sensibiliser l'utilisateur sur les enjeux de la vie privée, c'est-à-dire, sur le niveau de sensibilité de ses différentes données.

## 2.3 Sécurité sensible au contexte dans l'IoT

Dans cette section, nous aborderons les notions de sensibilité au contexte (*context-awareness* en anglais) et de sécurité sensible au contexte (*context-aware security* en anglais). Ensuite nous soulignerons l'importance de l'utilisation de la sécurité sensible au contexte comme outil pour une mise en œuvre efficace de l'approche de sécurité et de protection de la vie privée centrées sur l'utilisateur dans les applications de la ville intelligente. Nous prendrons en particulier comme exemple la maison intelligente et la e-santé.

### 2.3.1 Sensibilité au contexte

#### 2.3.1.1 Définitions

Définitions L'informatique sensible au contexte est principalement basée sur la notion de contexte. La plupart des chercheurs se sont basés sur la position géographique

et l'identité pour identifier le contexte, mais cela peut être insuffisant. Pour identifier un contexte avec plus de précision, il faut avoir les informations répondant aux questions suivantes : Qui, Quoi, Où, Quand et Pourquoi [AM00]. Le contexte a été défini dans le cadre de plusieurs travaux de recherche. La définition que nous retenons ici est celle donnée par Abowd et al. [ADB+99]. Ils ont défini le contexte comme suit :

"*Le contexte est toute information qui peut être utilisée pour caractériser la situation d'une entité, une entité pouvant être une personne, un lieu, ou un objet physique ou informatique* [ADB+99]."

Les données utilisées pour identifier le contexte proviennent généralement de capteurs. Elles peuvent être de n'importe quel type. Ces données brutes sont traitées pour produire de l'information contextuelle. Cette définition s'adapte alors à n'importe quelle donnée permettant de déduire un contexte. Parmi les informations contextuelles, nous pouvons citer le temps, la localisation géographique, la vitesse, le réseau d'accès utilisé, l'adresse IP, etc.

La notion d'Informatique sensible au contexte a été évoquée pour la première fois en 1994 par Schilit et Theimer [ST94]. Ils l'ont défini comme suit :

"*L'informatique sensible au contexte est la capacité des applications d'un utilisateur mobile à découvrir et à réagir aux changements de l'environnement dans lequel elles se trouvent* [ST94]."

Cette définition ne permet pas de déterminer si un système est sensible au contexte ou non. Depuis, plusieurs chercheurs ont voulu définir ce qu'était un système sensible au contexte. Ainsi Abowd et al. [ADB+99] ont donné la définition suivante :

"*Un système est sensible au contexte s'il utilise le contexte pour fournir des informations pertinentes et/ou des services à l'utilisateur, où la pertinence relève de la tâche de l'utilisateur* [ADB+99]."

Comme indiqué par Perera et al. [PZCG21], cette définition convient parfaitement pour un système sensible au contexte. À partir de cette définition, nous définissons un système IoT sensible au contexte comme suit :

"*Un système IoT sensible au contexte est un système qui utilise le contexte pour obtenir des informations pertinentes qui optimisent les services rendus à l'utilisateur grâce à des actions et des adaptations dynamiques effectuées sans intervention de cet utilisateur.* "

Un système sensible au contexte s'appuie sur un modèle de contexte [PZCG21]. Henricksen s'est basé sur les travaux de Abowd et al. pour donner une définition d'un

modèle de contexte [PZCG21]. Selon Henricksen, un modèle de contexte est défini comme suit : "*un modèle de contexte identifie un sous-ensemble concret du contexte qui est atteignable de manière réaliste à partir des capteurs, des applications et des utilisateurs et qui peut être exploité dans l'exécution de la tâche* [Hen03] ".

Un modèle de contexte est constitué d'attributs de contexte [PZCG21]. Ainsi, dans [PZCG21], les auteurs ont adopté la définition donnée par Henricksen : "*Un attribut de contexte est un élément du modèle de contexte décrivant le contexte. Un attribut de contexte a un identificateur, un type et une valeur, et éventuellement un ensemble de propriétés décrivant des caractéristiques spécifiques.*". Pour aller plus loin, selon Perera et al., un autre élément important de l'informatique sensible au contexte est la qualité du contexte (QoC : *Quality of Context*). Selon eux, "*La QoC est définie à l'aide d'un ensemble de paramètres qui exprime la qualité des exigences et des propriétés des données contextuelles* [PZCG21]". En effet, la QoC permet d'éviter les problèmes de faux positifs et négatifs dans la détermination du contexte.

Après avoir défini les différents éléments de la sensibilité au contexte, dans la sous-section suivante, nous nous intéresserons à un élément très important de la sensibilité au contexte, à savoir, le cycle de vie d'un contexte.

#### 2.3.1.2 Cycle de vie d'un contexte

Le cycle de vie d'un contexte comporte quatre phases [PZCG21] : acquisition, modélisation, raisonnement et diffusion du contexte. La figure 2.1 illustre ces phases.

La phase d'acquisition du contexte consiste à obtenir les informations contextuelles des dispositifs sources. Les informations contextuelles reçues sont stockées dans une base de données avant d'être traitées. L'acquisition de contexte peut se faire par requête/réponse, à une fréquence régulière, etc. La phase de modélisation du contexte est la phase dans laquelle les informations contextuelles reçues seront traitées en termes d'attributs, de caractéristiques et de QoC, etc. Une fois cela effectué, ces informations contextuelles sont stockées pour la prochaine phase, c'est-à-dire le raisonnement. Il existe plusieurs méthodes de modélisation du contexte : clé-valeur, basée sur l'ontologie, basée sur la logique, etc. [PZCG21].

La phase de raisonnement du contexte utilise les informations contextuelles issues de la modélisation du contexte pour déterminer le contexte approprié. C'est le processus

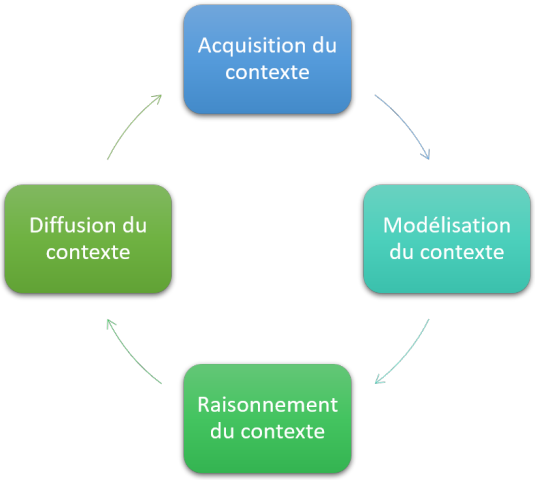



FIGURE 2.1 – Cycle de vie d'un contexte

par lequel le contexte est déduit à partir des informations contextuelles issues de la phase de modélisation. Il existe plusieurs modèles de raisonnement du contexte : les règles, les arbres de décision, naïve bayes, modèles de Markov cachés, etc. [PZCG21].

La diffusion ou la distribution du contexte, est la dernière phase du cycle de vie. Elle consiste à rendre le contexte disponible auprès des entités consommatrices.

### 2.3.2 Sécurité sensible au contexte

La sécurité sensible au contexte est récemment apparue pour pallier les nouveaux problèmes de sécurité qu'apporte l'accentuation de la présence des systèmes informatiques ubiquitaires, hétérogènes, mobiles et pervasifs. Brézillon et Mostéfaoui [BM04] ont défini la sécurité sensible au contexte comme suit :

"*la sécurité sensible au contexte consiste à explicitement considérer le contexte dans la spécification de solutions de sécurité (modèles de contrôle d'accès, protocoles cryptographiques, etc.)* [BM04]."

Nous définissons la sécurité sensible au contexte dans l'Internet des Objets comme :

"*La sécurité est sensible au contexte dans un système IoT si le choix et la mise en œuvre des mécanismes de sécurité sont basés sur le contexte de l'utilisateur et se font sans intervention explicite de l'utilisateur.*"

Selon Mostéfaoui et Brézillon [MB04], la sécurité sensible au contexte repose sur deux concepts fondamentaux : le contexte de sécurité et les politiques de sécurité. Le contexte de sécurité est défini comme étant : "*un ensemble d'informations collectées à par-*

*tir de l'environnement de l'utilisateur et de l'environnement de l'application et pertinentes pour l'infrastructure de sécurité de l'utilisateur et de l'application* [MB03]."

Dans [BM04], les auteurs ont défini le contexte de sécurité comme étant : "*une situation décrite par un ensemble d'informations qui nécessite de prendre une décision de sécurité spécifique en adaptant le protocole cryptographique utilisé dans la communication, en exigeant une méthode d'authentification plus puissante ou en refusant automatiquement l'accès à un service lorsque la détection d'intrusion est déclenchée* [BM04]." Cette définition du contexte de sécurité s'applique bien à l'IoT, soumis à un changement continu de contexte de l'utilisateur.

Prenons l'exemple d'un système IoT de suivi de la santé cardiovasculaire d'un patient [KJ17]. Les données du rythme cardiaque sont envoyées par le capteur ECG via la gateway vers le cloud. Le médecin consulte les données de l'électrocardiogramme du patient depuis l'hôpital, en s'authentifiant simplement sur l'application de suivi. Durant la nuit, l'application de suivi du patient détecte une crise de tachycardie du patient et envoie une alerte au médecin. Ce dernier essaie de s'authentifier sur l'application pour pouvoir visualiser les données. Le système a détecté un nouveau contexte, le médecin n'est pas à l'hôpital, il est 2h du matin et le médecin s'est connecté depuis un réseau mobile 4G. Les risques de sécurité émanant de la demande d'accès à l'application, par conséquent aux données du patient, sont élevés dans ce contexte. Le système doit déployer un mécanisme de vérification supplémentaire pour s'assurer qu'il s'agit bien du médecin dans ce nouveau contexte. Cet exemple illustre le fait que les risques de sécurité peuvent varier et ceci en fonction du contexte. Cela exige la mise en œuvre de services de sécurité spécifiques à chaque contexte.

Les politiques de sécurité sont essentielles à la mise en œuvre d'une sécurité sensible au contexte. Elles permettent de définir les mécanismes de sécurité à déployer en fonction d'un contexte bien déterminé. Il s'agit d'un ensemble de règles qui doivent être définies afin de prendre une décision permettant la mise en place automatique de la sécurité adéquate [ST94], [BM04].

Plusieurs travaux ont été effectués dans le but de proposer une sécurité sensible au contexte pour diverses applications. Par exemple, Mowafi et al [MAA+14] proposent une architecture de sécurité sensible au contexte pour les applications mobiles. Cette architecture utilise les informations de contexte pour adapter de manière dynamique les paramètres de sécurité des applications mobiles aux différentes situations et actions

de l'utilisateur. Elle utilise principalement le réseau comme paramètre contextuel. Elle s'incruste dans le système d'exploitation du mobile et intercepte les demandes d'accès aux ressources du mobile, par exemple la connexion au réseau. Elle évalue ensuite le contexte de l'utilisateur en considérant le niveau de sécurité du réseau auquel le mobile est connecté, c'est-à-dire réseau sécurisé ou non sécurisé. En fonction du contexte de l'utilisateur, lorsqu'une application doit être lancée, l'architecture informe l'utilisateur des risques de sécurité que pose l'exécution de cette application dans ce contexte. In fine, c'est à l'utilisateur d'autoriser ou d'annuler l'ouverture de l'application, en se basant sur les recommandations de l'architecture.

### 2.3.3 Sécurité sensible au contexte pour l'IoT

Comme nous l'avons déjà indiqué dans le chapitre 1, l'omniprésence des objets et applications de l'IoT pose plusieurs problèmes de sécurité et de protection de la vie privée. Ainsi, plusieurs travaux ont été effectués pour la résolution de ces problèmes. En revanche, la plupart des solutions proposées sont statiques, non centrées sur l'utilisateur et non adaptées au caractère dynamique de l'IoT.

Pour mieux répondre aux besoins de sécurité et de protection de la vie des utilisateurs dans les villes intelligentes, les solutions proposées doivent permettre à l'utilisateur de jouer un rôle central. En ce sens, ces solutions doivent comprendre des mécanismes qui s'adaptent au changement de contexte de l'utilisateur (par exemple en fonction de la localisation : maison, travail, lieux publics, etc.). Une réponse possible à ce défi est d'assurer la sécurité dynamiquement en fonction du contexte de l'utilisateur. Contrairement aux méthodes classiques qui sont statiques, l'utilisateur sera continuellement protégé convenablement et efficacement quel que soit son contexte. En effet, le changement de contexte ou de situation à un instant "$t$" peut rendre un niveau de sécurité insuffisant alors qu'il était tout à fait convenable et suffisant dans le contexte ou la situation précédente à l'instant "$t-1$". La figure 2.2 illustre un système IoT mettant en œuvre une sécurité sensible au contexte.

Plusieurs travaux ont prouvé qu'il est possible d'utiliser la sensibilité au contexte comme outil pour assurer la sécurité centrée sur l'utilisateur dans l'IoT [HDA+13] [NSB+15] [KW15] [dTAH18b]. Comme exemples d'une telle approche, nous citons l'authentification de l'utilisateur ou des objets connectés en fonction de la position géographique (maison, bureau, transport en commun, hôpital, etc.), l'utilisation d'un canal

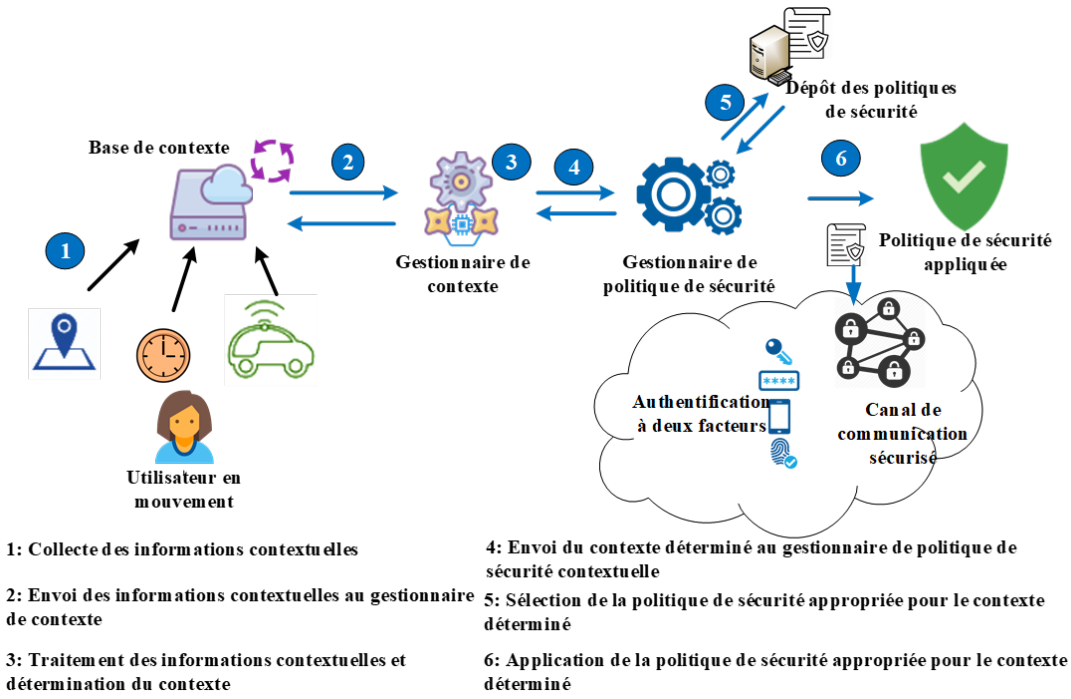



FIGURE 2.2 – Mise en œuvre de la sécurité sensible au contexte dans l'IoT

de communication sécurisé avec certains objets en fonction des activités de l'utilisateur, ou encore la non divulgation des données de l'utilisateur en présence d'une application douteuse. Par exemple, Bob peut commander et interagir avec sa maison à travers une application de pilotage à distance. Bob travaille généralement les jours ouvrables de 8h à 17h. Lorsque Bob, depuis son bureau essaye d'accéder à son application de pilotage, celle-ci lui demande de s'authentifier avec un simple code PIN. Cela est dû à la réputation sûre de son contexte à cet instant, c'est-à-dire son lieu de travail.

Quelques jours plus tard, Bob est signalé à son lieu de travail à 23h30. Il tente d'accéder à son application de pilotage, celle-ci lui demande d'utiliser son empreinte digitale afin de vérifier son identité avant de procéder à l'authentification par code PIN. Comme Bob est censé être chez lui à 23h30, le paramètre contextuel indiquant le temps a introduit un doute sur la présence effective de Bob sur son lieu de travail à une pareille heure. Dans ce contexte, même si l'environnement (lieu de travail) est réputé sûr, il est fort probable que Bob soit chez lui, et qu'une tierce personne tente un accès non autorisé à l'application de pilotage de maison intelligente de Bob. Par conséquent, la méthode d'authentification par code PIN ne suffit plus, et le système doit ajouter un facteur d'authentification supplémentaire (ici, il s'agit de son empreinte digitale) pour

vérifier l'identité de Bob. La nécessité d'avoir une sécurité qui s'adapte au changement de contexte est fortement liée aux besoins de protection des entités, données et applications de la smart city à tout moment et en tout lieu.

Revenons sur le cas du service d'authentification. Un système d'authentification classique requiert que l'utilisateur prouve son identité en fournissant un couple login/mot de passe, un code PIN ou tout autre facteur d'authentification tel que la biométrie (empreinte digitale, empreinte rétinienne, etc.). Toutefois, ces mécanismes d'authentification peuvent être compromis par l'ingénierie sociale, c'est-à-dire que l'utilisateur peut être amené à involontairement fournir ses informations d'identification, voire même par leurs vulnérabilités connues telle que la vulnérabilité à l'attaque par force brute. En outre, la plupart de ces mécanismes nécessitent l'intervention de l'utilisateur, cela n'est pas toujours adéquat dans certaines applications IoT [dTAH18b] [AKM17].

L'authentification sensible au contexte prend en compte des paramètres contextuels et permet non seulement de rendre dynamique ce service de sécurité mais aussi de le renforcer lorsque le contexte l'exige tout en limitant l'intervention de l'utilisateur. Maintenant, nous allons considérer la protection de la vie privée. Les mécanismes classiques de protection de la vie privée sont difficilement utilisables dans les applications susmentionnées à cause de ses caractéristiques dynamiques et de la faible capacité de la plupart des objets. Avec la mise en œuvre d'une sécurité sensible au contexte, des politiques de sécurité qui tiennent compte du contexte permettront aux objets de ne divulguer les informations que lorsque certaines conditions sont réunies. Lorsque c'est spécifié dans les politiques de sécurité, les données sont anonymisées et/ou chiffrées avant la transmission ou le stockage conformément à la règle précédemment définie. Lorsque l'utilisateur, l'application ou encore les objets doivent communiquer certaines données sensibles, un canal chiffré peut être automatiquement établi pour cette transmission. Cette sécurité adaptative permet de rendre la vie privée inviolable, même en cas de négligence des utilisateurs. Par exemple, lorsque Bob a une crise de diabète, dans l'urgence, certaines informations privées concernant Bob seront accessibles pour le médecin traitant et aucune autre personne.

La mise en œuvre des mécanismes de sécurité sensible au contexte sera transparente à l'utilisateur, que ce soit lors de l'authentification et du contrôle d'accès, mais aussi au moment de la transmission et du stockage des données. La transparence de la mise en œuvre dynamique de ces mécanismes en fonction du contexte rend la solution beaucoup

plus viable pour l'utilisateur et répond au dilemme "compromis entre sécurité et facilité d'utilisation". En ce sens, la sécurité sensible au contexte peut être utilisée comme outil pour la mise en œuvre efficace de la sécurité et de la protection de la vie privée centrées sur l'utilisateur dans la smart city. Le tableau 2.1 résume les avantages de la sécurité sensible au contexte.

## 2.4 Projets intégrant une sécurité sensible au contexte

La sécurité sensible au contexte dans les applications informatiques ubiquitaires et pervasives a suscité l'intérêt des chercheurs ces dernières années. Ainsi, plusieurs travaux ont été réalisés dans l'objectif de proposer des architectures et des systèmes offrant une sécurité sensible au contexte dans différents cadres, notamment celui de l'Internet des Objets. Dans les sous-sections suivantes, nous étudieront les projets proposés dans ce contexte entre 2013 et 2018.

### 2.4.1 Context-Aware Scalable Architecture (CASA)

Dans [HDA+13], les auteurs ont souligné que la plupart des systèmes d'authentification utilisent un simple mot de passe comme facteur d'authentification et qu'en fonction du contexte de l'utilisateur l'authentification basée sur un simple mot de passe peut présenter plusieurs vulnérabilités. Ainsi, ils ont proposé d'utiliser un ensemble de capteurs situés dans l'entourage d'un utilisateur pour déterminer certaines informations contextuelles comme les habitudes ou la localisation géographique de ce dernier afin d'alléger ou de durcir l'authentification. La solution proposée (CASA : Context-Aware Scalable Architecture) permet de choisir dynamiquement et en fonction du contexte de l'utilisateur la méthode d'authentification adéquate parmi celles disponibles sur leurs mobiles.

Dans un premier temps, le système utilise des capteurs présents dans l'entourage des utilisateurs (capteurs corporels, capteurs domotiques, etc.) pour déduire des facteurs passifs comme la localisation géographique ou encore le temps de la dernière connexion. Un classificateur de Bayes est par la suite utilisé pour combiner plusieurs facteurs afin de déduire le contexte de l'utilisateur (par exemple : maison, travail). Le contexte déduit est ensuite utilisé pour déterminer le type d'authentification adéquat, en l'occurrence : authentification forte avec un mot de passe et un code PIN, simple avec un code PIN seulement. Le principal paramètre contextuel utilisé dans le cadre de ces travaux est la localisation du mobile de l'utilisateur.

Tableau 2.1 – Avantages de la sécurité sensible au contexte

| Services de sécurité | Classiques | Sensible au contexte | Avantages |
|---|---|---|---|
| Authentification et contrôle d'accès | • Méthode d'authentification fixe.<br>• Attribution fixe de jetons.<br>• Autorisation unique de l'utilisateur ou pas. | • Méthode d'authentification simple ou forte en fonction du contexte et des risques.<br>• Attribution ou réattribution de jetons d'autorisation dynamique en fonction du contexte et de la réputation.<br>• Contrôle d'accès dynamique aux données basé sur les attributs ou les rôles en fonction du contexte<br>• Contrôle d'accès fixe basé sur les rôles ou sur les attributs définis. | • Compromis entre sécurité et facilité d'utilisation.<br>• Ajout et révocation dynamique d'autorisation.<br>• Protection contre les accès non autorisés. |
| Vie privée | • Divulgation statique des données.<br>• Anonymisation statique des données. | • Collecte des données sous contrôle de l'utilisateur.<br>• Divulgation des données basée sur le contexte et les risques encourus sans intervention de l'utilisateur.<br>• Anonymisation implicite des données en fonction du contexte.<br>• Intervention dynamique de l'utilisateur requise si les risques définis en fonction du contexte sont élevés. | • Meilleur contrôle sur les données de l'utilisateur.<br>• Contrôle d'accès (aux données privées) contextuel et dynamique.<br>• Les données sont divulguées ou anonymisées de façon automatique en utilisant un contrôle d'accès très granulé et contextuel.<br>• Vie privée protégée quel que soit le contexte. |
| Sécurité des communications | • Un seul mode de communication : protocole sécurisé (application, transport, réseau, etc.). | • Sélection du mode de communication en fonction du contexte de l'utilisateur : choix de la couche qui implémentera la sécurité.<br>• Choix du niveau de sécurité (algorithmes, clés, etc.). | • Les communications sensibles sont sécurisées quelle que soit la situation de l'utilisateur. |
| Sécurité des données stockées | • Données chiffrées lors du stockage dans des serveurs dédiés.<br>• Vérification statique de l'intégrité des données. | • Données chiffrées lors du stockage quel que soit l'endroit (Cloud, Fog, etc.).<br>• Vérification dynamique de l'intégrité des données. | • Des données sensibles sécurisées en fonction du lieu de stockage et du contexte de l'utilisateur. |

D'après Hayashi et al. [HDA+13], l'authentification sensible au contexte permet de pallier la négligence de l'utilisateur. Selon eux, les utilisateurs par négligence ne mettent pas de mot de passe d'authentification sur leurs téléphones mobiles. Les résultats obtenus montrent la faisabilité de la solution proposée ainsi que sa capacité à améliorer la sécurité de l'authentification des utilisateurs sans affecter la facilité d'utilisation des périphériques mobiles.

### 2.4.2 Context-Aware Security Framework for Mobiles Applications (CASFMA)

Les auteurs de [MAA+14] ont proposé une solution de sécurité sensible au contexte CASFMA (Context-Aware Security Framework for Mobiles Applications) pour répondre aux problématiques de sécurité et protection de la vie privée engendrées par la multitude d'applications mobiles utilisées dans tous les domaines tels que la maison intelligente et la e-santé et ayant accès aux informations personnelles des utilisateurs.

L'objectif de cette solution est d'obtenir des informations sur le contexte de l'utilisateur (par exemple le réseau d'accès, la localisation géographique) afin d'adapter de manière dynamique les paramètres de sécurité des applications mobiles aux différentes situations et actions de cet utilisateur. La solution consiste à placer l'exécution des applications mobiles à l'intérieur d'enclaves permettant de contrôler les communications de ces applications aux ressources réseau. Cela permet d'appliquer individuellement des mécanismes de sécurité et de contrôle de communication à chaque application. Ainsi, la tentative d'accès d'une application aux ressources réseaux déclenche une alerte d'accès au réseau qui demande l'action de l'utilisateur. Cet utilisateur aura le choix d'autoriser le lancement de l'application ou non en fonction des recommandations reçues grâce à la solution.

La figure 2.3 montre le diagramme de séquence du fonctionnement de l'architecture mise en œuvre lorsqu'une application mobile tente d'accéder au réseau. L'élément principal de l'architecture est la shadow Application (Fig. 2.4), qui s'incruste dans le système d'exploitation du mobile afin d'intercepter les demandes d'accès au réseau provenant des applications et d'appliquer les différentes mesures de sécurité. Elle est constituée de 3 composants (Fig. 2.4) : Application, AHP (Analytic Hierarchy Processor) Factor Processor et Context component. Application est le composant principal qui gère toutes les requêtes provenant des enclaves et applique les décisions de sécurité. Le Context com-

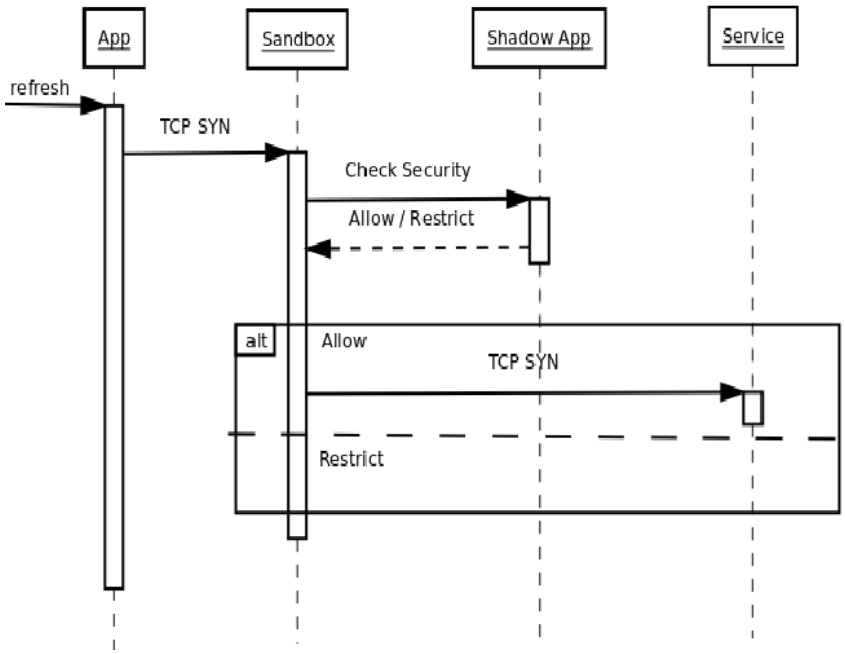



FIGURE 2.3 – Diagramme de séquences du mécanisme de sécurité de CASFMA [MAA+14]

ponent rassemble et catégorise les informations contextuelles. "AHP Factor Processor" utilise la méthode AHP afin de prendre les décisions par rapport à l'accès des applications au réseau en fonction des informations contextuelles.

La méthode AHP a été proposée par Thomas L Saaty pour traiter les problèmes de décision complexes [Ros97]. Selon Rosemnbloom [Ros97], la méthode AHP résout un problème de prise de décision complexe en le décomposant en une hiérarchie de sous problèmes. Les objectifs de la décision représentent le premier niveau et les alternatives de décisions le dernier niveau. Mowafi et al. [MAA+14] ont utilisé les paramètres contextuels suivants comme sommet de la hiérarchie : le lieu, le temps, la vitesse du mouvement et le réseau. La méthode AHP étant utilisée pour déterminer les décisions d'accès des applications mobiles de l'utilisateur au réseau, les alternatives de décision choisies par les auteurs sont : accès réseau sécurisé et accès réseau non sécurisé. Les critères, c'est-à-dire les éléments de décision, sont les sous composants des paramètres contextuels : le lieu (maison/travail, public, inconnu), le temps (matin, après midi, nuit), la vitesse de mouvement (immobile, faible, rapide) et le réseau (domicile/travail, cellulaire). L'arbre de décision est constitué de liens entre les objectifs de décision et les critères et de liens entre les critères de décisions et les alternatives de décision.

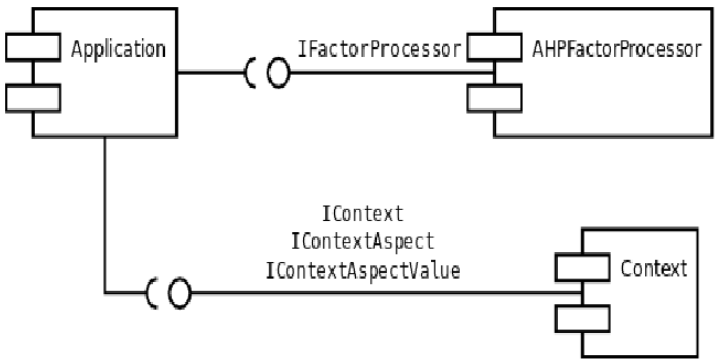



FIGURE 2.4 – Diagramme de composants de la Shadow Application [MAA+14]

Par ailleurs, des tests ont été effectués sur un prototype du framework sur un mobile Android et ont permis de confirmer l'efficacité de la proposition.

### 2.4.3 Managing Context Information for Adaptive Security in IoT environments (MCIASIoTE)

Ramos et al. ont considéré la problématique de l'utilisation des informations contextuelles dans la mise en œuvre des décisions de sécurité dans l'IoT [RBS15]. Les travaux réalisés ont permis d'étendre l'architecture de sécurité pour l'IoT proposée dans [BBHMSG14] en ajoutant deux composants : Group manager et Context manager.

L'objectif principal de cette architecture est de prouver que les différents composants proposés dans le projet Privacy Preserving Security Framework for a Social-Aware Internet of Things (PPSFSAIoT) [BBHMSG14] peuvent utiliser les informations contextuelles pour permettre aux objets de prendre des décisions par rapport à la sécurité à mettre en place. Une nouveauté de cette approche est qu'elle considère les questions de sécurité dans la gestion de l'information contextuelle. Par exemple, le gestionnaire de contexte ne traitera que les informations contextuelles provenant des objets sûrs. La figure 2.5 illustre l'architecture proposée et ses interactions [RBS15]. Elle est principalement composée des composants suivants : Identity Management, Authorization, Trust and Reputation et Group Manager. L'Identity Management est chargé de gérer les identités des objets en protégeant la vie privée. En effet, en se basant sur les politiques de protection de la vie privée, il anonymise l'identité de l'utilisateur en fonction du contexte. Le composant Authorization gère le contrôle d'accès. Il utilise le langage XACML (eXtensibe Access Control Markup Language) pour définir les autorisations sous

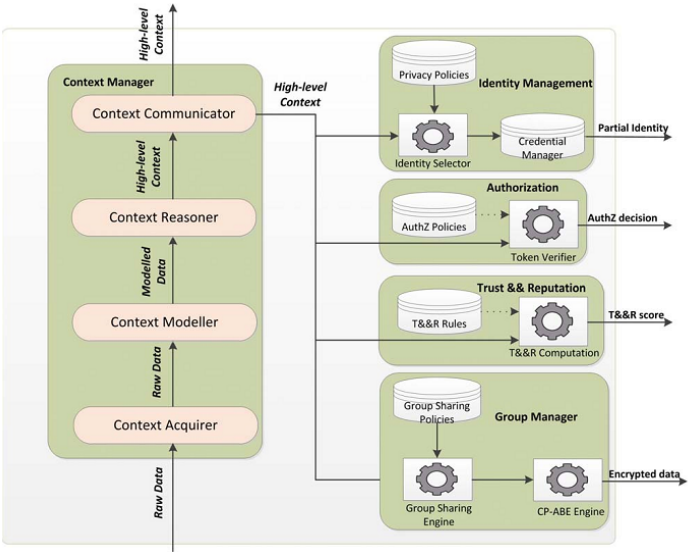



FIGURE 2.5 – Composants de l'architecture et leurs interactions [RBS15]

forme de jetons d'accès. Concrètement, chaque jeton d'autorisation délivré contient, en plus des autorisations, les informations relatives au contexte dans lequel il sera valable.

Le Trust and Reputation est le composant chargé d'établir et de gérer un environnement d'échange sûr et fiable. Pour y arriver, il analyse les informations en provenance du gestionnaire de contexte avant d'utiliser différents algorithmes pour établir la relation de confiance. Le Group Manager est chargé de sécuriser les échanges des informations contextuelles entre les objets. Pour cela, il utilise la technique *Ciphertext Policy Attribute Based Encryption* (CP-ABE) proposée par Bethencourt dans [BSW07]. Ce système permet à un utilisateur de chiffrer des données et de définir des politiques d'accès aux données chiffrées. Les politiques d'accès aux informations contextuelles sont définies par le group manager. A chaque fois qu'une information contextuelle doit être disséminée, les politiques CP-ABE seront utilisées pour chiffrer l'information contextuelle. Ceci permet aux seuls objets destinataires de déchiffrer cette information. Cependant, aucun détail n'a été donné sur l'implémentation de la solution proposée [RBS15].

### 2.4.4 Dynamic Context-Aware Scalable and Trust-Based IoT Security, Privacy Framework (DCASTBISPF)

Neisse et al. [NSB+15] ont adressé les problèmes de la sécurité et de la protection de la vie privée des citoyens dans le cadre des environnements suivants : smart city,

smart car et smart home. Ils ont proposé une architecture qui permet de définir des politiques de sécurité qui peuvent être déployées dynamiquement en fonction du contexte de l'utilisateur.

L'architecture proposée est essentiellement basée sur SecKit (fig. 3.6), une boîte à outils de sécurité développée par les mêmes auteurs [NSFB15]. SecKit utilise la gestion par politique de sécurité définie par l'IETF (Internet Engineering Task Force) [SHC+01]. Les principaux éléments de la gestion par politique sont le PEP (Policy Enforcement Point) et le PDP (Policy Decision Point). Le PEP est chargé d'appliquer les politiques de sécurité en fonction des différents contextes. Il se base sur les évaluations du composant PDP pour l'application des politiques de sécurité. Le PDP détecte les changements de contexte en provenance du Context Manager. Le Context Manager collecte les informations contextuelles à partir des objets et notifie au PDP les différents changements de contexte. Les règles de sécurité sont décrites dans les politiques de sécurité. La solution a été déployée dans le scénario d'une smart city. Les auteurs se sont focalisés sur l'architecture et le fonctionnement de la gestion de la sécurité par politiques sensibles au contexte. Ils ne sont pas aller plus loin dans les implémentations et les tests.

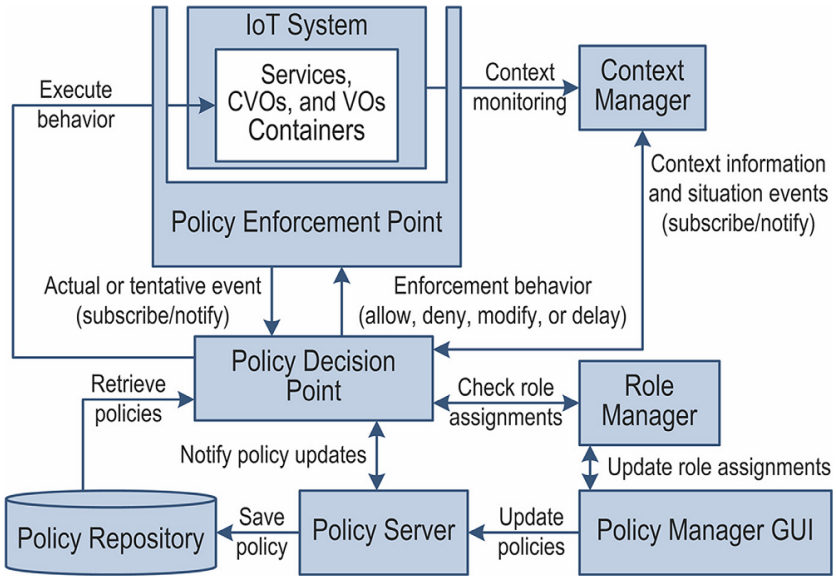



FIGURE 2.6 – Composants de la boite à outils SecKit [NSB+15]

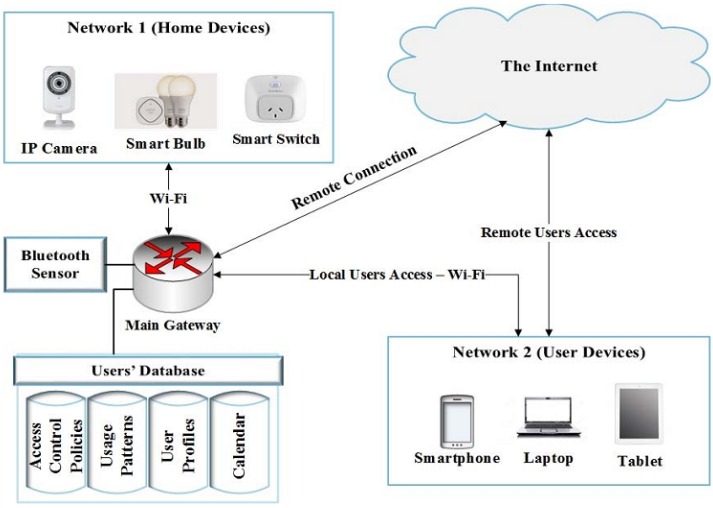



FIGURE 2.7 – Modèle d'authentification sensible au contexte [AKM17]

### 2.4.5 Context-Aware Authentication Service for Smart Homes (CASSH)

Ashibani et al. [AKM17] se sont concentrés sur la problématique de la dynamisation de l'authentification des utilisateurs mobiles dans les environnements smart homes. Ils ont proposé comme solution, un service d'authentification sensible au contexte pour les utilisateurs mobiles dans les environnements smart home.

Les principaux objectifs de ce système sont, d'une part, la mise en œuvre d'un modèle d'authentification dynamique pour les utilisateurs en leur permettant d'accéder aux services de smart home à l'aide d'informations d'identification traditionnelles et d'informations contextuelles et, d'autre part, un accès flexible et sécurisé aux services. L'architecture proposée combine plusieurs paramètres contextuels pour en déduire le type d'authentification à mettre en place. Les sources d'information contextuelles identifiées sont : le profil de l'utilisateur (noms, identifiant, etc.), la localisation (adresse IP et Bluetooth) et le calendrier de l'utilisateur et les informations sur l'historique (log et motifs d'accès).

La figure 2.7 illustre l'architecture proposée. Le point central de l'architecture est une gateway. Il s'agit d'une application installée sur une Raspberry pi. Elle est différente de la passerelle Internet habituelle parce qu'elle applique les mesures de sécurité en se basant sur les informations contextuelles. Par exemple, elle peut envoyer des commandes d'autorisation de redirection de port au pare-feu de la passerelle Internet via

SSH (Secure Shell) afin d'autoriser un utilisateur à accéder à un objet depuis Internet. L'architecture a été implémentée et ses performances ont été évaluées. Selon les auteurs, les résultats des évaluations prouvent l'efficacité du système, avec un temps de réponse moyen de 94 ms qui reste tout à fait acceptable si l'utilisateur accède depuis l'Internet.

### 2.4.6 Context-Based Security and Privacy for Healthcare IoT (CBS-PHIoT)

Alagar et al. [AAOW18] ont traité la problématique de la sécurité et de la protection de la vie privée centrées sur le patient dans le système Healthcare Internet of Things (HIoT). Le système HIoT est constitué essentiellement d'objets interconnectés pour les soins de santé (capteurs et dispositifs médicaux) et facilite les interactions avec les différents acteurs comme les patients, les médecins, les infirmiers, les pharmaciens, les autorités de régulation et les chercheurs en santé. Le système HIoT génère énormément de données sensibles rendant leur protection indispensable. En effet, les informations sur la santé des patients devraient être partagées seulement avec les médecins ayant le droit d'y accéder. La solution proposée permet d'assurer la sécurité et la protection de la vie privée dans le système HIoT avec l'usage des avancées technologiques apportées par l'IoT. La solution se base sur l'application de stratégies différentes dans des conditions ou contextes différents.

La figure 2.8 illustre l'architecture proposée. Cette architecture est composée principalement d'un système de supervision (SS : Supervisory System), d'une composante analytique (HBDAA : Healthcare Big Data Analytics Architecture) et d'une unité de contrôle (CU : Controller Unit). Le système de supervision comporte une unité pour l'authentification et une autre pour l'autorisation. En outre, il permet la validation de l'attribution des rôles aux acteurs (patients, médecins, etc.). Il permet également de certifier les capteurs et les objets de suivi médical qui doivent être autorisés dans le système. Cette certification concerne la conformité avec les réglementations sanitaires et les politiques gouvernementales en vigueur. Le HBDAA est contrôlé par le système de supervision. Son rôle est d'effectuer différents types d'analyses sur les interactions des objets (par exemple : respect de la règlementation en vigueur). Ces analyses interviennent lorsque de nouveaux objets prêts à l'emploi sont ajoutés au système.

L'architecture comprend un système de contrôle d'accès sensible au contexte dé-

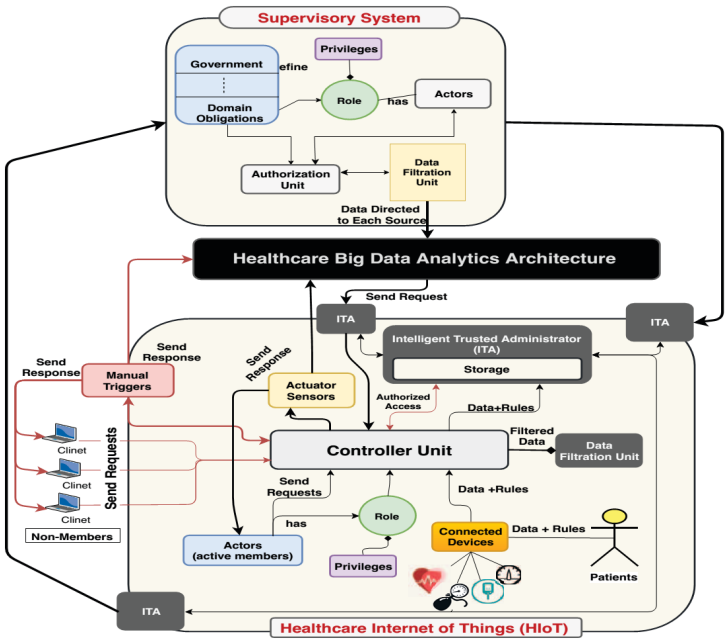



FIGURE 2.8 – Architecture de sécurité et de vie privée pour le HIoT [AAOW18]

nommé Contextual Role Based Access Control (CRBAC). Il est chargé de contrôler l'accès aux informations d'un patient. Les informations d'un patient sont représentées dans le système par le Personal Health Model (PHM). Les informations de suivi de santé du patient sont associées au PHM pour former l'Electronic Patient Record (EPR). Il utilise une conjonction de contraintes pour autoriser ou refuser une action. Les contraintes sont imposées à chaque acteur du système, et sont représentées par une paire (clé, valeur). La clé représente la dimension et la valeur une expression booléenne. La dimension est une combinaison des valeurs suivantes : qui, quand, quoi, où et pourquoi. Par exemple, un médecin possédant l'identifiant mid autorisé à lire la température d'un patient pid à travers le capteur tid. La contrainte de ce scénario est exprimée sous la forme : (1) quoi : accèsEnLecture(pid,tid), (2) qui : rôle ='médecin' et identifiant ='mid',(3) quand : temp = 'heure de travail', (4) lieu : 'chambre du patient(pid)', (5) raison : 'verifier fievre' [AAOW18]. Cette solution a été proposée dans le cadre d'un projet de smart city dans lequel la santé constitue un objectif majeur.

### 2.4.7 Edge-centric Context Sharing Architecture (ECSA)

Matos et al. [dTAH18b] ont proposé une architecture visant à résoudre les problèmes liés à la sécurité et à la protection de la vie privée dans l'IoT : Edge-centric Context Sha-

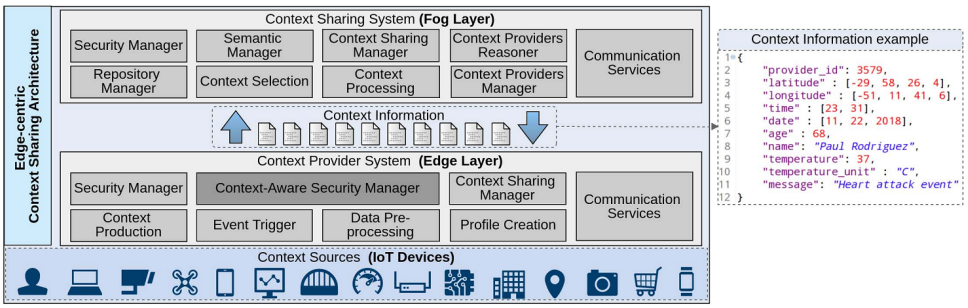



FIGURE 2.9 – Architecture ECSA [dTAH18b]

ring Architecture (ECSA). Cette architecture fournit une sécurité sensible au contexte en utilisant des informations contextuelles partagées. Le partage de contexte consiste à partager l'information contextuelle à tous les niveaux de l'architecture d'un système, afin qu'ils aient la même compréhension du contexte [PZCG21].

Selon les auteurs, les services de sécurité suivants doivent être sensibles au contexte : authentification, autorisation, contrôle d'accès et protection de la vie privée [dTAH18b]. L'environnement IoT est constitué de plusieurs objets et d'applications d'architectures différentes et distribuées. Le partage de contexte selon les auteurs permet de décharger les entités hétérogènes du traitement de l'information contextuelle. Cette réduction de charge de traitement est possible grâce à l'utilisation du contexte reçu, sans avoir à raisonner.

L'architecture proposée (Fig. 2.9) tire avantage du *Fog* et du *Edge Computing* afin d'améliorer l'évolutivité et de réduire la latence du réseau. En effet, elle comprend deux grands composants : Context Sharing et Context Provider. Le composant Context Sharing localisé dans la couche Fog a pour rôle le traitement et le partage des informations contextuelles en provenance des objets IoT. Localisé dans la couche Edge, le Context Provider est responsable de la génération des informations contextuelles et de l'application des décisions de sécurité. Une couche Cloud sert au stockage et à la coordination de l'ensemble des composants de l'architecture. Le composant de l'architecture responsable de la sécurité sensible au contexte est le Context-Aware Security Manager (CASM). Le CASM (Fig. 2.10) est constitué de plusieurs modules : Event Handler, Context Acquisition, ConSec Instance et Context Security Reasoner. L'Event Handler est chargé de la réception et de l'analyse des informations contextuelles. Il interprète l'information contextuelle dans le but de trouver l'action de sécurité correspondante. Le Context Ac-

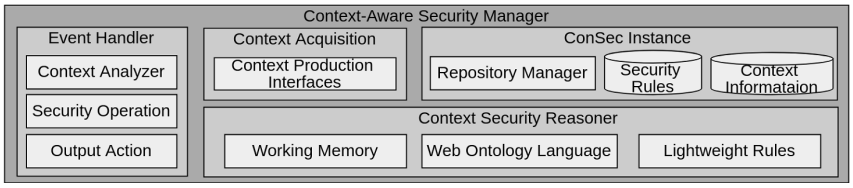



FIGURE 2.10 – Composant sécurité sensible au contexte de l'architecture ECSA [dTAH18b]

quisition est responsable de l'acquisition de contexte. Le Context Security Reasoner est chargé du processus de raisonnement du gestionnaire de sécurité sensible au contexte. Il est composé du Web Ontology Language, qui est responsable de la classification / modélisation du contexte pour le processus de raisonnement. Il se base sur le ConSec Instance pour vérifier la correspondance des informations contextuelles aux règles de sécurité préenregistrées. Dans cette proposition, l'opération de base pour fournir la sécurité sensible au contexte est l'utilisation de règles de sécurité prédéfinies. L'architecture ECSA a été implémentée et évaluée dans [dTAH18a].

Le tableau 2.2 résume les architectures proposées dans le cadre de la sécurité sensible au contexte dans les applications de la smart city. Dans ce qui suit, nous fournirons une comparaison entre ces différentes architectures en termes de gestion de la sensibilité au contexte et de services de sécurité fournis. Nous effectuerons, également, une comparaison entre ces différentes architectures en termes d'hétérogénéité et de passage à l'échelle.

## 2.5 Revue critique des besoins de sécurité sensible dans l'IoT

Cette section fait une revue critique des besoins de sécurité dans l'IoT et aborde les moyens permettant de rendre ces services centrés sur l'utilisateur, dynamiques et sensibles au contexte de l'utilisateur. Dans un premier temps, nous comparerons les projets selon un élément transversal à tous les services de sécurité et de protection de la vie privée sensibles au contexte : la gestion de la sensibilité au contexte. Ensuite, nous effectuerons une comparaison détaillée des projets étudiés en termes de protection de la vie privée et de services de sécurité sensibles au contexte. Ainsi, nous comparerons ces projets en fonction de l'approche utilisée (centrée sur l'utilisateur ou classique) et des mécanismes sensibles au contexte proposés.

Tableau 2.2 – Architectures de sécurité sensibles au contexte dans l'IoT

| Projets | CASA | CASF MA | MCIAS IoTE | DCAST BISPF | CASSH | ECSA | CBSPH IoT |
|---|---|---|---|---|---|---|---|
| Portée | PMC | PMC | IoT | IoT | IoT | IoT | IoT |
| Vie privée | Non | Non | Oui | Oui | Non | Oui | Oui |
| Authentification | Oui | Non | Oui | Oui | Oui | Oui | Oui |
| Contrôle d'accès | Non | Oui | Oui | Oui | Oui | Oui | Oui |
| Confidentialité | Non | Non | Oui | Non | Non | Non | Oui |
| Integrité | Non | Non | Non | Non | Non | Non | Non |
| Disponibilité | Non | Non | Non | Non | Non | Non | Non |
| Auditabilité | Non | Non | Non | Oui | Non | Non | Non |
| Sensible au contexte | Oui | Oui | Oui | Oui | Oui | Oui | Oui |
| Evalué | Oui | Oui | Non | Oui | Oui | Non | Non |
| Passage à l'échelle | Non | Non | Non | Oui | Non | Oui | Non |
| Hétérogénéité | Non | Non | Non | Non | Non | Oui | Non |

*PMC : Pervasive Mobile Computing, IoT : Internet of Things*

### 2.5.1 Gestion de la sensibilité au contexte

Dans les sous-sections suivantes nous nous intéresserons aux différents aspects de la gestion de la sensibilité au contexte (paramètres contextuels utilisés, gestion de la qualité du contexte, fréquence d'actualisation) dans les villes intelligentes. Nous effectuons une comparaison détaillée des projets étudiés suivant les paramètres contextuels utilisés, la fréquence d'actualisation du contexte et la gestion de la qualité du contexte. Ensuite nous identifions un certain nombre de travaux de recherche à mener pour une meilleure gestion de la sensibilité au contexte.

#### 2.5.1.1 Analyse critique des projets étudiés

Un élément essentiel de la sécurité sensible au contexte est la gestion de la sensibilité au contexte. La gestion de la sensibilité au contexte nécessite une bonne perception du contexte. La majorité des travaux étudiés a utilisé la sensibilité au contexte pour offrir une protection de la vie privée adaptative au contexte. Le tableau 2.3 résume la comparaison entre ces travaux en termes de gestion de la sensibilité au contexte. A part les projets MCIASIoTE et ECSA, tous les projets étudiés ont spécifiés les paramètres contextuels qui ont été utilisés. Dans les projets CASA et CASFMA, les auteurs ont utilisé une combinaison de la localisation géographique et de facteurs passifs (par exemple la

proximité Bluetooth du téléphone et d'un ordinateur) comme paramètres contextuels. Ces paramètres contextuels ne suffisent pas à déduire avec certitude le contexte de l'utilisateur. Ainsi, les solutions sont vulnérables à plusieurs attaques notamment l'imposture, et la localisation factice.

Dans le projet CASSH, les paramètres contextuels utilisés sont entre autres les informations du profil de l'utilisateur (nom, identifiant, etc.), la localisation (adresse IP et la proximité d'un périphérique Bluetooth) et l'agenda. La combinaison de ces paramètres est insuffisante pour une perception précise du contexte. En effet, l'utilisation de l'adresse IP et de la détection de proximité ne permet de faire qu'une approximation de la position géographique. De ce fait, la solution est vulnérable aux attaques d'usurpation d'adresse IP en plus des vulnérabilités des solutions CASA et CASFMA. L'utilisation du temps comme paramètre contextuel supplémentaire dans ces projets peut permettre de pallier aux attaques de localisation factice. De ce fait, avec le temps comme paramètre supplémentaire, il est possible d'apprendre les habitudes et de détecter les localisations erronées.

Les paramètres contextuels spécifiés dans le projet CBSPHIoT, c'est-à-dire, le temps et la localisation, peuvent permettre d'identifier le contexte de l'utilisateur, tout en évitant les attaques de localisation factice. Toutefois, il serait possible d'utiliser plus de paramètres contextuels afin de permettre une meilleure perception des différents contextes. Ainsi, dans le projet DCASTBISPF, les paramètres contextuels utilisés sont le temps, la localisation, le réseau et la vitesse de mouvement. Ces quatre paramètres peuvent effectivement permettre de déduire le contexte de l'utilisateur avec plus de précision par rapport aux autres projets. Néanmoins, grâce aux capteurs des dispositifs IoT disponibles à ce jour, il est possible d'avoir plus de paramètres contextuels, ce qui pourrait permettre de déterminer le contexte avec une grande précision et avec peu de faux positifs ou négatifs.

Ces problèmes de faux positifs ou négatifs pourraient être résolus grâce à la prise en compte de la qualité du contexte (QoC) [PZCG21]. Cependant, aucun projet étudié n'a mis en œuvre un mécanisme permettant de gérer la qualité du contexte. La sécurité sensible au contexte dans l'IoT, nécessite une actualisation en temps réel du contexte afin de déclencher automatiquement et aussitôt les mécanismes de sécurité appropriés. Cependant, aucun des projets étudiés n'a abordé la gestion de l'actualisation du contexte. Une actualisation du contexte en temps réel passe par une surveillance et une prise en

charge des événements contextuels en temps réel.

Le système de gestion de la sensibilité au contexte peut être menacé par des adversaires qui surveillent le système, essayent de reproduire les contextes pour tromper la perception du contexte par le système. Par conséquent, la sécurité du système devra être assurée. En ce sens, les informations contextuelles doivent être protégées des regards indiscrets. Ainsi, à part le projet MCIASIoTE, aucun des projets étudiés n'a abordé la sécurité du système de gestion de la sensibilité au contexte.

#### 2.5.1.2 Travaux de recherche à mener

Compte tenu des problèmes soulevés à la suite des comparaisons effectuées sur la gestion de la sensibilité au contexte dans les projets étudiés, il est évident que des mécanismes de gestion de la sensibilité au contexte efficaces sont nécessaires pour une sécurité sensible au contexte. Par conséquent, il est opportun d'effectuer des recherches afin d'améliorer la perception du contexte. Dans un premier temps, ces recherches doivent porter sur l'usage de paramètres contextuels supplémentaires comme la biométrie, les habitudes, l'analyse du comportement de l'utilisateur, en plus du temps, de la localisation et du réseau.

Dans un second temps, les recherches doivent être menées afin de permettre l'évaluation de la QoC. Cela permettra de pallier aux problèmes liés aux données contextuelles : incorrecte, ambiguïté, confusion, etc. Ces problèmes peuvent fausser la détermination du contexte en entraînant des faux positifs et négatifs. Par rapport à ce point, Chabridon et al. [CLD+14] ont effectué une revue des exigences pour l'évaluation de la qualité de contexte dans les systèmes IoT sensibles au contexte. Selon eux, la QoC et la vie privée sont liées, c'est-à-dire que la QoC peut être une donnée sensible (ex : la localisation). Ainsi, les recherches à effectuer sur la gestion de la QoC devront prendre en compte la préservation de la vie privée.

Par ailleurs, les projets étudiés se sont focalisés sur les mécanismes de sécurité sensibles au contexte et n'ont pas traité la gestion des évènements contextuels. Une sécurité sensible au contexte nécessite une bonne perception du contexte, donc une analyse en temps réel du contexte. A cet effet, des recherches devront être effectuées afin de permettre une meilleure surveillance et détection d'évènements contextuels en temps réel dans le but de permettre une gestion efficace de la sensibilité au contexte. En ce sens,

Tableau 2.3 – Comparaison de la gestion de la sensibilité au contexte dans les projets étudiés

| Projets | Paramètres contextuels | Limites |
|---|---|---|
| CASA [HDA+13] | • Localisation<br>• Facteurs passifs | • Paramètres contextuels insuffisants<br>• Absence de gestion de la QoC<br>• Vulnérable aux attaques d'imposture<br>• Absence de gestion de l'actualisation du contexte<br>• Gestion du contexte non sécurisée |
| CASFMA [MAA+14] | • Localisation<br>• Facteurs passifs | • Paramètres contextuels insuffisants<br>• Absence de gestion de la QoC.<br>• Vulnérable aux attaques d'imposture<br>• Absence de gestion de l'actualisation du contexte<br>• Gestion du contexte non sécurisée |
| MCIASIoTE [RBS15] | • Non spécifiés | • Absence de gestion de la QoC<br>• Absence de gestion de l'actualisation du contexte<br>• Gestion du contexte non sécurisée |
| CASSH [AKM17] | • Profil utilisateur<br>• Localisation<br>• Agenda<br>• Logs d'accès | • Vulnérable aux attaques d'usurpation d'adresse IP<br>• Absence de gestion de la QoC<br>• Gestion du contexte non sécurisée |
| DCASTBISPF [NSB+15] | • Temps<br>• Localisation<br>• Réseau<br>• Mouvement | • Mécanisme non implémenté et non évalué<br>• Absence de gestion de la QoC<br>• Gestion du contexte non sécurisée |
| ECSA [dTAH18b] | • Non spécifiés | • Absence de gestion de la QoC<br>• Absence de gestion de l'actualisation du contexte<br>• Gestion du contexte non sécurisée |
| CBSPHIoT [AAOW18] | • Temps<br>• Localisation | • Paramètres contextuels insuffisants<br>• Mécanisme non implémenté et non évalué<br>• Absence de gestion de la QoC<br>• Gestion du contexte non sécurisée |

les évènements jouent un rôle primordial dans l'IoT, car la détection des évènements amène le système à déclencher des actions de façon autonome (par déploiement de mécanismes de sécurité, de protection de la vie privée, etc.)[PZCG21].

Les attaques annoncées dans la section 2.5.1.1 peuvent sérieusement menacer la gestion de la sensibilité au contexte, et par conséquent le bon fonctionnement de la sécurité sensible au contexte. Seul le projet MCIASIoTE propose des mécanismes pour sécuriser les échanges des informations contextuelles. En ce sens, des recherches devront être menées afin de permettre un transfert sécurisé des informations contextuelles des sources (dispositifs IoT) vers le système de gestion du contexte. Ceci permettra de réduire les risques d'attaques d'interception et de modification des informations contextuelles lors du transfert. Des recherches devront également être effectuées pour permettre l'exclusion de tout dispositif non autorisé pour l'envoi d'informations contextuelles vers le système de gestion. Ainsi, ceci permettra au système de gestion de ne traiter que les informations contextuelles venant de dispositifs de confiance. Le gestionnaire de contexte devra également être protégé contre les attaques identifiées dans la section 1.3.2.

## 2.5.2 Protection de la vie privée

Dans les sous-sections suivantes, nous nous focaliserons sur les aspects de la protection de la vie privée. Nous effectuerons une comparaison détaillée des projets étudiés suivant les approches (centrée sur l'utilisateur/classique) utilisées et les différents mécanismes proposés. Nous terminerons par l'identification des recherches à mener pour une meilleure protection de la vie privée centrée sur l'utilisateur.

### 2.5.2.1 Analyse critique des projets étudiés

Nous avons effectué une étude approfondie des projets ayant intégré une sécurité sensible au contexte dans la section 2.4. Parmi ces projets, certains ont proposé des mécanismes de protection de la vie privée sensibles au contexte, en utilisant différentes approches (centrée sur l'utilisateur/classique). Le tableau 2.4 résume la comparaison de ces travaux en termes d'apports et de défis vis-à-vis de la protection de la vie privée sensible au contexte dans l'IoT.

L'approche centrée sur l'utilisateur pour la protection de la vie privée dans les applications IoT susmentionnées a été très peu abordée dans les projets étudiés. Dans le projet DCASBISPF, les auteurs ont mis en œuvre l'approche centrée sur l'utilisateur,

c'est-à-dire, en permettant à l'utilisateur d'être au centre des décisions relatives à la protection de sa vie privée. Ils ont considéré les questions de protection de la vie privée depuis la conception (privacy by design). Ils ont également pris en compte les questions liées à la fracture numérique, la facilité d'utilisation et la conformité avec les réglementations en vigueur. Selon les auteurs, l'utilisateur définit ses préférences en matière de protection de la vie privée pour chaque contexte et cela à travers une interface graphique. Selon les auteurs, cela permet à l'utilisateur de contrôler la divulgation de ses données, et cela en fonction du contexte. Néanmoins, la définition des préférences ne garantit pas à l'utilisateur que ses données sont effectivement divulguées comme il se souhaite. Cependant, le mécanisme mis en œuvre ne permet pas à l'utilisateur d'exercer un contrôle sur la collecte de ses données par ses différents dispositifs.

Les autres projets ayant pris en charge la protection de la vie privée (CBSPHIoT, ECSA et MCIASIoTE) ont utilisé une approche classique, c'est-à-dire, ne permettant pas à l'utilisateur de jouer un rôle dans la protection de sa vie privée. Les mécanismes ainsi proposés dans ces projets ne sont pas adéquats pour les applications IoT susmentionnées. En effet, l'utilisateur ne peut pas définir ses préférences en termes de protection de sa vie privée, ni exercer un contrôle effectif sur ses données. L'utilisateur n'est également pas sensibilisé aux conséquences éventuelles de ses actions sur sa vie privée. En outre, ces projets ne prennent pas en compte les réglementations en vigueur en termes de protection de la vie privée, par exemple le RGPD.

Enfin, la protection de la vie privée sensible au contexte doit utiliser la perception du contexte pour le déploiement des mécanismes idoines adaptés aux différents contextes. Le projet MCIASIoTE a mis en œuvre la protection de la vie privée sensible au contexte en utilisant l'anonymisation lors des échanges et cela en fonction du contexte de l'utilisateur. Le mécanisme proposé n'a pas été implémenté, ni évalué. Le projet DCASBISPF a mis en œuvre la pseudonymisation des identités en fonction du contexte et le retard dans la livraison des messages. Contrairement au projet MCIASIoTE, le mécanisme proposé a été implémenté. L'un des inconvénients de l'anonymisation et de la pseudonymisation est qu'elles ne permettent pas de protéger la localisation des utilisateurs et de leurs dispositifs. Les données ne sont également pas protégées au niveau des dispositifs [ASEJY18]. De plus, il est possible pour un adversaire d'identifier une personne en inférant sur les données ou en les recoupant avec d'autres données de la même personne, le risque de dé-anonymisation est d'autant plus grand avec les réseaux sociaux,

via lesquels les utilisateurs partagent plusieurs données [LLJ+17].

Le projet CBSPHIoT a également mis en œuvre l'anonymisation des données. Cependant, ce mécanisme, à la différence des mécanismes mis en œuvre dans les projets MCIASIoTE et DCASBISPF, n'est pas sensible au contexte. En outre, ce mécanisme n'est pas adéquat aux applications susmentionnées, car il ne tient pas compte de leurs caractéristiques (cf. section 1.2.8). De plus, comme dans le projet MCIASIoTE, le mécanisme proposé n'a pas été implémenté et évalué. Dans le projet ECSA, aucun détail n'est donné sur le mécanisme mis en œuvre. A l'instar des projets MCIASIoTE et CBSPHIoT, le mécanisme proposé n'a pas été implémenté et évalué.

#### 2.5.2.2 Travaux de recherche à mener

Au vu de tout ce qui précède, il est évident que des mécanismes standards de protection de la vie privée doivent être proposés. En effet, cela permettra d'uniformiser les techniques de protection de la vie privée à l'instar des systèmes cryptographiques standardisés et reconnus (par exemple : RSA : Rivest Shamir Adleman, AES : *Advanced Encryption Standard* [MS13]). En outre, les solutions proposées pour assurer la protection de la vie privée de manière adaptative au contexte dans les projets discutés ci-dessus ne proposent qu'un mécanisme comportant plusieurs limites.

Les attaques décrites dans la section 1.3.2, peuvent menacer la vie privée dans toutes les couches d'une architecture IoT. Dans la section 2.2.3, nous avons mis en avant les problèmes auxquels la protection de la vie privée des utilisateurs est confrontée dans les villes intelligentes. Dans un premier temps, les recherches à mener doivent porter sur la protection de la vie privée centrée sur l'utilisateur. Dans un second temps, les solutions à développer devront prendre en charge la sensibilité au contexte pour une protection efficace, en exploitant les caractéristiques des applications IoT susmentionnées. Ainsi, les solutions à proposer devront permettre de protéger l'identité et les données privées de l'utilisateur, et cela quel que soit son contexte.

Ces solutions devront mettre l'accent sur la possibilité pour l'utilisateur de définir ses préférences en matière de protection de la vie privée. En effet, les préférences en matière de vie privée varient d'un utilisateur à un autre. Les réglementations sont élaborées pour permettre une protection de la vie privée centrée sur l'utilisateur. A cet effet, les recherches à mener devront permettre de mettre en œuvre les recommandations issues

des réglementations en vigueur (par exemple le RGPD), et de les respecter.

Tableau 2.4 – Comparaison des travaux ayant proposé des mécanismes pour la protection vie privée sensible au contexte dans l'IoT

| Projets | Apports | Limites |
|---|---|---|
| MCIASIoTE [RBS15] | • Anonymisation de l'identité des objets et des utilisateurs en fonction du contexte | • Insuffisant pour une protection optimale de la vie privée<br>• Non centré sur l'utilisateur<br>• Absence de mécanismes permettant à l'utilisateur d'exercer un contrôle effectif sur ses données (collecte et divulgation)<br>• Mécanisme non évalué |
| DCASTBISPF [NSB+15] | • Pseudonymisation de l'identité de l'utilisateur selon la politique de sécurité contextuelle<br>• Retard de livraison des messages pour empêcher le pistage | • Identification de l'utilisateur possible en inférant sur les données<br>• Absence de contrôle sur la collecte des données |
| ECSA [dTAH18b] | • Règles de sécurité contextuelles | • Absence de détails sur les mécanismes et les techniques mis en œuvre<br>• Non centré sur l'utilisateur<br>• Mécanisme non évalué |
| CBSPHIoT [AAOW18] | • Anonymisation des données lors de l'extraction | • Mécanisme non évalué<br>• Non centré sur l'utilisateur<br>• Absence de mécanismes permettant à l'utilisateur d'exercer un contrôle effectif sur ses données (collecte et divulgation).<br>• Anonymisation des données non sensible au contexte |

Par exemple dans [BPGB18], les auteurs ont proposé une architecture de protection de la vie privée centrée sur l'utilisateur et respectant les réglementations. Ce mécanisme est indépendant des autres services de sécurité. Or, la protection de la vie privée devra être complémentaire aux services de sécurité (authentification, intégrité, contrôle d'accès, confidentialité, etc.) pour pallier les risques d'attaques menaçant la vie privée.

Plus concrètement, l'approche de la protection de la vie privée centrée sur l'utilisateur devra permettre à ce dernier de jouer un rôle central, c'est-à-dire, lui permettre d'exercer un contrôle effectif sur ses données. Ainsi, d'autres recherches devront être menées afin de permettre aux utilisateurs d'exercer un contrôle plus effectif sur la col-

lecte et la divulgation de leurs données en fonction du contexte, mais également de réduire la quantité de données collectées. Une piste possible est la proposition d'un module de gestion des données qui permettra de filtrer, en fonction des préférences de l'utilisateur, les données collectées par les objets avant leurs envois vers les entités consommatrices. Ce module permettra également de divulguer les données en fonction des préférences de l'utilisateur et des risques, en utilisant des mécanismes d'anonymisation appropriés. Par exemple Torre et al. ont proposé dans [TKSA16] un module similaire. Cependant, le module proposé n'est pas sensible au contexte. Comme pour l'architecture proposée dans [BPGB18], ce module ne comporte pas de service de sécurité complémentaire.

Au vu de ce qui précède, il est évident qu'un seul mécanisme de protection de la vie privée n'est pas suffisant. Cela s'explique par la différence entre les types d'informations sensibles à protéger : les identités, les données de localisation, les données sur le profil, etc. En ce sens, les recherches devront être menées afin de permettre la prise en charge d'une combinaison de plusieurs mécanismes de protection de la vie privée pour une protection optimale. Par exemple, une combinaison de l'anonymisation pour la protection des identités, de l'obscurcissement pour la protection des données de localisation géographique et les techniques de chiffrement et de gestion de données pour la sécurité des données (suppression périodique des données et suppression des parties identifiantes). La protection de la vie privée comprend également la sécurité des données au niveau du stockage et lors du transit. Ces points seront discutés dans les sections 2.5.4.1 et 2.5.5.

### 2.5.3 Authentification et contrôle d'accès

Dans les sous-sections suivantes, nous nous intéresserons aux mécanismes d'authentification et de contrôle d'accès sensibles au contexte proposés ou mis en œuvre dans les projets étudiés.

#### 2.5.3.1 Authentification

**2.5.3.1.1 Analyse critique des projets étudiés** La grande majorité des projets étudiés a proposé ou mis en œuvre un mécanisme d'authentification dans leur solution. Certains de ces mécanismes sont sensibles au contexte, d'autres non. Le tableau 2.5 résume la comparaison entre ces travaux pour une authentification sensible au contexte dans l'IoT. Nous avons présenté plus haut dans ce chapitre, plusieurs cas d'utilisation

des applications de villes intelligentes, dans lesquelles une authentification sensible au contexte pourrait résoudre les problèmes de sécurité et de facilité d'utilisation de ces applications. Ces applications sont généralement installées sur les smartphones et les tablettes afin de permettre aux utilisateurs d'interagir avec elles, l'approche sécurité centrée sur l'utilisateur étant axée sur les spécificités des utilisateurs dans la mise en œuvre de la sécurité. Dans ce contexte, plusieurs mécanismes d'authentification sensibles au contexte des utilisateurs sur ces applications ont été proposés.

Dans le projet CASA, le mécanisme d'authentification proposé permet de dynamiquement proposer à l'utilisateur une méthode d'authentification simple ou forte, dans le but d'apporter de la sécurité tout en lui facilitant l'accès à son téléphone. Ainsi, ce mécanisme s'inscrit dans l'approche centrée utilisateur, car il met l'accent sur la facilité d'utilisation. Cependant, le mécanisme mis en œuvre ne propose que l'usage d'un code PIN ou d'un mot de passe comme facteur d'authentification. Les mots de passe ou les codes PIN sont vulnérables à plusieurs attaques (par exemple, force brute). De plus, la solution est limitée à l'authentification des utilisateurs sur le système d'exploitation de leurs smartphones. Les applications hébergées sur le smartphone devront également être protégées contre les accès non autorisés. La solution est également vulnérable aux attaques soulignées dans la section 2.5.1.1, menaçant la gestion de la sensibilité au contexte.

Dans le projet CASSH, un mécanisme d'authentification sensible au contexte pour une smart home est proposé. La solution proposée s'inscrit dans une approche centrée sur l'utilisateur, car selon les auteurs, son utilisation quotidienne est plus robuste que celle proposée dans le projet CASA. En effet, elle utilise un mécanisme d'évaluation du niveau de confiance en plus de la sensibilité au contexte. Comme la solution CASA, cette solution utilise les facteurs d'authentification classiques (mot de passe). Ainsi, la solution est vulnérable aux attaques contre les mots de passe. En effet, la plupart des utilisateurs choisissent des mots de passe courts, faciles à deviner (date d'anniversaire d'un parent, nom d'un animal domestique, etc.) et à craquer. Bien que les auteurs aient utilisé plusieurs paramètres contextuels, la solution est vulnérable à certaines attaques décrites dans la section 2.5.1.1, menaçant le bon fonctionnement de la gestion de la sensibilité au contexte. Contrairement au projet CASA, les auteurs n'ont pas donné de détails sur le mécanisme mis en œuvre.

Les mécanismes d'authentification des utilisateurs proposés dans les projets MCIA-

SIoTE, CBSPHIoT et DCASTBISPF ne sont pas sensibles au contexte. La sensibilité au contexte du mécanisme proposé dans le projet ECSA n'a pas été démontrée. En premier lieu, ces mécanismes utilisent le couple nom d'utilisateur et mot de passe comme facteurs d'authentification. Ces mécanismes sont vulnérables à plusieurs attaques, notamment à l'attaque par force brute. En second lieu, les mécanismes proposés dans ces projets sont basés sur une approche de sécurité classique.

Par ailleurs, une authentification forte et fiable des dispositifs IoT auprès du système IoT est nécessaire, afin de s'assurer que les dispositifs connectés au système sont bien des dispositifs de confiance. Pour ce faire, chaque dispositif a besoin d'une identité unique qui lui permet d'être authentifié lors de ces échanges avec le système et qui doit utiliser un ou plusieurs facteurs d'authentification. A part les projets DCASTBISPF et CBSPHIoT, aucun des projets étudiés n'a mis en œuvre de mécanisme pour l'authentification des dispositifs. Cependant, les mécanismes d'authentification des dispositifs mis en œuvre dans les projets DCASTBISPF et CBSPHIoT ne sont pas sensibles au contexte. Néanmoins, contrairement au projet CBSPHIoT, le mécanisme mis en œuvre dans le projet DCASTBISPF a été implémenté et évalué. Cependant, les mécanismes mis en œuvre n'ont pas été décrits. Toutefois, nous notons l'absence de détail par rapport à la gestion de l'identification des dispositifs (par exemple, l'adresse EUI-64), aux facteurs d'authentification (certificat numérique, clé publique/privée, etc.), à la gestion des clés de session, etc.

**2.5.3.1.2 Travaux de recherche à mener** L'authentification des utilisateurs dans les applications IoT a été le service de sécurité le plus abordé dans les projets étudiés. Les projets CASA et CASSH se sont focalisés sur la problématique du compromis entre sécurité (authentification sur les applications) et facilité d'utilisation. Force est de constater que la réponse à cette problématique est non négligeable dans l'adoption de l'IoT par un large public. Il en est également de même pour l'approche sécurité centrée sur l'utilisateur. Cependant, les mécanismes d'authentification sensibles au contexte proposés dans ces projets sont vulnérables aux attaques menaçant la gestion de la sensibilité au contexte des différentes solutions (cf. section 2.5.1). Comme nous l'avons souligné dans la section 2.5.1.2, des recherches doivent être menées afin d'apporter des solutions à ces problèmes.

Par ailleurs, ces mécanismes d'authentification sensibles au contexte sont vulné-

Tableau 2.5 – Comparaison des travaux ayant proposé des mécanismes d'authentification sensible au contexte dans l'IoT

| Projets | Apports | Limites |
|---|---|---|
| CASA [HDA+13] | • Authentification sur téléphone mobile sensible au contexte<br>• Facilité d'utilisation | • Absence de plusieurs méthodes d'authentifications fortes (par exemple, double facteur, triple facteur)<br>• Absence de l'authentification des objets |
| CASSH [AKM17] | • Authentification sur application smart home sensible au contexte<br>• Facilité d'utilisation | • Mécanismes d'authentification proposés non décrits<br>• Absence de l'authentification des objets |
| ECSA [dTAH18b] | • Authentification sensible au contexte | • Absence de détails sur les mécanismes et les techniques mis en œuvre<br>• Absence de l'authentification des objets<br>• Mécanisme non évalué |

rables à plusieurs attaques connues contre les systèmes d'authentification. Cela s'explique par le fait que, quel que soit le contexte de l'utilisateur, c'est une authentification avec un seul facteur (code PIN ou mot de passe) qui est mise en œuvre. Ainsi, des recherches doivent être menées afin de trouver des mécanismes d'authentification, qui peuvent mettre en œuvre, en fonction du contexte, la méthode d'authentification adéquate, c'est-à-dire, simple (avec un seul facteur) dans un milieu comportant peu de risque, ou forte (avec plusieurs facteurs : empreinte digitale, reconnaissance faciale, etc. en plus d'un mot de passe fort) dans un milieu hostile avec un risque d'attaque élevé. Certaines applications IoT susmentionnées dans ce chapitre comportent souvent plusieurs autres utilisateurs en plus de l'utilisateur principal, par exemple la conjointe, les enfants ou d'autres proches. Dans ce contexte, des recherches doivent être menées afin de mettre davantage l'accent sur la possibilité pour l'utilisateur de facilement ajouter ou retirer les personnes qu'il veut. Cela devra se faire en lui permettant de définir qui peut s'authentifier sur l'application, comment doit-il s'authentifier et dans quel contexte.

Malgré les mécanismes proposés dans les projets étudiés, il reste beaucoup à faire pour l'authentification des objets dans le cadre des applications IoT susmentionnées. Les recherches à mener sur ce point doivent couvrir les axes suivants. D'abord développer des mécanismes d'authentification basés sur les protocoles cryptographiques allégés, l'usage de certificat numérique, ou une combinaison des deux, et cela en fonction du

contexte de l'utilisateur. Cela, permet d'empêcher le clonage, l'intrusion et le remplacement par des dispositifs illégitimes. Par exemple, dans [CRT17], Claeys et al. ont proposé une nouvelle technique d'authentification basée sur OAuth1.0a et ACE (*Authentication and Authorization for Constrained Environments*). Dans cette solution, un protocole cryptographique (EDHOC : *Ephemeral Diffie Hellman over Cose* (*Concise Object Signing and Encryption*)) est utilisé pour l'authentification.

Le deuxième axe concerne le renouvellement des clés de session en fonction du contexte. En effet, de manière générale, lorsqu'un dispositif IoT est authentifié, une clé de session lui est attribuée, et cela pour une durée non déterminée. Or, certaines attaques décrites dans la section 1.3.2, permettent de récupérer depuis les dispositifs compromis, les informations sensibles comme les clés cryptographiques : clé de session, clé publique, clé privée. Le renouvellement de la clé de session permet de pallier à la réutilisation de clés déjà utilisées, et donc d'empêcher un objet compromis de s'authentifier sur le système. Le troisième axe de recherche porte sur la sécurité physique des dispositifs IoT. La sécurité physique permet de pallier aux attaques d'extraction d'informations sensibles soulignées dans la section 1.3.2. Dans l'industrie, plusieurs propositions ont été faites en ce sens. Par exemple, nous avons Intel Software Gard Extension [2] (SGX) incorporé dans certains processeurs Intel et les processeurs CryptoIsland-300P et Cortex-M35P d'ARM [3]. Cependant, ces solutions sont propriétaires.

#### 2.5.3.2 Contrôle d'accès

**2.5.3.2.1 Analyse critique des projets étudiés** La plupart des projets étudiés a proposé ou mis en œuvre un mécanisme de contrôle d'accès sensible au contexte. Le tableau 2.6 fournit une comparaison de ces travaux.

Nous avons défini le contrôle d'accès dans le chapitre 1. Il permet d'assurer que seules les entités légitimes (utilisateurs, applications et services) accèdent aux ressources des objets et aux données. C'est pourquoi, il est essentiel de disposer d'un mécanisme de contrôle d'accès efficace et sensible au contexte, pour la mise en œuvre de l'approche sécurité centrée sur l'utilisateur dans les applications susmentionnées. Dans les projets CASSH et ECSA, un mécanisme de contrôle d'accès est annoncé. Cependant, rien n'est dit sur le mécanisme mis en œuvre, ni sur son implémentation. Cela peut

2. https://software.intel.com/en-us/sgx/details
3. https://www.arm.com/products/silicon-ip-security/physical-security-solutions

s'expliquer par le fait que d'une part, les auteurs du projet CASSH se sont focalisés sur la mise en œuvre d'une authentification sensible au contexte et, d'autre part, que les auteurs du projet ECSA se sont basés sur la gestion de la sécurité sensible au contexte.

Dans le projet CASFMA, le mécanisme proposé permet, en fonction du contexte de l'utilisateur, de contrôler l'accès des applications aux ressources réseaux. Les auteurs ont mis en œuvre l'approche centrée sur l'utilisateur. Contrairement au projet CASSH, le mécanisme proposé a été décrit par les auteurs. En effet, ils se sont focalisés sur le contrôle d'accès. Néanmoins, le mécanisme ne propose qu'un avertissement lorsque les risques pour de la vie privée sont élevés, l'utilisateur pouvant passer outre cet avertissement. De plus, seules les connexions TCP sont contrôlées, donc les connexions UDP sont ignorées. Par rapport aux deux projets précédents, le mécanisme proposé dans le projet MCIASIoTE est beaucoup plus adapté à l'IoT, car il permet de gérer les autorisations à travers l'utilisation de jetons d'autorisation contextuels. Cependant, les auteurs ont utilisé une approche classique, c'est-à-dire, qu'ils n'ont pas prévu de mécanisme permettant à l'utilisateur de dynamiquement gérer les autorisations (attribuer ou révoquer les autorisations). De plus, le mécanisme proposé n'a pas été implémenté ni évalué.

Le mécanisme de contrôle d'accès proposé dans le projet CBSPHIoT combine les attributs, les rôles et le contexte. L'avantage de ce mécanisme par rapport aux précédents est la possibilité de définir des autorisations avec une granularité élevée. En effet, il tire avantage du contrôle d'accès basé sur les rôles. Toutefois, la solution proposée dans ce projet n'est pas adaptée aux applications susmentionnées, car elle résout le problème de contrôle d'accès dans un système IoT hospitalier, dans lequel les patients ne sont pas des utilisateurs.

Dans le projet DCASTBISPF, les auteurs se sont basés sur la gestion par politiques afin de définir un mécanisme basé sur l'utilisation de règles d'accès contextuelles. Ces règles sont sous forme de listes de contrôle d'accès (ACL : *Access Control List*). Contrairement aux projets MCIASIoTE et CBSPHIoT, le mécanisme proposé dans le projet DCASTBISPF est centré sur l'utilisateur, adapté aux applications susmentionnées et a été implémenté. En ce sens, une interface graphique permet à l'utilisateur de définir les différentes autorisations. Néanmoins, l'utilisation des ACL peut être problématique, car les ACL sont assez complexes à manipuler pour un utilisateur et gourmandes en ressources de calcul pour les dispositifs IoT. Cependant, il faut noter que l'utilisation de jetons d'autorisation contextuels pourrait résoudre le problème de complexité des ACL soulevé dans le projet

DCASTBISPF.

Tableau 2.6 – Comparaison des travaux ayant proposé des mécanismes de contrôle d'accès sensible au contexte dans l'IoT

| Projets | Apports | Limites |
|---|---|---|
| CASFMA [MAA+14] | • Contrôle de l'ouverture des applications mobiles sensible au contexte | • Mécanisme de contrôle à titre informatif seulement, n'empêche pas l'utilisateur d'ouvrir les applications<br>• Ne contrôle pas les connexions UDP<br>• Insuffisant pour un contrôle d'accès optimal |
| MCIASIoTE [RBS15] | • Jeton d'autorisation pour un contrôle d'accès sensible au contexte | • Absence de dynamisme dans l'attribution et la révocation des jetons d'autorisations contextuels<br>• Proposition non évaluée à ce jour |
| DCASTBISPF [NSB+15] | • Contrôle d'accès dynamique et sensible au contexte | • Gestion complexe des ACL<br>• Pas de passage à l'échelle |
| ECSA [dTAH18b] | • Règles de sécurité contextuelles | • Absence de détails sur les mécanismes et les techniques mis en œuvre<br>• Non centré sur l'utilisateur<br>• Mécanisme non évalué |
| CBSPHIoT [AAOW18] | • Contrôle d'accès basé sur les rôles et sensible au contexte | • Absence de dynamisme dans l'attribution et la révocation des jetons d'autorisations contextuels<br>• Pas de passage à l'échelle<br>• Proposition, non implémentée à ce jour |

**2.5.3.2.2 Travaux de recherche à mener** Le contrôle de l'accès est un service essentiel dans la sécurité et la protection de la vie privée dans l'IoT. En effet, il permet de définir une granularité de niveau d'accès aux informations de l'utilisateurs et aux ressources des dispositifs. Les mécanismes implémentant le contrôle d'accès sensible au contexte sont présents dans la plupart des projets étudiés. Malgré les solutions proposées, beaucoup reste à faire dans la mise en œuvre du contrôle d'accès sensible au contexte dans les applications IoT susmentionnées.

Dans un premier temps, des recherches doivent être conduites dans le but de mieux mettre en œuvre l'approche centrée sur l'utilisateur. Cela passe par la définition de mécanismes permettant à l'utilisateur de définir ses préférences et cela avec une granula-

rité élevée. Ces mécanismes doivent également permettre de préserver la vie privée et doivent être adéquats pour des dispositifs IoT faibles en ressources. En ce qui concerne les mécanismes de contrôle d'accès adéquats pour les dispositifs IoT, plusieurs propositions ont été faites, notamment le contrôle d'accès basé sur la cryptographie. Plusieurs variantes de ces mécanismes ont été mis en œuvre et ont prouvé leur efficacité. Par exemple, dans [CRT17], un système de contrôle d'accès sécurisé basé sur les jetons est mis en œuvre. L'avantage principal d'un tel système est qu'il être utilisé dans des environnements réseaux non sécurisés.

Dans un second temps, des recherches doivent être menées sur la sensibilité au contexte de ces mécanismes. Par exemple, il pourrait être possible de définir des jetons d'autorisation contextuels comme proposé dans le projet MCIASIoTE, mais basé sur la solution mise en œuvre dans [CRT17]. En effet, contrairement au mécanisme proposé dans le projet MCIASIoTE, le mécanisme proposé dans [CRT17] est basé sur ACE [SDK18] et OAuth et ne nécessite pas de connexion sécurisée pour la gestion des jetons. Cependant, ce mécanisme n'est pas centré sur l'utilisateur, c'est-à-dire que l'utilisateur n'a pas la possibilité de définir et de gérer les autorisations. Un mécanisme de contrôle d'accès centré sur l'utilisateur devra permettre à ce dernier d'être un acteur central de la gestion des autorisations.

Dans un troisième temps, des recherches devront être menées dans le but de proposer des mécanismes qui seront centrés sur l'utilisateur, c'est-à-dire en lui permettant d'attribuer ou de révoquer des autorisations contextuelles à la volée. Ensuite, la mise à disposition de ces jetons auprès des entités destinataires pourrait être effectuée en toute sécurité et de manière distribuée en utilisant les registres distribués tel que la blockchain [ZXD+18]. En effet, la blockchain permettra de conserver l'anonymat de l'utilisateur tout en assurant la confiance que seules les entités auxquelles les jetons sont destinés pourront les utiliser.

Enfin, des recherches doivent également porter sur le passage à l'échelle des solutions qui seront proposées. Cela permettra à la solution d'être efficace, au fur et à mesure que le nombre de dispositifs et d'applications d'un utilisateur augmentera.

### 2.5.4 Sécurité des communications

Lors de la création de modèles de menaces pour l'IoT, les attaquants ciblent les canaux de communication réseau en premier lieu. Ce ciblage est dû aux nombreuses

vulnérabilités présentes au niveau des différentes couches de l'architecture (cf. section 1.3.2). Par conséquent, les communications devront être sécurisées afin de pallier aux attaques menaçant le système et les applications. La sécurité des communications comprend la confidentialité et l'intégrité des données ainsi que l'authentification de leur origine. Les projets étudiés ont peu, voire pas du tout abordé la sécurité des communications. Cette sécurité est très importante à partir du moment où les données sensibles transitent entre les différents objets, équipements, acteurs et lieux de stockage. Le tableau 2.7 compare les projets étudiés en termes de sécurisation des communications d'une manière sensible au contexte.

#### 2.5.4.1 Confidentialité

La confidentialité est primordiale pour assurer la transmission de données en toute sécurité et contribue à la protection de la vie privée dans l'IoT lorsque les données privées ou liées aux identités sont transmises.

**2.5.4.1.1 Analyse critique des projets étudiés** La confidentialité des communications a été très peu évoquée dans les projets étudiés. A part les projets CBSPHIoT et MCIASIoTE, la confidentialité n'a été abordée dans aucun des autres projets étudiés.

Dans le projet CBSPHIoT, la confidentialité des communications a été annoncée. Les auteurs ont proposé l'utilisation de la norme IEEE 802.15.6 afin d'assurer la confidentialité des données transmises hors des capteurs. Cette norme est utilisée dans les communications Wireless Body Area Network (WBAN) et vise à garantir la confidentialité, l'intégrité et l'authentification de l'origine des données échangées. La portée des communications WBAN est limitée aux alentours du porteur, en l'occurrence au niveau d'une passerelle très proche (en termes de distance) du porteur. D'une part, la limite d'une telle approche est que la confidentialité des données n'est pas assurée au niveau des protocoles réseaux, transport ou application. D'autre part, les communications extra-WBAN, par exemple avec une infrastructure Fog ou cloud, ne seront pas confidentielles. Or, la confidentialité des communications devra être assurée au niveau de chaque nœud prenant part à la communication. Par conséquent, la proposition du projet CBSPHIoT est insuffisante.

L'architecture proposée dans le projet MCIASIoTE, est basée sur les composants d'ARM IoT-A (cf. section 1.2.4.3). Par rapport au projet CBSPHIoT, cette solution pro-

pose une meilleure solution, qui consiste à assurer la confidentialité de bout en bout avec le concept de tunnel [BBF+13]. Cependant, même si cette solution peut être sûre, elle n'est pas toujours réalisable à cause des ressources restreintes en énergie et en calcul des dispositifs IoT. Toutefois, les mécanismes proposés dans les deux projets ne sont pas sensibles au contexte.

**2.5.4.1.2 Travaux de recherche à mener** Au vu de ce qui précède, force est de constater que malgré les menaces auxquelles la confidentialité des communications est confrontée, elle n'a pas été traitée en priorité dans les projets étudiés. L'absence de confidentialité des communications dans l'IoT expose les données des utilisateurs aux attaques pouvant provenir d'Internet mais également des réseaux locaux. Les recherches à mener sur ce service afin d'assurer des communications sécurisées dans les réseaux et les applications IoT susmentionnées doivent couvrir les axes suivants. Le premier axe porte sur la mise en œuvre de protocoles de communication sans fil robustes pour les dispositifs IoT afin d'éliminer les vulnérabilités [ZDWM17]. Ces protocoles devront utiliser des systèmes cryptographiques allégés, efficaces et adaptés aux dispositifs de l'IoT afin d'assurer la confidentialité des données échangées dans les réseaux sans fil. Par exemple, parmi les cryptos systèmes disponibles pour l'IoT, la cryptogrphie à courbe elliptique (*Elliptic Curve Cryptography* - ECC) prend le devant du fait de la sécurité fournie et sa faible consommation de ressources [Dub18].

Dans les applications IoT susmentionnées, certains dispositifs utilisent une passerelle pour communiquer sur Internet tandis que d'autres peuvent directement communiquer sur Internet. Cependant ces communications peuvent passer par plusieurs réseaux. Elles sont ainsi exposées à plusieurs attaques (cf. 1.3.2). C'est pourquoi, le deuxième axe porte sur la mise en œuvre en fonction du contexte de la confidentialité des données lors de ces communications. Certains travaux ont tenté d'apporter une solution à la problématique de la confidentialité des communications. Par exemple, dans [GM19, GMS14, WM17], les différents auteurs se sont inspirés du protocole IPSec en proposant des protocoles similaires au VPN (Virtual Private Network), c'est-à-dire, mettant en œuvre la confidentialité des communications au niveau de la couche réseau.

Cependant, la mise en œuvre de ces protocoles n'est pas sensible au contexte. Une autre solution est la sécurité au niveau de la charge utile du message de la couche application. Compte tenu des ressources restreintes d'un grand nombre de dispositifs,

l'utilisation de protocoles de chiffrement allégés et de mécanismes de signature cryptographique pour chiffrer et signer les messages de la couche application peut permettre à ces dispositifs de communiquer sur des réseaux non sécurisés. Par exemple, dans OSCAR [VTR+15], les auteurs ont proposé un mécanisme permettant l'envoi et la réception de paquets de données CoAP chiffrés sur un réseau non sécurisé. Cependant, la solution est basée sur l'utilisation d'un serveur de confiance. Dans [AAC+18], les auteurs se sont basés sur OSCAR pour proposer IoTChain, A la différence d'OSCAR, la confiance est décentralisée et gérée par la blockchain. Toutefois, la mise en œuvre de ces solutions n'est pas sensible au contexte.

Comme nous l'avons décrit dans la section 2.2, dans une ville intelligente les habitants (utilisateurs) sont mobiles. Ceci entraîne un changement fréquent de réseau utilisé pour les communications. En effet, un mécanisme permettant d'assurer la confidentialité des communications dans un réseau local (maison/entreprise) ne suffit plus dès que les communications passent par un réseau public (café, aéroport, etc.). C'est pourquoi le troisième axe devra porter sur la sensibilité au contexte de la confidentialité des communications. Ainsi, en fonction du contexte de l'utilisateur et des risques, un canal de communication confidentiel pourrait être établi.

#### 2.5.4.2 Intégrité des données et authentification de l'origine

La sécurité d'une communication passe par la mise en œuvre de mécanismes permettant la vérification de l'intégrité des données et de leurs origines.

**2.5.4.2.1 Analyse critique des projets étudiés** A part le projet CBSPHIoT, aucun des projets étudiés n'a proposé ou mis en œuvre de mécanismes permettant d'assurer l'intégrité des données lors des communications. Cela s'explique par le fait que la plupart des projets se sont focalisés sur la mise en œuvre d'un service de sécurité sensible au contexte. Dans le projet CBSPHIoT, les auteurs proposent l'utilisation de la norme 802.15.6 afin d'assurer l'intégrité des données échangées au sein du WBAN. Les WBAN ne représentent qu'une partie des réseaux utilisés parmi les applications IoT susmentionnées. Ainsi, la solution proposée est très limitée.

L'authentification des données, définie dans la section 1.3.3.3.1.1 du chapitre 1, permet de s'assurer de l'authenticité de l'origine des données reçues. Cependant, ce

service n'a pratiquement pas été abordé dans les projets étudiés. Cela est dû à la non prise en compte de la sécurisation des communications par la plupart de ces projets.

**2.5.4.2.2 Travaux de recherche à mener** Généralement, les protocoles déployés pour assurer la confidentialité des données permettent également d'assurer l'intégrité des données et de vérifier l'authenticité de leurs origines. Par exemple, la norme IEEE 802.15.4 assure l'intégrité des données par les mécanismes suivants : AES-CBC-MAC-X, AES-CCM-MAC-X, où X représente 64, 128 ou 256 selon la variante de la suite de chiffrement utilisée.

Jusque-là les mécanismes proposés pour assurer l'intégrité et l'authentification des données ne concernent que la couche liaison de données. Ces services devront être assurés dans les communications de la couche réseau. Des mécanismes permettant d'assurer l'intégrité et l'authentification des données sont présents dans les spécifications du protocole IPv6. Cependant ceux-ci ne sont pas automatiquement implémentés.

Par exemple, les mécanismes mis en œuvre dans les solutions permettant d'assurer la confidentialité des données mettent également en œuvre l'intégrité et l'authentification des données. Néanmoins, cela nécessite l'établissement d'un tunnel de communication sécurisé. IPv6 étant le protocole utilisé pour l'acheminement des paquets dans l'IoT, des recherches devront être menées afin de permettre la mise en œuvre des mécanismes, par exemple, ceux prévus dans les spécifications d'IPv6 pour assurer l'intégrité et l'authentification des données, et cela même en l'absence d'un tunnel de communication sécurisé IPSec. Ces mécanismes seront mis en œuvre par des objets IoT non contraints et qui servent généralement d'intermédiaire dans les communications.

Enfin, les recherches doivent être menées pour la mise en œuvre de mécanismes permettant d'assurer l'intégrité et l'authentification des données au niveau de la couche application. Ces mécanismes devront être adaptés aux dispositifs IoT, faibles en ressources (cf. section 1.2.7.1, tableau 1.2). En effet, protéger les données au niveau de la couche application est un moyen sûr et allégé d'assurer la sécurité des communications dans l'IoT, et cela même si les communications s'effectuent sur un réseau non sécurisé.

### 2.5.5 Sécurité des données stockées

La sécurité des données devra également être assurée au niveau du stockage. Elle est assurée par la mise en œuvre de la confidentialité et de l'intégrité des données. Dans les

Tableau 2.7 – Comparaison des projets proposé des mécanismes permettant d'assurer la sécurité des communications dans l'IoT

| Projets | Apports | Limites |
|---|---|---|
| MCIASIoTE [RBS15] | • Confidentialité des communications sur Internet par tunneling | item [•] La confidentialité des communications n'est pas de bout en bout et n'est pas sensible au contexteProposition non évaluée à ce jour |
| CBSPHIoT [AAOW18] | • Confidentialité de la transmission dans un réseau WBAN | • La confidentialité ne va pas au-delà du réseau WBAN<br>• La confidentialité des communications n'est pas de bout en bout et n'est pas sensible au contexte<br>• Proposition, non implémentée à ce jour |

sous-sections suivantes, nous effectuerons une comparaison détaillée des projets étudiés en termes de mécanismes de sécurité des données proposés ou mis en œuvre. Nous terminerons par l'identification d'un certain nombre de travaux de recherche à mener pour assurer la sécurité des données dans les applications IoT de villes intelligentes.

#### 2.5.5.1 Analyse critique des projets étudiés

La sécurité des données stockées a été très peu prise en compte dans les projets énoncés. Elle devra être assurée au moment de la collecte, la transmission et jusqu'au stockage. A part le projet CBSPHIoT, aucun des projets étudiés n'a proposé ou mis en œuvre de mécanisme permettant d'assurer la sécurité des données au niveau du stockage. Dans le projet CBSPHIoT, les données sont chiffrées et stockées dans le cloud. Selon les auteurs, le chiffrement des données est réalisé en utilisant la paire de clés (PID, DID). Le PID (*Patient IDentifier*) représente l'identifiant du patient et le DID (*Device IDentifier*) représente l'identifiant de l'objet source des données. Non seulement les auteurs n'ont pas spécifié le système de chiffrement utilisé, mais la gestion des clés de chiffrement n'a pas été décrite.

#### 2.5.5.2 Travaux de recherche à mener

Compte tenu de tout ce qui précède, un certain nombre de travaux reste à faire pour assurer la sécurité des données stockées. Les mécanismes à mettre en œuvre pour assu-

rer la sécurité des données stockées devront, également, permettre d'assurer l'intégrité et la confidentialité des données. Dans ce contexte, des recherches devront être menées afin de trouver des suites de chiffrement robustes, efficaces et moins gourmands en temps et en ressources pour assurer la confidentialité des données stockées (cela dépend du lieu de stockage). Les solutions à proposer devront également assurer l'intégrité des données stockées par le hachage et l'horodatage. Les mécanismes actuels de confidentialité des données mettent en œuvre les crypto-systèmes tels que AES et ECC (*Elliptic Curve Cryptography*) [ARC18].

Cependant, la gestion efficiente des clés de chiffrement/déchiffrement reste l'un des défis à relever. Dans les systèmes de chiffrement symétrique, la même clé est utilisée pour chiffrer et déchiffrer les données. L'entité ayant chiffrée les données doit partager la clé en toute sécurité avec l'entité souhaitant déchiffrer les données. Quant au système de chiffrement asymétriques, chaque entité possède son couple de clé publique/privée. Les clés sont généralement codées en dur dans les différents dispositifs. Souvent ces dispositifs ne disposent pas de mécanisme pour la mise à jour de ces clés. Dans ce contexte, des recherches doivent être menées afin de permettre une mise à jour sécurisée des différentes clés cryptographiques. Des recherches devront également être menées afin d'assurer la sécurité physique des dispositifs IoT. Cela permettra d'empêcher les attaques visant l'extraction des clés cryptographiques.

D'autres recherches doivent être menées afin de permettre l'accès aux données chiffrées de l'utilisateur sans déchiffrement préalable. En effet, cela prend un temps considérable pour déchiffrer de grandes quantités de données. Une des solutions possibles est l'utilisation d'un système de chiffrement homomorphe. Le chiffrement homomorphe est un système cryptographique qui permet d'effectuer des opérations sur les données chiffrées sans déchiffrement préalable. Ainsi, les données seront toujours en état chiffré lors de leur traitement. Cependant, le chiffrement homomorphe est complexe à mettre en œuvre et très gourmand en ressources. Un chiffrement homomorphe allégé a été proposé dans [SHB+17] pour les services cloud mobiles. Ainsi, des recherches doivent être effectuées pour un chiffrement homomorphe adapté au contexte de l'IoT afin de permettre l'accès aux données chiffrées sans déchiffrement préalable.

### 2.5.6 Hétérogénéité et passage à l'échelle

L'hétérogénéité et le passage à l'échelle sont des caractéristiques importantes de l'IoT. Dans les sous-sections suivantes, nous effectuerons une comparaison détaillée des projets étudiés en termes de support de l'hétérogénéité et de passage à l'échelle. Nous terminerons par l'identification d'un certain nombre de travaux de recherche à mener pour permettre l'hétérogénéité et le passage à l'échelle dans les applications IoT des villes intelligentes.

#### 2.5.6.1 Analyse critique des projets étudiés

Les objets composants un système IoT possèdent différents systèmes de communication, différentes capacités de calcul et de mémoire. Parmi les projets étudiés, seuls DCASTBISPF et ECSA ont considéré l'hétérogénéité et le passage à l'échelle.

En ce qui concerne l'hétérogénéité, le projet DCASTBISPF permet l'implémentation d'un PEP spécifique à chaque groupe d'objets. Cela permet à la solution de supporter divers types d'objets. Dans le projet ECSA, les auteurs ont également annoncé la prise en charge de l'hétérogénéité. Etant donné que l'architecture ECSA est centrée sur le partage de contexte, l'hétérogénéité est gérée dans le module Semantic Manager. Cependant, le mécanisme proposé dans le projet DCASTBISPF permet de mieux prendre en charge l'hétérogénéité que celui du projet ECSA, car le Semantic Manager de ce dernier ne gère que le partage de contexte entre les objets. Cependant, ce mécanisme ne gère pas l'application des règles de sécurité prévue dans l'architecture.

#### 2.5.6.2 Travaux de recherche à mener

Le passage à l'échelle est l'un des défis à relever pour un système IoT, car le nombre d'objets présents dans l'architecture peut être très dense (106 équipements au $Km^2$ [Dum16]). Le passage à l'échelle a été très peu abordé dans les projets étudiés et n'a pas été identifié comme un critère à prendre en compte. Ainsi, des recherches devront être menées afin de permettre le passage à l'échelle dans les applications susmentionnées tout en assurant le même niveau de sécurité et protection de la vie privée.

L'hétérogénéité est une caractéristique fondamentale que toute architecture de sécurité dans l'IoT devra prendre en charge. Des recherches devront être menées pour la

prise en charge de l'hétérogénéité dans la mise en œuvre de la sécurité. Le *Software Defined Networking* (SDN) pourrait résoudre le problème de l'hétérogénéité dans l'application des règles de sécurité au niveau des objets. En effet, SDN pourrait être utilisée pour l'application des politiques de sécurité. De plus, la distribution de l'architecture SDN devrait faciliter le passage à l'échelle.

## 2.6 Conclusion

Dans ce chapitre, nous avons commencé par introduire l'importance de l'utilisateur dans les applications de la smart city. En effet, ces applications ont un impact significatif sur la vie privée des citoyens et elles représentent plusieurs risques pour la sécurité.

Par la suite, nous avons décrit les limites des méthodes de sécurité classiques dans ces applications. Ensuite, nous avons présenté et décrit l'approche sécurité et protection de la vie privée centrées sur l'utilisateur dans ces applications. Compte tenu des caractéristiques de ces applications IoT (intelligence, sensibilité au contexte, temps réel, etc.), il nous a semblé opportun d'utiliser la sécurité sensible au contexte comme outil pour la mise en œuvre de l'approche sécurité et protection de la vie privée centrées sur l'utilisateur. L'un des atouts majeurs de la sécurité sensible au contexte est la mise en œuvre dynamique des services de sécurité en fonction du contexte de l'utilisateur. En effet, la sécurité sensible au contexte nous permet d'exploiter les caractéristiques des applications IoT pour une mise en œuvre efficace de l'approche sécurité et protection de la vie privée centrées sur l'utilisateur.

La sécurité sensible au contexte a fait l'objet de plusieurs travaux de recherche. Ainsi, nous avons étudié les travaux de recherche qui ont proposé une solution de sécurité sensible au contexte dans les applications susmentionnées. Nous avons par la suite effectué une revue critique des besoins de sécurité dans ces applications et abordé les moyens permettant de rendre ces services centrés sur l'utilisateur, dynamiques et sensibles au contexte de l'utilisateur. Ce qui ressort de cette revue, est que beaucoup de travaux de recherche doivent être menés pour la mise en œuvre d'une sécurité sensible au contexte efficace et, par la même, une mise en œuvre efficace de l'approche sécurité et protection de la vie privée centrées sur l'utilisateur dans les applications susmentionnées. Dans le chapitre suivant, nous proposerons une architecture de sécurité et de protection de la vie privée centrées sur l'utilisateur dans la smart city qui répond aux principaux défis identifiés dans ce chapitre.

# Chapitre 3

# Vers une architecture de sécurité et de protection de la vie privée sensibles au contexte en tant que service dans l'IoT

## 3.1 Introduction

Les applications de l'Internet des objets permettent d'offrir des services avancés et intelligents qui facilitent la vie quotidienne des utilisateurs.

En ce sens, elles permettent de collecter des données du monde physique et de les envoyer dans le monde virtuel (Internet) en exploitant les importantes avancées réalisées dans le domaine des technologies de l'information et de la communication : collecte des données, transmission des données via différentes techniques de communication (filaire, sans fil et mobile), stockage et analyse des données, etc. Selon le type d'application (e-santé, maison intelligente, réseau de véhicules, réseau électrique intelligent, etc.) et ses caractéristiques (temps réel, critique, etc.), ces différentes étapes peuvent avoir des besoins spécifiques, tels que la transmission sécurisée ou encore l'analyse des données en temps réel. Dans cette thèse, nous nous intéressons au domaine de la ville intelligente car, non seulement c'est un domaine d'actualité, mais aussi parce qu'il comprend un certain nombre d'applications IoT intéressantes telles que la e-santé, la maison intelligente, le réseau électrique intelligent et les réseaux de véhicules. Dans ce contexte, l'analyse des données peut être effectuée en temps réel et le résultat peut être utilisé dans la prise de décision.

La mise en œuvre des applications IoT de la ville intelligente pourrait avoir un impact négatif sur la sécurité et la vie privée des utilisateurs [BZWL19, GKS18, IVH16, MS18]. En effet, les dispositifs (capteurs, actionneurs, etc.) et les applications (e-santé, maison intelligente, etc.) utilisés peuvent comporter des risques liés à la sécurité et à la vie privée des utilisateurs (divulgation, espionnage, utilisation sans consentement, vol,

etc.). Ces problèmes ont été abordés dans les chapitres 1 et 2 ainsi que dans un bon nombre de travaux [AAOW18, CRT17, KW15, dTAH18a, NSB+15]. La majorité des solutions proposées se sont concentrées sur un service de sécurité spécifique (par exemple, l'authentification et le contrôle d'accès ou la confidentialité des données) ou sur la protection de la vie privée.

Par ailleurs, ces solutions ne tiennent pas compte du profil de l'utilisateur, telles que ses préférences en matière de protection de la vie privée, sa grande mobilité, sa convivialité, ses habitudes d'utilisation, etc. Pour résoudre ces problèmes (sécurité limitée à certains services et inadaptée au profil de l'utilisateur), l'accent doit être mis sur une approche centrée sur l'utilisateur. En raison de son importance et de sa pertinence pour les services et applications numériques, y compris l'IoT, l'ETSI a adopté une série de normes pour cette approche dans l'IoT, notamment les normes TR 103 438, EG 203 602 et TR 103 603 [ALC+19a, ALC+19b, SAL+19]. L'ETSI a défini l'approche centrée sur l'utilisateur dans la norme TR 103 438 comme une approche dans laquelle l'utilisateur est au centre du système [ALC+19b]. Ainsi, comme spécifiée dans la section 2.2.3, la sécurité et la protection de la vie privée centrées sur l'utilisateur permettent à ce dernier de jouer un rôle central dans la mise en place d'une protection adaptée. Cela nécessite la mise en œuvre de mécanismes de sécurité et de protection de la vie privée en fonction de certaines informations pertinentes sur l'utilisateur (par exemple, les préférences, le contexte) et sans une intervention explicite.

En outre, les mécanismes de sécurité et de protection de la vie privée spécifiés dans de nombreux travaux de recherche sont proposés ou mis en œuvre pour répondre à des modèles de menace spécifiques auxquels l'utilisateur est exposé. Comme la situation d'un utilisateur donné peut changer en raison de nombreux facteurs (par exemple, la mobilité), les modèles de menace peuvent changer également. Par conséquent, pour assurer une sécurité optimale et adaptée au modèle de menace correspondant à la situation de l'utilisateur, la mise en œuvre des mécanismes de sécurité doit prendre en compte tout changement dans cette situation. Cela permet de s'assurer de la bonne prise en charge des menaces et des vulnérabilités identifiées.

La sécurité et la protection de la vie privée sensibles au contexte constituent un moyen efficace de mettre en œuvre une sécurité et une protection de la vie privée centrées sur l'utilisateur. Ce moyen nous permettra également de nous attaquer au problème de l'évolution des modèles de menace en raison des changements fréquents de

contexte des utilisateurs en exploitant les connaissances des dispositifs IoT. En effet, d'une part, il permet à l'utilisateur de contrôler la mise en œuvre des mécanismes de sécurité et de protection de la vie privée dans l'IoT. D'autre part, il permet à l'utilisateur d'être protégé en permanence, en déployant dynamiquement des mécanismes de sécurité et de protection de la vie privée qui répondent au modèle de menace caractérisant le contexte actuel de l'utilisateur sans son intervention. Aussi, la protection de la vie privée en fonction du contexte peut permettre à l'utilisateur de contrôler la collecte des données (quelles données ? quand ? quels dispositifs ? pour quelle application et quels sont les risques encourus ?) et leur transmission (par exemple, obscure, anonyme, etc.) en fonction du contexte.

L'authentification sensible au contexte peut contribuer à améliorer la facilité et la sécurité de l'accès aux applications IoT. La gestion des autorisations basée sur le contrôle d'accès sensible au contexte peut permettre à l'utilisateur de contrôler qui accède à quelle donnée, quand et dans quelle situation il peut y accéder. La sécurité des communications tenant compte du contexte peut aider à identifier les communications qui doivent être sécurisées dans une situation donnée (réseaux sous-jacents, nature des extrémités, types de données, etc.) et à définir la meilleure manière de les sécuriser.

À cet égard, différentes propositions ont été introduites dans plusieurs travaux. Ces travaux sont présentés et comparés dans la section 3.2. Cependant, à notre connaissance, aucune de ces propositions ne définit une solution mettant en œuvre les exigences de sécurité et de protection de la vie privée sensibles au contexte. Ainsi, nous présentons dans ce chapitre une solution de sécurité et de protection de la vie privée sensibles au contexte. Cette solution met en œuvre la gestion sécurisée de la sensibilité contextuelle, la protection de la vie privée, la confidentialité, l'authentification, le contrôle d'accès et la sécurité des communications (cf. Section 2.5). En outre, pour répondre aux exigences des nouvelles architectures réseau, la sécurité et la protection de la vie privée de l'IoT pourraient être basées sur l'approche "*as a service*", variante de l'architecture orientée service (*Service Oriented Architecture* - SOA). Cela lui permet de fournir plus de flexibilité, de dynamicité, d'évolutivité, de services personnalisés et un meilleur support de l'hétérogénéité des applications et de la mobilité des utilisateurs [AALS16].

C'est pourquoi nous avons basé notre solution sur l'approche "*as a service*" pour garantir la sécurité et la protection de la vie privée sensibles au contexte pour l'IoT

(*Context-Aware Security and Privacy as a Service* : CASPaaS). Ce service sera capable de fournir une protection continue de la sécurité et de la vie privée de l'utilisateur, et de supporter les exigences fonctionnelles et non fonctionnelles identifiées [AALS16].

La principale innovation de cette nouvelle architecture réside dans l'introduction de deux nouveaux plans. D'abord, le Plan de Connaissance qui sera responsable de la gestion indépendante de la sensibilité au contexte à travers l'utilisation de l'intelligence artificielle et de la gestion de la qualité du contexte (QoC). Ensuite, le Plan de sécurité et de protection de la vie privée qui sera en charge de la mise en œuvre de mécanismes de sécurité et de protection de la vie privée sensibles au contexte par la composition dynamique de services contextuels. En ce sens, chaque service représente un mécanisme spécifique de sécurité ou de protection de la vie privée.

Le reste de ce chapitre est organisé comme suit : la section 3.2 présente et compare les différents travaux qui ont été réalisés sur la sécurité et la protection de la vie privée s'adaptant au contexte dans l'IoT. La section 3.3 introduit les technologies abordées dans ce chapitre. La section 3.4 détaille notre proposition, à savoir l'architecture de sécurité et de protection de la vie privée sensibles au contexte en tant que service. Enfin, la section 4.5 conclut le chapitre.

## 3.2 Travaux connexes

La sécurité et la protection de la vie privée sensibles au contexte dans l'IoT ont fait l'objet de plusieurs travaux dans la littérature. Malgré ces travaux, plusieurs défis restent à relever. Dans cette section, nous comparons les différentes solutions proposées dans ces travaux. Cela nous permettra par la suite d'identifier les problèmes à résoudre afin d'avoir une meilleure sécurité sensible au contexte : gestion sécurisée de la sensibilité au contexte, mise en œuvre des services de protection de la vie privée et de sécurité sensibles au contexte, etc. Ainsi, nous soulignerons les limites de ces solutions et nous détaillerons le besoin d'une nouvelle solution pour une sécurité et une protection de la vie privée sensibles au contexte dans l'IoT, et notamment dans le domaine de la ville intelligente.

### 3.2.1 Solutions proposées

Une solution de sécurité et de protection de la vie privée adaptative au contexte dans les applications IoT de la ville intelligente a été proposée dans [NSB+15]. Cette solution

met en œuvre une gestion des politiques de sécurité basée sur le contexte. Elle utilise une combinaison de plusieurs paramètres contextuels (heure, localisation, réseau, vitesse) pour la perception du contexte. Elle permet à l'utilisateur de définir certaines de ses préférences (divulgation de données, contrôle d'accès, etc.). L'utilisation de ces paramètres contextuels aide à déterminer le contexte avec plus de précision. Cependant, cette proposition ne s'est intéressée qu'à l'implémentation de la sécurité basée sur les politiques. Elle ne considère pas la sécurité de la gestion des informations contextuelles. Ainsi, ce mécanisme est vulnérable aux attaques d'usurpation d'identité, de fausse localisation, etc. La solution décrite dans [RBS15] met également en œuvre la sécurité et la protection de la vie privée en fonction du contexte. Contrairement à la solution proposée dans [NSB+15], le système de gestion de la connaissance du contexte proposé met en œuvre une sécurité de gestion des informations contextuelles. Néanmoins, la gestion de la QoC n'est pas prise en compte dans ces travaux. Par conséquent, les contextes déterminés par ces solutions peuvent être sujets à des conflits.

La protection de la vie privée sensible au contexte est complémentaire aux mécanismes de sécurité sensibles au contexte dans l'IoT. Dans ce sens, dans [NSB+15], les auteurs ont décrit un mécanisme de protection de la vie privée sensible au contexte basé sur la pseudo-anonymisation et la livraison différée des messages. La livraison différée des messages peut empêcher le suivi de l'utilisateur, par exemple dans la géolocalisation. Dans [RBS15], les auteurs ont présenté un système de protection de la vie privée sensible au contexte basé sur l'anonymisation des données de l'utilisateur. Cependant, la pseudo-anonymisation et l'anonymisation sont vulnérables aux attaques par inférence sur les données de l'utilisateur. Dans [dTAH18b], un module de sécurité sensible au contexte et offrant une protection de la vie privée adaptative au contexte est décrit. Cependant, les auteurs n'ont pas fourni de détails sur la technique de protection de la vie privée utilisée dans ce module.

Pour mettre en œuvre la sécurité sensible au contexte dans l'IoT, différents mécanismes peuvent être utilisés. L'authentification est le premier service de sécurité qui permet de vérifier l'accès à un système. À cette fin, dans [AKM17], les auteurs se sont focalisés sur la mise en œuvre de l'authentification adaptative au contexte. Le mécanisme proposé utilise une combinaison du nom de l'utilisateur et de son mot de passe comme facteur d'authentification, ce qui le rend vulnérable aux attaques de divulgation de mot de passe. En outre, les auteurs de [dTAH18b] et [NSB+15] ont abordé

l'authentification et le contrôle d'accès sensibles au contexte dans les applications IoT. Cependant, le module de sécurité sensible au contexte proposé dans [dTAH18b] ne définit pas spécifiquement comment l'authentification et le contrôle d'accès sont sensibles au contexte.

En ce qui concerne le service de contrôle d'accès sensible au contexte, dans [RBS15], les auteurs ont proposé un mécanisme de contrôle d'accès sensible au contexte basé sur des jetons d'accès contextuels. Cependant, ce mécanisme ne permet pas à l'utilisateur de définir dynamiquement et à la volée des autorisations, et n'a pas la flexibilité nécessaire pour supporter les fonctionnalités susmentionnées. En outre, le système de gestion des autorisations est centralisé, ce qui peut causer un problème de goulot d'étranglement ou un point de défaillance unique.

La mise en œuvre de la sécurité des communications en fonction du contexte dans l'IoT permet de sécuriser suffisamment les communications selon que les réseaux et les protocoles de communication sous-jacents sont sécurisés ou non. Cependant, aucun des travaux étudiés n'a proposé un mécanisme de sécurité des communications tenant compte du contexte.

### 3.2.2 Positionnement

Les travaux décrits ci-dessus proposent des solutions de sécurité sensibles au contexte dans l'IoT. Cependant, ces solutions ne répondent pas aux exigences identifiées. Le tableau 3.1 résume la comparaison de ces solutions en matière de services de sécurité offerts. Cette comparaison nous permet de conclure que la prise en charge des exigences fonctionnelles, c'est-à-dire les mécanismes de sécurité sensible au contexte proposés sont pour la plupart incomplets pour l'IoT. Par ailleurs, ces travaux ont abordé la question de sécurité et de protection de la vie privée sensibles au contexte dans le cadre d'une application bien précise. Dans l'IoT, chaque utilisateur peut disposer de plusieurs appareils et applications. Ainsi, compte tenu de la diversité des applications IoT de la ville intelligente, proposer une solution qui permet de répondre aux exigences identifiées indépendamment des applications et des dispositifs IoT des villes intelligentes devient nécessaire.

En outre, la nécessité d'évoluer vers une architecture orientée logicielle dans l'IoT est de plus en plus forte. Cela est principalement dû au fait que l'Architecture Orientée Service (AOS) met en œuvre un modèle basé sur les composants, qui permet de

concevoir un système décomposé en parties fonctionnelles [AALS16]. En effet, en l'état actuel, la mise en œuvre de la sécurité sensible au contexte dans l'IoT est confrontée à plusieurs défis, tels que : la dynamicité, la flexibilité, le support de la mobilité des utilisateurs, la personnalisation des services et la généricité. De même, l'hétérogénéité, la mutualisation et l'évolutivité sont également des défis à surmonter pour faire face à la forte densité des dispositifs IoT.

Tableau 3.1 – Comparaison des travaux qui ont proposé des solutions de sécurité et de protection de la vie privée sensibles contexte dans l'IoT

| | | Mécanisme et Architecture | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Authentification | Contrôle d'accès | Sécurité des communications | Vie privée | Sensibilité au contexte sécurisée | Composition dynamique des services | Intégration arch. de réf. UIT-T |
| Travaux | [NSB+15] | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ |
| | [RBS15] | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ |
| | [AKM17] | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| | [dTAH18b] | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ |
| | [AAOW18] | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ |
| | [BPGB18] | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ |
| | Notre proposition | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

C'est pourquoi, nous proposons une nouvelle architecture de sécurité sensible au contexte basée sur le paradigme "*as a service*". Cette nouvelle architecture prend en charge les questions de : la sécurité et la protection de la vie privée sensibles au contexte, la dynamicité, la flexibilité, mais aussi la mobilité, la personnalisation des services de sécurité et de protection de la vie privée et le support des applications génériques de l'IoT, notamment pour la ville intelligente. Ainsi, un Plan de connaissance permettant une gestion efficace de la sensibilité du contexte et un Plan de sécurité et de protection

de la vie privée permettant la mise en œuvre de mécanismes de sécurité et de protection de la vie privée sensibles au contexte comme une composition de services basés sur le contexte, la confiance et la sécurité de l'architecture, sont proposés.

## 3.3 Concept Fondamentaux

Dans cette section, nous introduisons les principaux concepts abordés dans ce chapitre en relation avec notre contribution. Ce sont l'informatique en tant que service ('*as a service*'), les nouvelles architectures pour les réseaux de communication telles que le réseau défini par logiciel (*Software-Defined Network* – SDN) et la virtualisation des fonctions réseau (*Network Function Virtualization* – NFV).

### 3.3.1 Informatique 'as a service'

L'informatique '*as a service*' est une variante plus flexible de l'information orientée service basée sur l'architecture orientée service (*Service Oriented Architecture*) et dont la portée est de niveau applicatif. En effet, dans ces architectures, la conception des systèmes et applications est basée sur les composants spécialisés réutilisables, encore appelés des services ou des micro-services. Les composants lâchement couplés, réutilisables et spécialisés travaillent indépendamment et collaborent pour l'accomplissement des tâches des applications.

Ces propriétés facilitent la mise en œuvre des applications de plus en plus distribuées, qui étaient jusque-là conçues suivant l'approche monolithique. En outre, l'informatique en tant que service permet de profiter de nouveaux paradigmes informatiques tels que la conteneurisation et la composition de service, le cloud natif, etc. La composition de service permet de construire des applications distribuées et à large échelle, flexibles, dynamiques et adaptées pour l'IoT. Ainsi, les entreprises et les fournisseurs peuvent mieux profiter des avantages offerts par le Cloud Computing et les nouvelles architectures réseau.

### 3.3.2 Software-Defined Networking

Le concept *Software-Defined Networking* (SDN) ou réseau défini par logiciel en français a vu le jour avec la première version d'OpenFlow 1.0 en 2009 [ONF09]. En 2011, plusieurs entreprises et constructeurs informatiques se sont joints (notamment Microsoft, Facebook, Google, Deutsche Telecom, Verizon etc.) pour créer la fondation Open

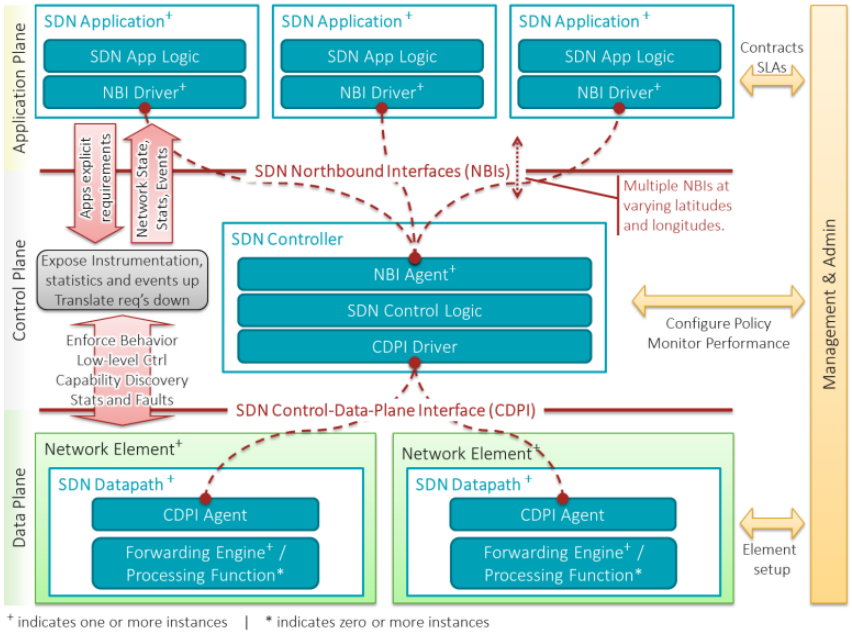



FIGURE 3.1 – Vue générale de l'architecture SDN [ONF13]

Networking Foundation (ONF). La figure 3.1 représente l'architecture proposée par l'ONF. L'objectif principal de la technologie SDN est d'affranchir les réseaux IP traditionnels de leurs nombreuses limites comme l'absence d'interopérabilité entre les équipements des différents constructeurs, la complexité de l'implémentation et de l'extensibilité des services, les coûts (CAPEX et OPEX) élevés, etc. [Puj19]. La technologie SDN répartit l'architecture réseau en trois couches distinctes, séparant le plan de contrôle du plan de données. Ces couches sont : la couche Application, la couche Contrôle et la couche Infrastructure. Le point central de cette architecture en couches est le contrôleur. En effet, ce dernier a une vue globale du réseau et peut injecter directement sur chaque équipement les règles appropriées au moment opportun. La couche Infrastructure est constituée de matériels SDN (switch, routeur etc.) [Men20]. La couche Application est la couche de programmation et d'abstraction. Les avantages offerts par la technologie SDN sont l'optimisation des ressources, la flexibilité du réseau, une vue globale des ressources disponibles dans le réseau, un routage amélioré et une meilleure utilisation de la bande passante. Ces avantages rendent cette technologie particulièrement adaptée pour les réseaux IoT et les réseaux véhiculaires, très dense, hétérogènes et nécessitant le passage à l'échelle.

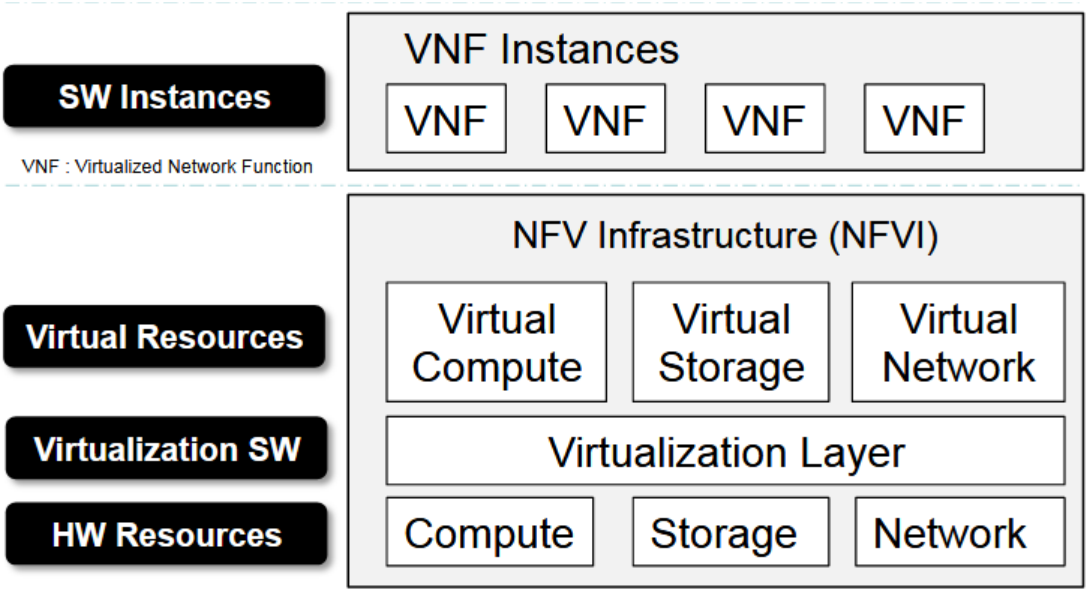



FIGURE 3.2 – Architecture ETSI NFV [Ers13]

### 3.3.3 Network Function Virtualization

La technologie *Network Function Virtualization* (NFV) ou virtualisation des fonctions réseaux en français, a pour but de séparer les fonctions réseaux (routeur, pare-feu, etc.) des équipements de réseau [Puj19, ZAR21]. Cette séparation permet l'utilisation de ressources réseau sans se soucier de leur localisation ou du matériel utilisé. NFV est standardisée par un groupe de travail de l'European Telecommunications Standards Institute (ETSI). Ainsi, une architecture de référence composée de trois couches a été proposée. Ce sont les couches "fonctions réseaux virtualisées" (*Virtualised Network Functions*),"infrastructure NFV" (NFVI) et "gestion et orchestration NFV" (NFV *Management and Orchestration*) [Ers13]. La figure 3.2 représente les couches de l'architecture de référence NFV. Les VNF sont les implémentations logicielles des fonctions réseaux classiques qui peuvent être déployées dans une infrastructure NFV (NFVI). L'infrastructure NFV est constituée de l'ensemble des configurations matérielles et logicielles sur lesquelles les VNF doivent être déployées. Comme ressources matérielles, nous avons les unités de calcul, de stockage et de réseau physique, et comme ressources logicielles nous avons les unités de calcul, stockage et de réseaux virtuels. La couche gestion et orchestration comprend l'orchestrateur NFV, le gestionnaire de fonction réseau virtuelle et le gestionnaire d'infrastructure virtuelle (Virtualized Infrastructure Manager – VIM).

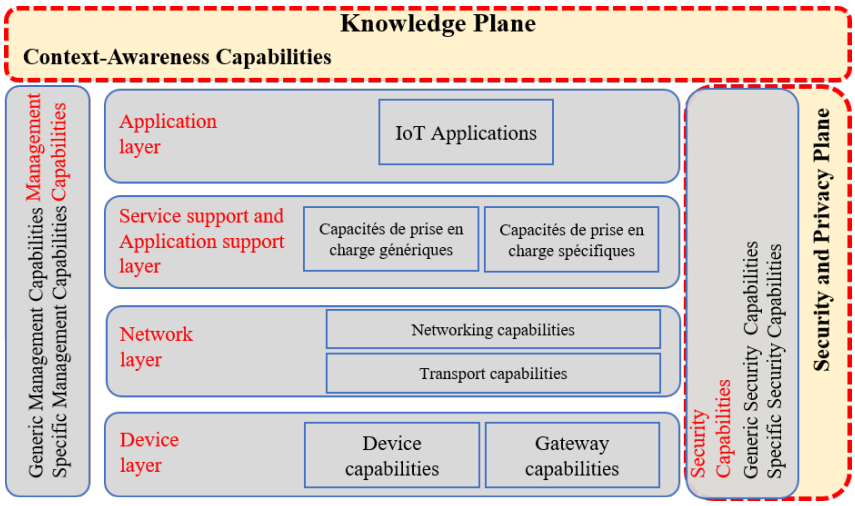



FIGURE 3.3 – Architecture de référence de l'UIT-T intégrant notre architecture

L'orchestrateur NFV gère les ressources, les services réseaux et leur cycle de vie, etc. Le gestionnaire de fonction réseau virtuelle est chargé de gérer le cycle de vie des instances VNF. Le gestionnaire d'infrastructure virtuelle est en charge de la gestion des ressources de la NFVI (calcul, stockage et réseau). Il reporte également les performances et les évènements de l'infrastructure.

## 3.4 Architecture de sécurité et de protection de la vie privée en tant que service

Dans cette section, nous présentons notre contribution, à savoir, la sécurité et la protection de la vie privée sensibles au contexte en tant qu'architecture '*as a service*'. En plus de la mise en œuvre efficace de mécanismes de sécurité sensibles au contexte, cette architecture relève les défis suivants : la dynamicité, la flexibilité, la mobilité, la personnalisation, la composition automatique de services et la prise en charge des applications IoT génériques par l'exposition sécurisée des API.

### 3.4.1 Vue d'ensemble

L'architecture de référence de l'UIT-T pour l'IoT intègre une couche transversale pour assurer la sécurité entre les différentes couches de l'architecture de référence (UIT, 2012). L'architecture que nous proposons vise à intégrer cette couche comme une capacité de sécurité spécifique afin de fournir une sécurité sensible au contexte en tant que

service pour l'IoT. Elle vise également à intégrer un Plan de Connaissance dans l'architecture de référence ITU-T IoT pour offrir des fonctionnalités de sensibilité au contexte dans la couche de gestion. Ainsi, notre proposition permettra à l'architecture de référence de l'UIT-T de prendre en charge la sécurité et la protection de la vie privée de l'utilisateur, tout en étant compatible avec l'intégration des réseaux de prochaines générations. La figure 3.3 illustre l'intégration de l'architecture proposée dans l'architecture de référence UIT-T pour l'IoT.

L'architecture que nous avons proposée est une innovation majeure dans la mise en œuvre de la sécurité sensible au contexte dans l'IoT. Pour être efficace, la sécurité et la protection de la vie privée sensibles au contexte nécessitent une séparation de la gestion de la sensibilité du contexte et la mise en œuvre des mécanismes de sécurité et de protection de la vie privée. En d'autres termes, cela représente la séparation de l'intelligence (informatique sensible au contexte) et de l'application des décisions de sécurité. Cette séparation permet une plus grande modularité et davantage de flexibilité. Ainsi, les complexités de la gestion de la sensibilité au contexte et de la mise en œuvre des mécanismes de sécurité et de protection de la vie privée peuvent être gérées plus facilement. En effet, pour prendre en charge les limites des travaux précédents dans lesquels la gestion de la sensibilité au contexte et la mise en œuvre des mécanismes de sécurité sensibles au contexte ne sont pas séparées, nous proposons de diviser notre architecture en deux plans : Le Plan de Connaissance (PC) et le Plan de Sécurité et de Protection de la Vie Privée (PSPVP). La figure 3.4 illustre les plans de l'architecture.

Les plans de gestion, de contrôle et de données de l'architecture de référence de l'UIT-T pour l'IoT ont besoin de capacités de connaissance du contexte pour permettre une gestion dynamique et souple des réseaux IoT (orientation dynamique du trafic, services tenant compte de la localisation de l'utilisateur, etc.). Par conséquent, le Plan de Connaissance pourra convenir à ces plans de l'architecture de référence de l'UIT-T pour l'IoT. Grâce à l'approche '*as a service*', l'architecture peut être intégrée dans de nouvelles architectures orientées services. L'approche orientée service permet de relever plusieurs défis en matière de sécurisation de l'IoT, notamment la dynamicité, la flexibilité, l'hétérogénéité, la mobilité et la personnalisation. De plus, cette approche permet l'évolutivité de l'architecture proposée. En effet, elle permet l'instanciation à la demande des services nécessaires. Les modules composant les différents plans sont donc conçus selon les exigences de l'approche '*as a service*' présentée pour la conception des fonctions de

réseau virtuel dans [BBS18]. Par conséquent, la sécurité et la protection de la vie privée dans les applications IoT seront dynamiques, flexibles, hétérogènes, personnalisables et centrées sur l'utilisateur.

Chaque plan de l'architecture joue un rôle clé :

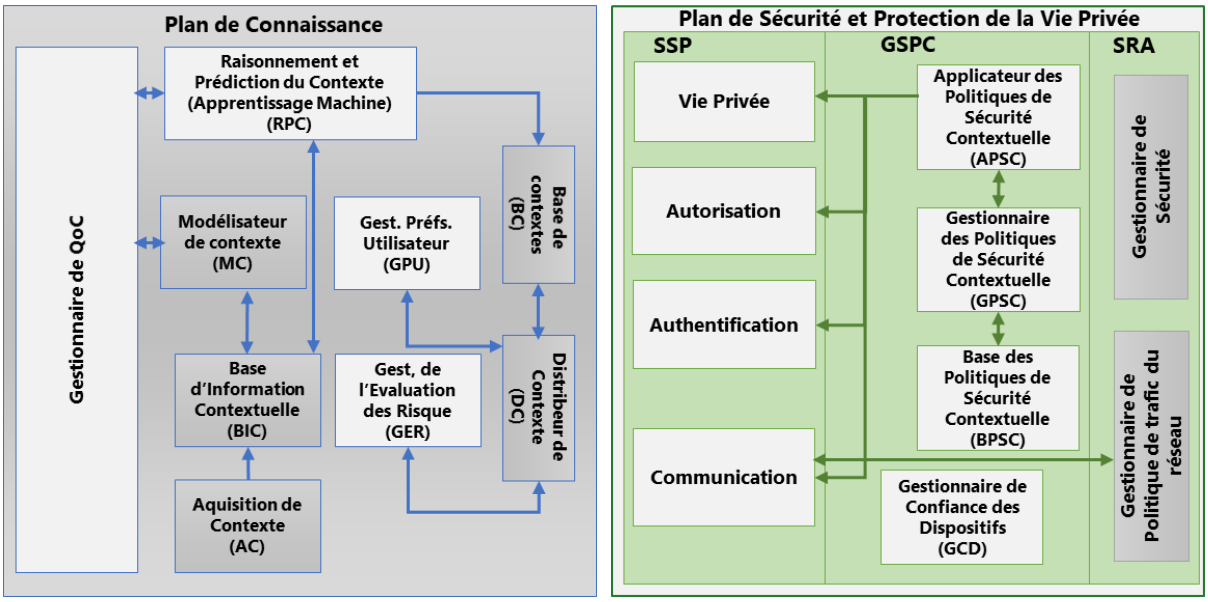
— le Plan de Connaissance vise à améliorer le fonctionnement du PSPVP et du plan de gestion de l'architecture de référence de l'UIT-T pour l'IoT. En effet, il fournira une gestion de la sensibilité au contexte en temps réel pour une sécurité et une protection de la vie privée sensibles au contexte et efficaces. Il est responsable de l'acquisition, la modélisation, la détermination et la distribution du contexte au PSPVP. Il permettra au PSPVP de composer dynamiquement les services/micro-services appropriés en fonction du contexte et des préférences de l'utilisateur. La gestion de la qualité du contexte (QoC) combinée à l'apprentissage automatique et aux techniques de traitement en temps réel permettra la détection automatique du nouveau contexte. Elle permettra également de disposer de capacités de sensibilisation au contexte pour la couche de gestion de l'architecture de référence de l'IoT-T.

— le Plan de Sécurité et de Protection de la Vie Privée (PSPVP) est responsable de la mise en œuvre de mécanismes de sécurité et de protection de la vie privée sensibles au contexte, tels que l'authentification, le contrôle d'accès, la sécurité des communications et la protection de la vie privée. En effet, en fonction du contexte fourni par le PC, le PSPVP composera les services et ou micro-services nécessaires pour la mise en œuvre des mécanismes requis pour ce contexte.

L'exemple de mise en œuvre de notre architecture sera basé sur le scénario suivant : Bob est un patient diabétique vivant dans une maison intelligente. Il est équipé d'une montre intelligente, qui surveille en permanence son taux de glucose et ses activités quotidiennes. Le système de santé intelligent de l'hôpital recueille et traite les informations de santé de Bob afin de lui fournir les soins nécessaires en cas de besoin et de lui conseiller une meilleure alimentation.

### 3.4.2 Architecture de réseau sous-jacente

Les nouvelles architectures réseau ouvrent la voie au développement de l'informatique orientée service, permettant le déploiement d'architectures "*as a service*" et d'environnements virtualisés dans lesquels seules les instances de fonctions de réseau néces-




a Plan de Connaissance

b Plan de Sécurité et de Protection de la Vie Privée

FIGURE 3.4 – Architecture de sécurité et protection de la vie privée sensibles au contexte en tant que service (CASPaaS)

saires seront utilisées. Ils apportent une nouvelle philosophie basée sur les transformations effectuées dans les architectures réseau, essentiellement basées sur la virtualisation et la programmation du réseau. Ils peuvent ainsi supporter l'informatique orientée service, la programmation dynamique des réseaux par le biais du SDN, la NFV, l'*Edge Computing*, etc. Ces réseaux supportent la haute densité des dispositifs IoT et offrent davantage de débit et une latence plus faible. Ce qui est essentiel pour l'IoT et ses applications critiques telles que la sécurité routière et la e-santé.

Sur la base de ces technologies et de l'architecture de référence ITU-T pour l'IoT, notre architecture peut être mise en œuvre en tant que fonctions réseaux VNF, qui peut être au besoin instantanément déployée dans le réseau, indépendamment de la localisation de l'utilisateur. Cela garantira un niveau de sécurité et de protection de la vie privée optimal pour l'utilisateur, où qu'il se trouve. Ainsi, notre architecture exploitera les possibilités offertes par le paradigme SDN/NFV et l'*Edge Computing* pour la programmation du réseau et l'orchestration dynamique du service réseau à la périphérie du réseau [VLG+17]. La mise en œuvre de l'architecture proposée sera étudiée dans chapitre 6 de ce document.

La programmation du réseau offerte par la technologie SDN permet de diriger dyna-

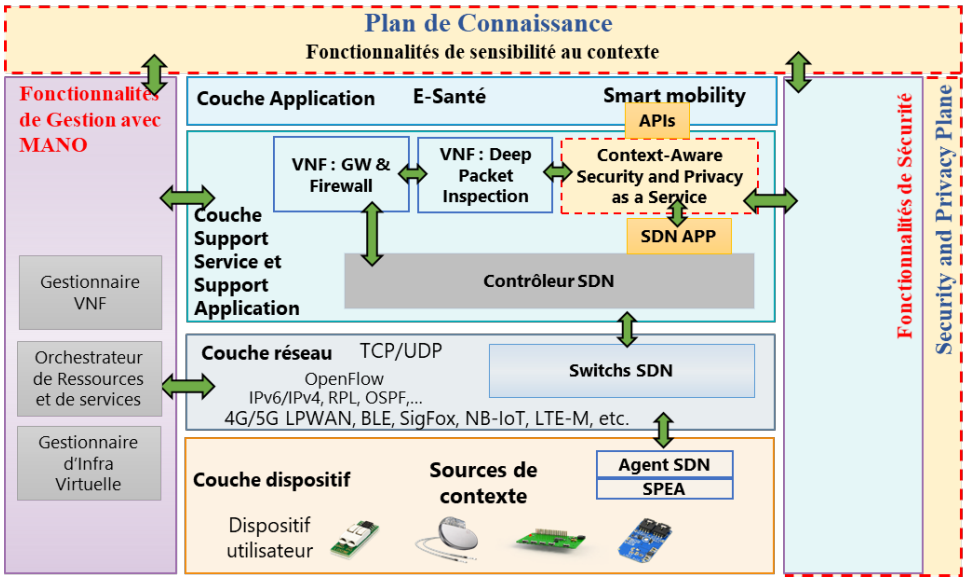



FIGURE 3.5 – Architecture de réseau sous-jacente de CASPaaS

miquement le trafic vers le service et permet aux dispositifs IoT d'appliquer les politiques de sécurité résultant du service [TYMR17]. Grâce à la technologie NFV, il sera possible d'orchestrer dynamiquement le déploiement du service au plus près de l'utilisateur via les possibilités offertes par ces réseaux [ASL+19]. Ces nouveaux paradigmes de réseaux s'adaptent à l'architecture de référence IoT de l'ITU-T. En effet, leur plan de gestion et d'orchestration peut étendre la couche de gestion de l'architecture de référence ITU-T pour l'IoT. Par exemple, le service sera disponible partout où se trouve Bob. Lorsque Bob est chez lui, l'instance de service sera placée sur le nœud de périphérie près de sa maison. Lorsque Bob est dans sa voiture, le service sera placé sur le nœud RSU (*Road Side Unit*) le plus proche du réseau véhiculaire. La figure 3.5 représente l'architecture UIT-T pour l'IoT incluant notre service avec la couche de gestion étendue pour les réseaux de prochaines générations.

### 3.4.3 Plan de Connaissance

Pour être efficace, la sécurité sensible au contexte doit avoir une excellente perception de l'environnement de l'utilisateur. Cela nécessite une bonne gestion de la connaissance du contexte. Ainsi, le PC, qui symbolise la séparation de la connaissance de la mise en œuvre des mécanismes de sécurité et de protection de la vie privée, est l'un des éléments clés de notre proposition. Son objectif principal est de fournir des contextes

spécifiques et pertinents pour le PSPVP (cf. Section 3.4.4). Sur la base du contexte fourni, le PSPVP proposé mettra en œuvre les mécanismes de sécurité et de protection de la vie privée les plus adaptés à la situation de l'utilisateur.

Le PC (figure 3.4a) est composé de modules nécessaires à la gestion du cycle de vie du contexte, c'est-à-dire l'acquisition, la modélisation, le raisonnement et la diffusion du contexte. Ainsi, le point de départ de la gestion du contexte est l'acquisition du contexte. Ensuite, le module d'acquisition de contexte (AC) reçoit les informations de contexte des sources de contexte de confiance. Nous désignons par source de contexte tout dispositif et capteur dans l'environnement de l'utilisateur qui collecte des informations contextuelles. Le module AC effectue un prétraitement (par exemple, les données brutes d'un capteur GPS doivent être mises dans un format qui représente l'emplacement géographique) et stocke les informations contextuelles prétraitées, également appelées contexte de bas niveau, dans la base d'informations contextuelles (BIC). Par exemple, Bob quitte sa maison et marche dans la rue. Dans ce cas, les sources de contexte disponibles sont : la montre connectée (contenant : pédomètre, capteur GPS) et le smartphone. Les informations contextuelles suivantes (contexte de bas niveau) sont envoyées au PC : date et heure, emplacement géographique de l'utilisateur, réseau de l'utilisateur et mouvement de l'utilisateur. Ces informations contextuelles sont ensuite prétraitées par le module AC et stockées dans le BIC. Les valeurs de ces informations contextuelles de bas niveau seront, dans cet exemple, comme suit : *latitude : 44.909356, longitude : -0.633280, Date : 2019-08-23 Heure : 13 :05 :19 Réseau : mobile_4G Adresse IP : 10.123.16.34 Vitesse : Lente*.

L'étape suivante de la détermination du contexte, après l'acquisition du contexte, est la modélisation du contexte. Cette étape de modélisation du contexte est réalisée par le module de modélisation du contexte (MC). En effet, il représente le contexte en termes d'attributs de contexte, de caractéristiques et d'attributs de qualité de contexte. Ensuite, la représentation obtenue est organisée en fonction du modèle de contexte choisi. Différents modèles de contexte existent : le modèle clé-valeur, le modèle basé sur le balisage, le modèle basé sur les objets et les rôles, le modèle basé sur la logique, le modèle basé sur l'ontologie et le modèle hybride [PZCG21]. Le choix d'un modèle dépend de la capacité du modèle à répondre aux exigences de la modélisation du contexte et du domaine d'application cible. Dans l'exemple considéré, un modèle clé-valeur est bien adapté en raison de sa simplicité et de sa flexibilité pour modéliser un tel contexte

```
{
  "date": [2019-08-23]
  "time" : [13:05:19]
  "location":[44.909356,-0.633280]
  "state": "in motion"
  "motion": "2 steps/second"
  "speed": "low"
  "network": "4G_LTE"
  "ip_address" : [10.123.16.34]
}
```

FIGURE 3.6 – Exemple de représentation d'un modèle clé-valeur

(informations contextuelles disponibles de petites tailles). Nous proposons de réaliser ces traitements de modélisation en collaboration avec le module de gestion de la qualité de contexte (QoC). Par exemple, après le traitement du contexte de bas niveau par le module MC (étape précédente), la représentation résultante dans un modèle clé-valeur est illustrée dans la figure 3.6. Le module de gestion de la QoC est l'une des améliorations de la gestion de la sensibilité au contexte que nous avons proposée dans le PC. En effet, il permet de résoudre les conflits dans la détermination du contexte. La QoC est caractérisée par un ensemble de paramètres : actualité, fiabilité, exhaustivité et importance. Tout d'abord, le module calcule ces paramètres de QoC pour mesurer la qualité du contexte de bas niveau reçu. Ensuite, les résultats de ces mesures seront interprétés pour déterminer l'existence de conflits. En fonction du type de conflit détecté, il applique un ensemble de politiques pour fournir un contexte de meilleure qualité. Par exemple, pour la détection de la position actuelle de l'utilisateur, une politique peut être basée sur la fraicheur du contexte, qui consiste à choisir la dernière valeur de la position GPS de l'utilisateur et à rejeter toutes les autres.

Après la modélisation du contexte, l'étape suivante de la gestion du cycle de vie du contexte est le raisonnement du contexte. Le raisonnement contextuel est le processus de déduction du contexte de haut niveau (le contexte) à partir de plusieurs informations contextuelles de bas niveau. La sortie du MC est utilisée par le module Raisonnement et Prédiction du Contexte (RPC) pour déterminer le contexte de haut niveau. En effet, il infère sur les informations contextuelles de bas niveau fournies par le MC en utilisant une technique de raisonnement pour déterminer le contexte de haut niveau. Dans le cas

de Bob, le contexte de haut niveau résultant sera : "*marche près de la maison*".

Il existe plusieurs techniques de raisonnement contextuel dans l'informatique contextuelle, notamment les règles, les techniques basées sur les ontologies, l'apprentissage automatique (supervisé ou non supervisé), la logique floue, etc. Une technique d'apprentissage supervisé sera utilisée par le module de raisonnement contextuel dans notre proposition. Ce choix s'explique par la bonne précision des algorithmes de cette technique. Le contexte de haut niveau déterminé est d'abord validé par le module de gestion de la qualité de contexte. Cette validation est effectuée par l'application de politiques de résolution de conflits (politique de qualité de contexte qui évalue l'actualité, la fiabilité, etc.). Ensuite, le contexte de haut niveau résultant est stocké dans la base de contextes (BC).

Enfin, la dernière étape de la gestion du cycle de vie du contexte est la distribution du contexte aux consommateurs de contexte. Avant la distribution du contexte, notre solution à travers le PC évalue le niveau de risque et les préférences de l'utilisateur associés au contexte. Ces opérations sont effectuées respectivement par le module Gestionnaire de l'Evaluation des Risques (GER) et le module Gestion des Préférences Utilisateur (GPU). Le contexte, le niveau de risque et les préférences de l'utilisateur seront directement distribués au PSPVP afin de prendre une décision de sécurité adaptée au contexte global de l'utilisateur. Dans notre architecture, le principal consommateur de contexte est le gestionnaire des politiques de sécurité contextuelle du PSPVP. Pour ce faire, dans le PC, nous proposons d'effectuer la distribution du contexte par le module Distributeur de Contexte (DC). Chaque fois qu'un nouveau contexte de "haut niveau" est disponible (dans BC), le module DC l'envoie au GER pour l'évaluation des risques.

Le module GER proposé calculera le niveau de risque d'un contexte donné sur la base du modèle de menace associé à ce contexte. Dans l'exemple considéré, lorsque Bob se trouve dans un jardin public avec ses amis, les dispositifs de Bob (smartphone, montre connectée) sont connectés au réseau Wifi du jardin public. Une fois que le DC a reçu le nouveau contexte de Bob, il l'envoie au GER pour l'évaluation des risques. Le GER évalue le risque du contexte donné en fonction de son modèle de menace (réseau non sécurisé, écoute clandestine, etc.), donc un risque élevé dans ce cas. Ensuite, le GER renvoie au DC le contexte de Bob avec le niveau de risque évalué.

Lorsque le DC reçoit le niveau de risque du contexte, il obtient les préférences de l'utilisateur correspondant à partir du module GPU. Ensuite, le DC envoie le contexte

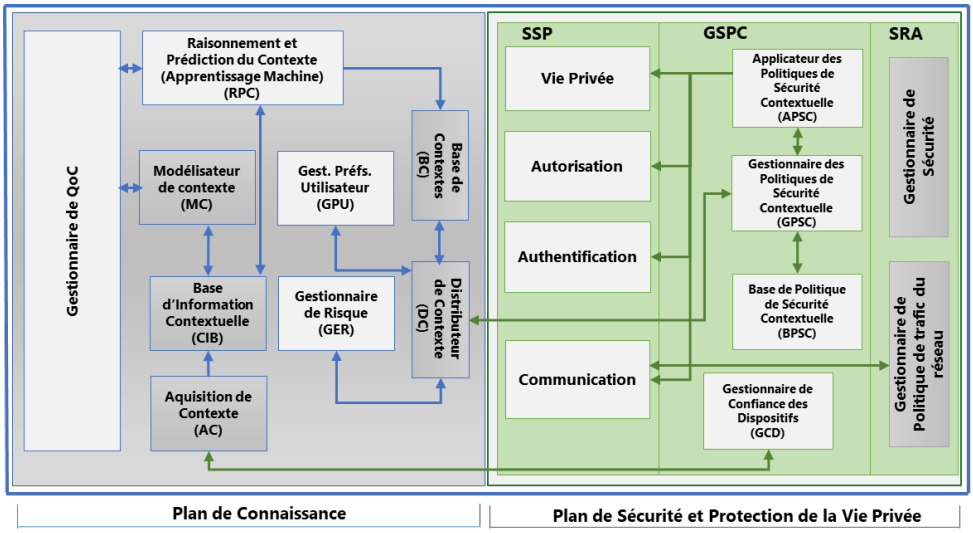



FIGURE 3.7 – Modules de l'architecture CASPaaS et leurs interactions

avec le niveau de risque associé et les préférences de l'utilisateur au Gestionnaire des Politiques de Sécurité Contextuelles (GPSC). Le PSPVP peut utiliser ce nouveau contexte et déployer les mécanismes de sécurité et de protection de la vie privée adaptés au niveau de sécurité. En fait, nous proposons de faire cette distribution du contexte et des informations connexes au gestionnaire des politiques de sécurité du contexte par un mécanisme de publication/souscription. Le GPSC souscrira à des sujets de contexte périodiques et événementiels (par exemple : changements de réseau d'utilisateur) auprès du module DC. Le module DC renverra ensuite le contexte résultant au module GPSC lorsque de nouveaux contextes seront disponibles. Ainsi, le PC fournit l'intelligence nécessaire à l'automatisation de l'architecture proposée. La figure 3.7 illustre les interactions entre les modules de l'architecture.

### 3.4.4 Plan de Sécurité et de Protection de la Vie Privée

Comme mentionné dans l'introduction, une mise en œuvre efficace des mécanismes de sécurité et de protection de la vie privée sensibles au contexte nécessite une séparation de l'intelligence et de l'application des décisions de sécurité et de protection de la vie privée. Le PC représente l'intelligence, ainsi, sur la base des connaissances fournies, les mécanismes de sécurité et de protection de la vie privée adaptés seront déployés.

Pour ce faire, notre architecture intègre un plan d'application des décisions appelé Plan de Sécurité et de Protection de la Vie Privée (PSPVP) (figure 3.4b). Ce plan répond

aux exigences fonctionnelles de la sécurité sensible au contexte identifiées et comprend des modules qui peuvent être déployés dynamiquement en fonction du contexte fourni par le PC. Nous avons proposé de diviser le PSPVP en trois composantes fonctionnelles : Services de Sécurité et protection de la vie Privée (SSP), de Gestion de la Sécurité et de la protection de la vie Privée Contextuelles (GSPC) et de Sécurité du Réseau et de l'Architecture (SRA). Cette division fonctionnelle permet de traiter les contextes de haut niveau, fournis par le PC, avant l'application de la décision de politique de sécurité et de protection de la vie privée sensibles au contexte. Cette division nous permet également de garantir la sécurité de l'architecture elle-même contre certaines attaques.

Les composantes Services de Sécurité et de protection de la vie Privée (SSP) et Gestion de la Sécurité et de la protection de la vie Privée Contextuelles (GSPC) constituent le cœur du PSPVP. En effet, la composante Gestion de la Sécurité et de la protection de la vie Privée Contextuelles comprend les modules chargés de la gestion des politiques de sécurité contextuelle et de la gestion de la sécurité de la connaissance du contexte. L'objectif de la composante GSPC est de prendre des décisions en matière de sécurité et de protection de la vie privée basées sur le contexte qui sera fourni par le PC. La composante SSP est composée de modules responsables de l'application des décisions de sécurité et de protection de la vie privée sensibles au contexte prises par la composante GSPC. Enfin, la composante SRA comprend les modules assurant la sécurité de l'architecture et du réseau.

Comme mentionné plus haut, l'architecture proposée sera en mesure de fournir une gestion sécurisée de la sensibilité au contexte. Pour ce faire, elle doit être capable de recueillir les informations contextuelles des sources de confiance. Ensuite, dans le GSPC, le rôle du module Gestion de la confiance des dispositifs (GCD) est d'assurer la sécurité de la gestion de la sensibilité au contexte grâce à la gestion de la confiance des sources de contexte. Un registre distribué et sécurisé, la Blockchain, peut être utilisé pour gérer la confiance de ces sources.

Les avantages de l'utilisation de la Blockchain pour assurer la confiance dans l'IoT et dans plusieurs autres domaines sont nombreux [AEM+18, FPS+19, MCK18, PSL+19]. Chaque source de contexte et les informations de contexte qu'elle est autorisée à collecter seront enregistrées dans la Blockchain. Le module GCD vérifiera la confiance de la source de contexte dans le registre avant tout traitement. Une autre possibilité de gestion de la confiance des dispositifs par le GCD est l'utilisation des techniques d'in-

telligence artificielle pour vérifier la fiabilité des informations contextuelles qu'ils transmettent et leurs comportements. La mise en œuvre effective d'un mécanisme du module GCD suivant cette dernière approche est détaillée dans la section 4.3 du chapitre 4.

Par ailleurs, chaque source de contexte doit envoyer les informations de contexte cryptées au GCD. En effet, la nécessité de sécuriser chaque information contextuelle avant de l'envoyer au système, quel que soit le contexte de l'utilisateur, permettra d'empêcher le système de traiter des informations contextuelles malveillantes. Seul le GCD sera en mesure de décrypter les informations de contexte transmises par la source du contexte. Une fois décryptées, le GCD transmet les informations au module d'acquisition du contexte PC. Un mécanisme d'échange sécurisé des informations contextuelles est proposé et détaillé dans la section 4.3.2.1 du chapitre 4 de ce document.

Un élément central de la sécurité sensible au contexte de notre proposition est la Gestion des Politiques de Sécurité Contextuelles (GPSC). Ce module est chargé de sélectionner la politique de sécurité contextuelle correspondant au contexte (par exemple, le contexte actuel de Bob), au niveau de risque et aux préférences de l'utilisateur fournies par le PC. Pour ce faire, lorsque le GPSC reçoit un contexte et des informations connexes, il obtient la politique correspondante de la Base de Politiques de Sécurité Contextuelle (BPSC) et l'envoie à l'Applicateur des Politiques de Sécurité Contextuelles (APSC). La politique de sécurité contextuelle décrit les mécanismes de sécurité et de protection de la vie privée à déployer dans un contexte spécifique.

Le rôle de l'APSC est d'utiliser la politique de sécurité fournie par le GPSC pour demander l'application des mécanismes de sécurité et de protection de la vie privée adaptés par les modules de la composante SSP : protection de la vie privée, authentification, contrôle d'accès (autorisation) et sécurité des communications. En d'autres termes, chaque fois qu'une politique de sécurité contextuelle (PSC) doit être mise en œuvre, l'APSC devra orchestrer la composition de services correspondant aux modules appropriés du composant SSP conformément à la politique fournie. Pour le dernier exemple, la politique de sécurité contextuelle dicte l'application des mécanismes suivants : authentification à deux facteurs, renouvellement des clés d'authentification des dispositifs et établissement d'une communication sécurisée avant tout transfert de données.

Après le traitement de la décision relative à la politique de sécurité contextuelle, la politique sélectionnée doit être appliquée par des mécanismes de sécurité et de protection de la vie privée sensibles au contexte. Ainsi, au niveau du SSP, nous proposons

les modules de protection de la vie privée, d'authentification, d'autorisation (contrôle d'accès) et de communication pour la mise en œuvre des mécanismes de sécurité et de protection de la vie privée sensibles au contexte. Nous proposons de dynamiquement composer ces modules en fonction du contexte par le module APSC. Par ailleurs, pour assurer la généricité de la solution et son indépendance vis-à-vis des applications IoT, notre architecture fournie des API (Application Programming Interface) aux développeurs de ces applications. En effet, grâce à l'approche "*as a service*", cette dernière permettra aux développeurs d'applications IoT d'exporter les tâches de sécurité et de protection de la vie privée de leurs applications (par exemple, l'authentification, l'autorisation, etc.) en appelant les API prévues à cet effet (par exemple, en utilisant REST : *REpresentational State Transfer*).

La protection de la vie privée sensible au contexte sera assurée par le module de protection de la vie privée. Ce dernier agira comme un assistant de protection de la vie privée. Il doit être capable d'analyser en permanence les données provenant des dispositifs de l'utilisateur et, en fonction du contexte, d'informer l'utilisateur s'il existe un risque avéré pour sa vie privée. Ce module met également en œuvre les règles fournies par le module d'application des politiques de sécurité contextuelle (anonymisation, occultation des données, etc.). Le module Authentification proposé est responsable de l'authentification en fonction du contexte des utilisateurs et des dispositifs IoT dans le cadre d'une application IoT. Ainsi, en fonction des règles fournies par l'applicateur des politiques de sécurité contextuelles, un type d'authentification est proposé à l'utilisateur : un facteur, double facteur ou encore triple facteur. Dans le cas d'un dispositif, en fonction du contexte, la clé de session peut être renouvelée.

Le service de contrôle d'accès sera assuré par le module de gestion d'autorisation. Le module de gestion des autorisations gérera le contrôle d'accès aux ressources en fonction du contexte. Il peut être possible à travers la Blockchain de définir et gérer les autorisations d'une entité de manière distribuée suivant le fonctionnement d'une application IoT et l'approche centrée sur l'utilisateur. En effet, les autorisations d'une entité doivent être représentées sous forme de jetons et inscrites dans un *smart contract* enregistré dans la Blockchain. Grâce au module GPU, l'utilisateur doit pouvoir modifier ou révoquer une autorisation à tout moment. Dans tous les cas, l'autorisation est mise à jour dynamiquement et mise en œuvre par le module. Un mécanisme de gestion des autorisations sensibles au contexte et s'appuyant sur la Blockchain est proposé dans le

chapitre 6 de ce document.

La sécurité des communications peut être nécessaire dans certains contextes, notamment pour les communications sur les réseaux non sécurisés. En effet, les communications sur des réseaux non sécurisés peuvent être facilement interceptées et leurs contenus (par exemple, les données sensibles de Bob) divulgués. Par conséquent, dans ces situations, les communications entre les dispositifs et les applications des utilisateurs doivent être sécurisées. Ainsi, le rôle du module de communication proposé est, selon le contexte, d'établir des communications sécurisées entre les dispositifs et les applications, en appliquant la politique de sécurité contextuelle (PSC) décidée par l'applicateur des politiques de sécurité contextuelles. Ceci peut être fait en implémentant la sécurité des messages (charge utile du message) de la couche application. En effet, plusieurs études ont prouvé l'efficacité de la sécurité de la couche application pour assurer des communications sécurisées sur des réseaux non sécurisés dans l'IoT [CRT17].

Cette approche de sécurisation des communications est également mise en œuvre dans le mécanisme de sécurisation des informations contextuelles proposé dans le chapitre suivant. Pour ce faire, selon un PSC, les messages seront chiffrés et signés grâce à des mécanismes cryptographiques asymétriques adaptées aux dispositifs IoT, limités en ressources. Supposons que le système de santé de l'hôpital ait besoin de lire le taux de glucose de Bob. Lors du dernier prélèvement, le contexte de Bob était à la maison. Maintenant, le nouveau contexte de Bob est au jardin public. Donc, le gestionnaire des politiques de sécurité contextuelles fournit une PSC spécifiant une communication sécurisée à l'applicateur des politiques de sécurité contextuelles. Ensuite, ce dernier appliquera les mécanismes de sécurité et de protection de la vie privée correspondant à ce contexte en composant les services suivants : sécurité des communications et protection de la vie privée. Le résultat est l'établissement de communications sécurisées entre la montre connectée de Bob et le système de santé de l'hôpital avant toute transmission de données. Après l'établissement de la communication sécurisée, le niveau de glucose de Bob est anonymisé/obfusqué selon la PSC fournie. Enfin, l'architecture devra être déployée en tant que service. Pour cela, elle doit être sécurisée afin d'éviter d'éventuelles attaques (par exemple déni de service). La partie sécurité du réseau proposé est constituée des modules : Gestion de la sécurité et Gestion des politiques du trafic réseau. Le gestionnaire de la sécurité a pour rôle d'assurer la sécurité de l'ensemble de l'architecture en mettant en place des mécanismes de filtration et d'inspection appro-

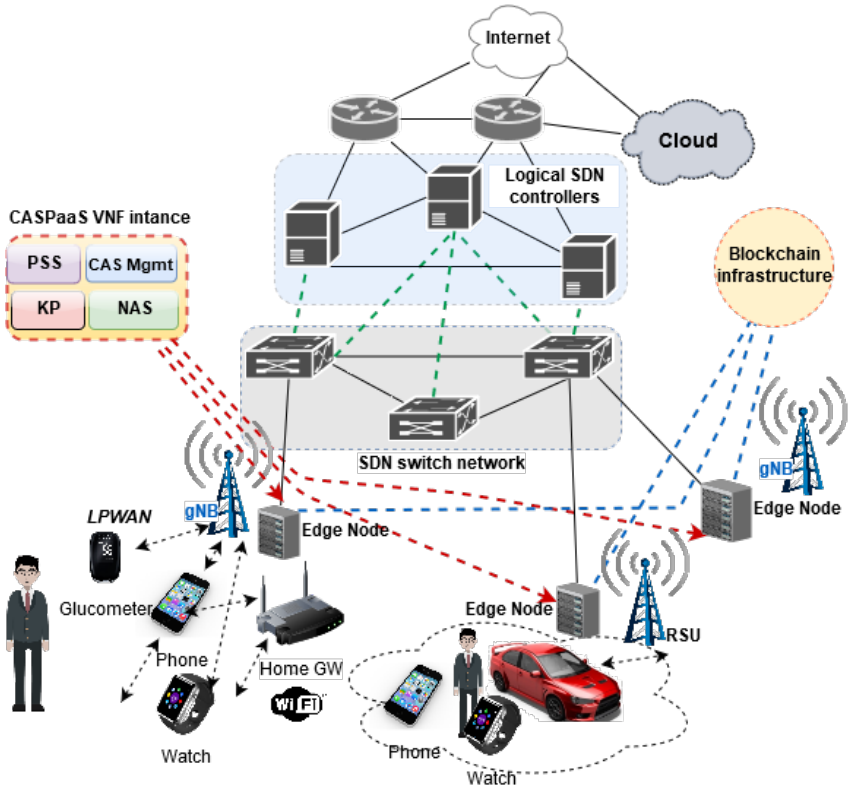



FIGURE 3.8 – Vue générale de l'architecture CASPaaS

fondis des paquets pour atténuer les attaques contre la disponibilité du service. Afin de répondre aux problèmes de mobilité des utilisateurs, d'hétérogénéité et d'évolutivité dus à la grande diversité des dispositifs IoT, nous proposons de transmettre les règles de sécurité contextuelles aux dispositifs IoT en utilisant les capacités offertes par la technologie SDN via un agent d'application de la politique de sécurité contextuelle par type de dispositif.

Le gestionnaire des politiques du trafic réseau est responsable de la transmission des règles aux dispositifs via la technologie SDN. Il est également chargé de gérer les communications réseau sous-jacentes. Il dicte au contrôleur SDN les chemins de trafic à suivre en fonction des résultats fournis par le gestionnaire de sécurité en cas d'attaque. Les dispositifs agiront alors comme des agents SDN, capables d'appliquer et de rediriger le trafic à la demande d'un contrôleur SDN. Le contrôleur SDN recevra des commandes des composants de mise en œuvre des mécanismes de l'architecture. La Figure 3.8 illustre la vue générale de l'architecture CASPaaS avec l'intégration de l'architecture de référence de l'UIT-T pour l'IoT.

## 3.5 Conclusion

Dans ce chapitre, nous avons décrit et comparé les travaux visant à fournir une sécurité adaptée au contexte dans l'IoT. Cela nous a permis d'identifier le besoin de disposer d'une architecture de sécurité et de protection de la vie privée sensibles au contexte, capable de garantir la sécurité et la protection de la vie privée d'un utilisateur de la manière la plus adaptée à la situation de ce dernier. Nous avons également identifié d'autres caractéristiques intéressantes comme l'évolutivité, la dynamicité, etc. Pour surmonter les défis identifiés, nous avons proposé une nouvelle architecture de sécurité sensible au contexte en tant que service. Contrairement aux propositions étudiées, cette architecture est composée de deux plans. L'objectif principal de cette division est de séparer l'intelligence de l'architecture (Plan de Connaissance - PC) de la mise en œuvre des mécanismes de sécurité et de protection de la vie privée (Plan de Sécurité et de Protection de la Vie Privée - PSPVP). Le PC est adapté au plan de gestion, voire au plan de contrôle et de données de l'architecture de référence ITU-T IoT. En effet, l'utilisation de l'apprentissage automatique pour l'amélioration de la qualité de contexte et de la déduction du contexte par le PC améliore la sensibilité au contexte, et donc la mise en œuvre efficace de mécanismes de sécurité et de protection de la vie privée adaptés au contexte.

La sécurité sensible au contexte permet de prendre en charge les changements de situation des utilisateurs des applications IoT des villes intelligentes. Cette architecture a été conçue selon l'approche "*as a service*", c'est-à-dire basée sur la composition de services. Ces services peuvent être composés en fonction du contexte afin d'implémenter uniquement les mécanismes de sécurité et de protection de la vie privée nécessaires. La motivation de cette approche est de mettre en œuvre l'approche centrée sur l'utilisateur à travers la prise en charge des exigences suivantes : la dynamicité, la flexibilité, la mobilité, l'évolutivité, la personnalisation, la composition en fonction du contexte des services et la prise en charge des applications génériques de l'IoT. Ensuite, l'architecture proposée peut être mise en œuvre en tant que fonction de réseau virtuelle (VNF). En effet, une VNF peut être déployée dynamiquement n'importe où dans les réseaux de nouvelle génération (dans les infrastructures *Edge Computing*) en utilisant les technologies SDN/NFV. Grâce au SDN, l'architecture pourra également gérer l'hétérogénéité des dispositifs IoT.

Pour assurer une mise en œuvre fiable de la sécurité et de la protection de la vie privée sensibles au contexte et mettre en œuvre l'approche centrée sur l'utilisateur (cf. section 2.3.3), une gestion sécurisée de la sensibilité au contexte dans un environnement de confiance est nécessaire. Dans le chapitre suivant, nous introduisons une solution de gestion sécurisée de la sensibilité au contexte dans un environnement de confiance pour l'IoT basée sur l'intelligence artificielle.

# Chapitre 4

# Gestion sécurisée et fiable de la sensibilité au contexte dans un environnement IoT de confiance : Application à la Smart City

## 4.1 Introduction

Une architecture de sécurité et de protection de la vie privée sensibles au contexte et prenant en charge l'ensemble des services de sécurité et de protection de la vie privée a été proposée dans le chapitre 3. Les dispositifs de l'utilisateur envoient les informations contextuelles vers le gestionnaire de contexte (position GPS, moment de la journée, vitesse de mouvement, etc.). Le gestionnaire de contexte modélise, fusionne ces informations pour déterminer un contexte : en approche de la maison, au bureau, etc. Le contexte déterminé est par la suite utilisé par le gestionnaire de sécurité contextuelle pour sélectionner et déployer les mécanismes de sécurité et de protection de la vie privée correspondant à ce contexte. Par exemple, il sera demandé à l'utilisateur de procéder à une authentification forte (deux ou trois facteurs).

Cependant, un système de sécurité et de protection de la vie privée sensibles au contexte est exposé à plusieurs types d'attaques. Premièrement, il est possible pour un adversaire de surveiller avec aisance un tel système, d'intercepter les informations contextuelles, de les modifier et ainsi de tromper la perception du système. En effet, dans la plupart des cas, les systèmes de gestion de la sensibilité au contexte utilisent des informations contextuelles non sécurisées [TP18]. Deuxièmement, les adversaires peuvent frauduleusement introduire des dispositifs malveillants dans le système. Troisièmement, il est possible que les dispositifs légitimes de l'utilisateur aient des capteurs défectueux. Cela a pour résultat la prise de décisions d'adaptation dynamique basée sur

des informations contextuelles imprécises ou complètement fausses. Par exemple, l'objectif recherché par l'attaquant peut être de fausser le déploiement de mécanismes de sécurité pour accéder à l'application de contrôle de la maison de l'utilisateur sans avoir besoin de s'authentifier, et ainsi pouvoir s'introduire dans la maison.

La sécurité des échanges des informations contextuelles est également très importante dans la gestion de la sensibilité au contexte. Elle permet de pallier les attaques visant la communication de ces informations. Dans ce sens, Zuo et al. [ZLG+20] ont étudié et évalué les problèmes de sécurité de l'information dans l'IoT. Ils ont alors proposé un cadre unifié pour l'évaluation de la sécurité des informations dans les systèmes IoT. Chen et al. [CWX+20] ont proposé une nouvelle technique pour garantir un contrôle d'accès sécurisé, qui préserve l'authenticité et l'intégrité des données dans l'IoT. Le Nguyen et al. [LNLLE+20] ont proposé une technique basée sur la blockchain pour le partage sécurisé et fiable des données IoT.

La gestion de la confiance est cruciale pour un système de sécurité et de protection de la vie privée sensibles au contexte. Elle permet de détecter les dispositifs qui ont un comportement malveillant. Ainsi, elle permet de prévenir les attaques telles que le clonage de dispositifs, l'usurpation et la forge d'information contextuelle [SSHY21]. La vérification minutieuse et approfondie de la fiabilité des informations contextuelles peut permettre au système de gestion de la sensibilité au contexte de ne pas traiter les informations contextuelles imprécises et/ou malveillantes. Ainsi, le système peut prendre des décisions d'adaptation fiables en utilisant des informations contextuelles sûres et provenant de dispositifs sûrs et sécurisés. Par conséquent, il devient nécessaire de proposer et de mettre en œuvre une gestion sécurisée et fiable de la sensibilité au contexte en utilisant des informations contextuelles sûres et provenant de dispositifs sûrs et sécurisés.

C'est pourquoi nous proposons dans ce chapitre un système de gestion sécurisée de la sensibilité au contexte dans un environnement de confiance. Le système proposé comprend un mécanisme permettant la transmission sécurisée des informations contextuelles et un mécanisme de gestion de la confiance basée sur la réputation. Il assure l'intégrité, la confidentialité et la protection contre le rejeu des informations contextuelles. Il pallie aux attaques d'usurpation, de modification et d'écoute clandestine. Il assure également la protection de la vie privée. Il permet la détection des dispositifs au comportement suspicieux et malveillant. En faisant appel à l'intelligence artificielle, il

permet la détection des mauvaises informations contextuelles. Aussi, ce système résiste bien aux attaques de bourrage de scores de confiance, à l'usurpation et à la forge d'information contextuelle. Il a, de plus, l'avantage d'être léger et adapté à un environnement de sécurité et de protection de la vie privée sensibles au contexte dans les applications IoT de la smart city.

Le système proposé présente également l'avantage d'être générique et de pouvoir ainsi l'intégrer facilement dans tout autre système sensible au contexte. Ainsi, notre système pourra permettre aux différents systèmes de sécurité et de protection de la vie privée sensibles au contexte pour les applications IoT susmentionnées de prendre des décisions d'adaptation fiables. Dans cette optique, il peut être intégré de l'architecture de sécurité et de protection de la vie privée sensibles au contexte pour l'IoT (CASPaaS) en tant que module de gestion de la confiance des dispositifs (GCD) (cf. Section 3.4.4).

Dans la suite de ce chapitre, nous introduisons la conception du système proposé et nous évaluons ses performances. À travers cette évaluation, nous démontrons que ce système présente un réel avantage en termes de performances dans un système de sécurité et de protection de la vie privée sensibles au contexte et qu'il protège ce système contre les problèmes énoncés. Les contributions de ce chapitre sont résumées comme suit :

— SETUCOM : un système de gestion intégrée et sécurisée de la sensibilité au contexte dans un environnement IoT de confiance ;

— Un mécanisme d'évaluation de la fiabilité des informations contextuelles basé sur les réseaux bayésiens qui associe les informations contextuelles au profil de l'utilisateur ;

— Un mécanisme d'évaluation du comportement des sources de contexte basé sur la logique floue qui calcule les statistiques liées aux sources de contexte (états anciens et actuels) pour déterminer leur comportement (bon, douteux ou malveillant).

La suite de ce chapitre est organisée comme suit. La section 4.2 compare les solutions existantes pour la gestion sécurisée de la sensibilité au contexte dans un environnement de confiance. La section 4.3 décrit le système proposé et la section 4.4 présente l'évaluation des performances du dit système. Enfin, la section 4.5 conclut le chapitre.

## 4.2 Travaux connexes

La sécurité et la protection de la vie privée sensibles au contexte dans les applications IoT de la ville intelligente ont fait l'objet de plusieurs travaux. Dans [TP18], les auteurs ont effectué une revue des problèmes de sécurité des systèmes sensibles au contexte. Mahalle et al. [MD20], ont décrit les exigences en matière de sécurité pour un système sensible au contexte. Cependant, plusieurs travaux ayant traité la problématique de la mise en œuvre de mécanismes de sécurité et/ou de protection de la vie privée sensibles au contexte dans certaines applications de l'IoT n'ont pas considéré la sécurité, la confiance des sources de contexte et la fiabilité des informations contextuelles [AKM17, BPGB18, KW15, dTAH18b, NSB+15].

Cette section étudie les travaux de recherche traitant de la sécurité et de la confiance des systèmes de gestion de la sensibilité au contexte dans l'IoT.

### 4.2.1 Sécurité de l'échange d'information contextuelle

La mise en œuvre de la sécurité de l'échange des informations contextuelles est une tâche importante dans la gestion de la sensibilité au contexte. En effet, des adversaires peuvent surveiller le système, tenter de reproduire les contextes et de tromper la perception du système. Ramos et al. [RBS15] ont étendu les fonctionnalités de sécurité de l'architecture de référence IoT-A EU en ajoutant des fonctionnalités de sécurité sensible au contexte. Cette architecture intègre un mécanisme permettant d'assurer la sécurité des informations contextuelles. Pour ce faire, les auteurs ont proposé l'utilisation du chiffrement CP-ABE [BSW07]. L'avantage de ce mécanisme est que les données chiffrées ne sont accessibles qu'aux entités autorisées. Cependant, CP-ABE est réputé pour sa complexité et son coût élevé en termes de calcul et de consommation d'énergie. En outre, il n'est pas flexible dans la révocation des autorisations et n'est pas évolutif [YCT15].

Ahamed et al. [AK19] ont abordé le problème de l'authentification et du contrôle d'accès sensibles au contexte dans les systèmes IoT pour la santé. Ils ont proposé un nouveau mécanisme appelé ECCAPAC (*Enhanced Context-aware Capability based Access Control*). Le mécanisme proposé applique le contrôle d'accès en utilisant une étiquette de capacité et des informations contextuelles. Cependant, la sécurité de l'échange d'informations contextuelles n'est pas prise en compte dans ce travail.

Das et al. [DNS+16] ont proposé un système de contrôle d'accès sensible au contexte pour les systèmes cyber-physiques, incluant l'IoT. Le système proposé est basé sur un contrôle d'accès sensible au contexte basé sur les attributs. Dans ce système, l'accès est autorisé ou refusé en fonction des politiques de sécurité sensibles au contexte préalablement définies dans la base de connaissances des politiques (*Policy KB*). Cependant, la sécurité des informations contextuelles fournies par les capteurs n'a pas été prise en compte dans ce travail.

Dans [AAOW18], Alagar et al. ont proposé un système de contrôle d'accès basé sur les rôles et sensible au contexte pour un système d'e-santé d'un hôpital. Dans cette solution, les auteurs proposent la mise en œuvre de la sécurité de la transmission des informations. Pour ce faire, ils proposent d'utiliser les mécanismes prévus à cet effet dans la norme IEEE 802.15.6. En effet, cette norme prévoit la mise en œuvre des communications intra-réseau corporel WBAN authentifiées et confidentielles. Cependant, la portée d'un réseau WBAN est limitée au corps du porteur. Autrement dit, la sécurité des informations n'est pas assurée dans les communications extra-WBAN (entre le réseau WBAN et l'extérieur).

Chouhan et al. [CMS18] ont fait ressortir les problèmes de sécurité dans les environnements IoT et ont proposé d'utiliser le concept d'évaluation de la situation pour assurer la sécurité des applications IoT. Ce concept est basé sur la détection d'événements qui peuvent aider à évaluer la situation et à proposer des mécanismes de sécurité appropriés. Cependant, la sécurité des données contextuelles permettant l'évaluation de la situation de l'utilisateur et la détection d'événements n'a pas été traitée. La confiance des dispositifs fournissant ces données n'a pas été prise en compte non plus.

Ashibani et al. [AKM17] ont proposé un service d'authentification sensible au contexte pour les applications domestiques intelligentes. Le principal avantage du système proposé est qu'il permet de renforcer l'authentification et le contrôle d'accès de l'utilisateur en les rendant adaptatifs au contexte de l'utilisateur. Cependant, les informations contextuelles recueillies ne sont pas sécurisées. De Matos et al. [dTAH18b] ont proposé un module de sécurité sensible au contexte pour une architecture de partage de contexte centrée sur l'*Edge Computing*. Cette solution tire avantage de l'intégration de l'infrastructure *Edge Computing*. En effet, elle permet une faible latence dans la transmission des informations contextuelles et prend en charge la mobilité. Néanmoins, la sécurité des échanges d'informations contextuelles n'est pas considérée dans cette ar-

chitecture.

Cependant, la sécurisation des communications dans les environnements IoT a fait l'objet de plusieurs travaux de recherche. Dans OSCAR [VTR+15], les auteurs ont proposé un mécanisme permettant l'envoi et la réception de paquets de données CoAP [SHB14] chiffrés sur un réseau non sécurisé. OSCAR adresse les limites du protocole DTLS (Datagram Transport Layer Security) en chiffrant la charge utile de la couche application. Dans [AAC+18], les auteurs se sont basés sur OSCAR pour proposer IoTChain. A la différence d'OSCAR, la confiance est décentralisée et gérée par la blockchain. Ce système a pour avantage de ne pas avoir une racine de confiance centralisée. Il permet également de pallier les limites du protocole DTLS. Cependant, ces solutions (OSCAR et IoTChain) mettent en œuvre plusieurs entités, notamment des serveurs de clés et des serveurs mandataires pour la blockchain. Les serveurs de clés et mandataires alourdissent ces solutions pour la sécurisation de l'échange d'information contextuelle. Par conséquent, elles ne sont pas adaptées à un environnement de sécurité et de protection de la vie privée sensibles au contexte dans l'IoT. De plus, d'une manière générale, le protocole MQTT est préféré au protocole CoAP dans la collecte d'information contextuelle. Cela s'explique principalement pour les performances de MQTT qui sont meilleures que celles de CoAP pour la transmission des mêmes charges utiles [ANRH15].

Malina et al. [MSD+19], ont proposé une architecture de sécurité pour le protocole MQTT. La solution proposée permet de disposer de trois couches de sécurité pour les communications sécurisées sous MQTT. La solution a été implémentée et évaluée. La couche de sécurité la plus robuste nécessite un tiers de confiance. Cependant, un tiers de confiance ajouterait une surcouche qui alourdirait l'architecture de sécurité et de protection de la vie privée sensibles au contexte.

### 4.2.2 Gestion de la confiance

La gestion de la confiance des sources de contexte et le contrôle de la fiabilité des informations de contexte sont des éléments indispensables au bon fonctionnement d'un système sensible au contexte. Arfaoui et al. [ACK+19] ont étudié le problème de la gestion des autorisations dans l'IoT. Ils ont proposé un mécanisme de contrôle d'accès sensible au contexte, nommé Context-Aware Attribute-Based Access Control (CAABAC). Le mécanisme proposé utilise les informations de contexte et les attributs des utilisateurs pour définir les règles de contrôle d'accès. Cependant, dans ce travail, les

auteurs n'ont pas abordé la fiabilité des informations contextuelles et la confiance des sources contextuelles. Par conséquent, le mécanisme peut recevoir de fausses informations contextuelles et ainsi prendre de mauvaises décisions de contrôle d'accès. Il peut également utiliser les informations contextuelles provenant des dispositifs malveillants.

Ramos et al. [RBS15] ont proposé de dynamiquement évaluer la confiance des dispositifs basée sur la réputation en analysant leurs différentes interactions avec le gestionnaire de contexte et les autres dispositifs (information contextuelle, précision, débit, etc.). Dans cette solution, le dispositif analyse ses interactions avec ses voisins et leur attribue un score de confiance (récompense négative ou positive). Cependant, les algorithmes utilisés dans ces travaux n'ont pas été spécifiés.

Par ailleurs, plusieurs recherches ont été effectuées sur la gestion de la confiance dans les applications de la ville intelligente. Chen et al. [CTC+19] ont proposé une architecture de confiance pour l'IoT incluant un protocole d'autorisation inter-couches dans un réseau SDN et une couche de gestion de la réputation. Les auteurs ont décrit deux mécanismes qu'ils ont nommés respectivement Behavior-based reputation evaluation scheme et Organization reputation evaluation scheme, pour l'évaluation du comportement des nœuds et de l'organisation. Toutefois, ces mécanismes de gestion de la réputation sont couplés à la gestion des autorisations. Cela constitue une surcharge supplémentaire non nécessaire pour la sécurité et la protection de la vie privée sensibles au contexte.

Assurer la sécurité et la confiance des réseaux informatiques et des environnements IoT, en particulier, en utilisant des techniques d'intelligence artificielle est un domaine de recherche prometteur. Dans ce contexte, plusieurs recherches avaient été faites pour la détection des comportements malveillants des dispositifs. Ces recherches se sont concentrées sur des techniques comme la logique floue, l'apprentissage automatique et profond, et les réseaux bayésiens. Ainsi, R.M. et al. [RMM+20] ont proposé un système de détection d'intrusion (IDS) efficace basé sur un réseau neuronal profond pour les systèmes d'e-santé. Selon les auteurs, le système proposé détecte et classifie efficacement les attaques. Rehman et al. [uRKI+21] ont proposé DIDDOS, une solution qui vise à détecter et à atténuer les attaques par déni de service distribué dans les réseaux informatiques. La solution proposée utilise l'unité récurrente de porte (Gate Recurrent Unit - GRU). L'évaluation a prouvé l'efficacité de la solution dans la détection des attaques DDoS. Par exemple, Shafiq et al. [STB+20] ont proposé une nouvelle technique basée

sur l'apprentissage automatique pour détecter le trafic malveillant des botnets IoT dans une ville intelligente. La technique proposée a été évaluée et son efficacité comparée à d'autres techniques de la littérature.

Dans le même contexte, Rehman et al. [RJJAM+20] ont proposé une nouvelle approche pour la détection des attaques de botnet dans les réseaux de véhicules connectés. Cette approche utilise des algorithmes d'apprentissage automatique sur le trafic réseau. Selon les résultats de l'évaluation, le schéma proposé a une bonne efficacité par rapport aux autres solutions proposées dans le même objectif. Néanmoins, ces propositions ([BZL+21, JBA+20, uRKI+21, RJJAM+20, RMM+20, STB+20]) ne sont pas adaptées aux environnements IoT à forte densité d'objets comme dans les villes intelligentes. De plus, ces propositions visent à protéger les environnements IoT contre des attaques réseau bien connues et spécifiques. Les données utilisées pour l'évaluation de la réputation des dispositifs ne sont pas disponibles dans un environnement de sécurité et de confidentialité sensible au contexte pour les villes intelligentes car il n'y a pas d'interaction entre les dispositifs dans un tel environnement. En effet, les échanges ont lieu entre le système et les sources de contexte. Par conséquent, des données supplémentaires sont nécessaires pour permettre une évaluation dynamique de la réputation dans cet environnement (profil de l'utilisateur, expériences, période d'exploitation, etc.).

Par ailleurs, la vérification minutieuse et approfondie de la fiabilité des informations est complémentaire à la gestion de la confiance. Les approches décrites dans [CTC+19, CGB16, RBS15] ne considèrent pas cette dimension dans la gestion de la confiance. Elle est importante car elle offre un levier supplémentaire permettant d'évaluer la crédibilité des sources de contexte elles-mêmes. Grâce à cela, le système peut facilement détecter les dispositifs malveillants.

### 4.2.3 Positionnement

Les travaux décrits ci-dessus ont proposé des solutions pour la sécurité de l'échange des informations contextuelles et la gestion de la confiance. Cependant, la mise en œuvre de la gestion sécurisée de la sensibilité au contexte dans un environnement de confiance dans l'IoT requiert l'association de la sécurité des informations contextuelles et de la confiance des sources de contexte.

Les solutions proposées pour assurer la sécurité des informations d'une manière générale ne sont pas adaptées à un environnement de sécurité et de protection de la vie

privée sensibles au contexte. En effet, les informations contextuelles doivent être sécurisées à la volée en utilisant des mécanismes de chiffrement robuste ne nécessitant pas le stockage de clés sur les sources. Cela permet de ne pas ajouter de surcharge supplémentaire dans la collecte et la transmission de l'information contextuelle. Ainsi, la solution que nous proposons est allégée et adaptée à l'environnement considéré. Elle permet de remédier aux problèmes identifiés et n'ajoute pas de surcharge supplémentaire. Elle permet également de protéger les informations contextuelles des attaques d'usurpation, d'écoute clandestine et de modification.

Bien que plusieurs travaux aient été effectués sur la gestion de la confiance dans l'IoT, fort est de constater que beaucoup reste à faire. En ce sens, la mise en œuvre de la sécurité sensible au contexte requiert la gestion de la confiance et de la fiabilité des informations contextuelles. Pour autant que nous sachions, un système d'échange sécurisé des informations contextuelles et de gestion de la confiance utilisant la fiabilité des informations contextuelles dans un environnement de sécurité sensible au contexte pour les applications IoT de la smart city n'a pas encore été proposé. Comme indiqué ci-dessus, un système de sécurité sensible au contexte dans une ville intelligente devra répondre aux exigences suivantes. Premièrement, il doit assurer la sécurité des échanges d'informations contextuelles. Pour ce faire, un système de sécurité de la communication léger peut être proposé. Ensuite, le système doit être capable d'évaluer la fiabilité des informations contextuelles et de rejeter les informations contextuelles non fiables. Enfin, il doit assurer la gestion de la confiance des sources de contexte, les informations contextuelles pouvant provenir de sources non fiables. Une comparaison des travaux étudiés est présentée dans le tableau 4.1.

## 4.3 Gestion sécurisée de la sensibilité au contexte dans un environnement de confiance

Dans cette section, nous introduisons le modèle de menace de l'environnement visé. Par la suite nous détaillons notre contribution et ses deux principaux mécanismes, à savoir, la sécurité des informations contextuelles et la gestion de la confiance.

Tableau 4.1 – Comparaison des principales solutions de sécurité et de confidentialité liées au contexte dans l'IoT

| Work | Sécurité I.C. | Fiabilité C.I. | Confiance | Protocole | IA | Evalué |
|---|---|---|---|---|---|---|
| [KW15] | ✗ | ✗ | ✗ | Not specified | ✗ | ✗ |
| [RBS15] | ✗ | ✗ | ✓ | CoAP, MQTT and XMPP | ✗ | ✗ |
| [DNS+16] | ✗ | ✗ | ✗ | Not specified | ✗ | ✗ |
| [AKM17] | ✗ | ✗ | ✗ | Not specified | ✗ | ✓ |
| [TTC17] | ✗ | ✗ | ✗ | Not specified | ✗ | ✗ |
| [AAOW18] | ✓ | ✗ | ✗ | Not specified | ✗ | ✗ |
| [CMS18] | ✗ | ✗ | ✗ | Not specified | ✗ | ✗ |
| [dTAH18b] | ✗ | ✗ | ✗ | Not specified | ✗ | ✗ |
| SETUCOM | ✓ | ✓ | ✓ | MQTT | ✓ | ✓ |

Légende : I.C. : Information Contextuelle, IA :Intelligence Artificielle

### 4.3.1 Modèle de menace

La mise en œuvre de la gestion sécurisée de la sensibilité au contexte dans un environnement de confiance requiert l'analyse préalable des différentes menaces pesant sur cet environnement. Ces menaces se situent à différents niveaux :

— Échange des informations contextuelles : Dans les projets de sécurité et de protection de la vie privée sensibles au contexte dans l'IoT [AKM17, BPGB18, KW15, dTAH18b, NSB+15], la transmission des informations contextuelles se fait sur des canaux de communication clairs. Par conséquent, les informations contextuelles sont vulnérables à l'attaque d'écoute clandestine (eavesdropping), d'autant plus facile à réaliser dans ces conditions. Les attaquants peuvent également intercepter les informations contextuelles et exploiter ces informations sans difficulté (comprendre le système, pister l'utilisateur à son insu, etc.). Il est également possible pour les attaquants de rejouer une information contextuelle capturée ou de la falsifier durant sa transmission.

— Dispositifs : Le système ne doit pas traiter l'information contextuelle provenant d'un dispositif compromis. Or, les menaces contre la confiance sont nombreuses. Elles comprennent le clonage, le vol, l'usurpation facilitée par l'utilisation de micrologiciel vulnérable et les mises à jour OTA (On The Air) via des canaux non sécurisés.

— Système de gestion de la confiance : Il existe également plusieurs menaces contre les systèmes de gestion de la confiance, notamment les attaques de manipulation du niveau de confiance (Ballot stuffing, Bad mouthing), d'identité multiple (Sybil attack), de comportement sélectif, etc. Les attaques ballot stuffing et bad mouthing ont pour objectif de, respectivement, augmenter et diminuer le score de confiance des nœuds. L'attaque de changement d'identité consiste pour un nœud ayant reçu un score de confiance bas après de mauvais comportements, de changer son identité afin que son score soit réinitialisé. Ainsi, le système de gestion de la confiance qui sera mis en œuvre devra résister à ces nombreuses attaques.

### 4.3.2 Cadre général

La solution proposée permet de protéger l'architecture CASPaaS (détaillée dans le Chapitre 3) contre les attaques d'usurpation d'identité, d'écoute clandestine, de modification (tampering) de données et de rejeu. Elle permet également le respect de la vie privée grâce à la sécurisation des échanges des informations contextuelles. Comme nous le verrons dans la section 4.4.2, la solution est également adaptée aux dispositifs contraints IoT de la smart city car elle a un faible impact sur la consommation énergétique et présente de meilleures performances comparées aux solutions existantes. La figure 4.1 illustre le principe de l'échange sécurisé des informations contextuelles proposé dans le cadre du placement de CASPaaS dans une infrastructure *Edge Computing*.

La collecte et la transmission sécurisées des informations contextuelles au système CASPaaS s'effectuent comme suit. Premièrement, les dispositifs IoT de l'utilisateur détectent les informations contextuelles. Deuxièmement, chaque dispositif chiffre les informations contextuelles collectées et les envoie au module de gestion de la confiance des dispositifs de CASPaaS (cf. Section 3.4.4).

Avant chaque échange, une clé authentifiée est établie entre le dispositif et le module de gestion de la confiance des dispositifs. L'authentification garantit l'identité des entités ayant pris part à l'établissement de la clé [dFW20]. Cette clé est utilisée par l'algorithme de chiffrement pour assurer la sécurité des informations contextuelles utilisées. L'établissement de la clé authentifiée et l'algorithme de chiffrement utilisé sont expliqués dans la section suivante. Le déploiement de CASPaaS en tant que service dans une infrastructure *Edge Computing* permet de prendre en charge, de manière sécurisée, la mobilité des utilisateurs. Le déploiement de notre architecture sera étudiée dans le

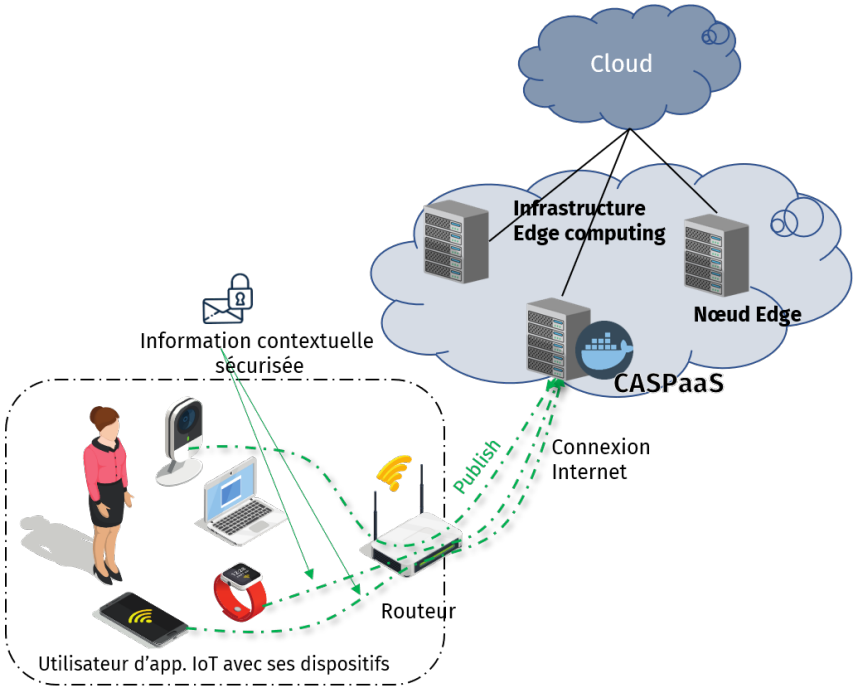



FIGURE 4.1 – Collecte et envoi sécurisés des informations contextuelles vers CASPaaS

chapitre 6 de ce document.

#### 4.3.2.1 Mise en œuvre de la sécurité de l'échange d'information contextuelle

Nous proposons de baser les communications sur le protocole MQTT car il est le mieux à même de répondre aux besoins de collecte des informations contextuelles. En effet, ce protocole est léger, robuste, supporte la QoS et facilite l'économie d'énergie [BBBG19]. Ainsi, grâce aux mécanismes Pub/Sub, les dispositifs IoT pourront remonter les informations contextuelles vers le courtier (*Broker*). Ce processus d'acquisition des informations contextuelles est représenté sur la figure 4.2. Le courtier est responsable de la distribution des messages des publieurs aux abonnés ayant souscrit à un sujet (*topic*). Par exemple, pour une application de maison intelligente le sujet pour les informations envoyées par un capteur de mouvement peut être *home/room/motion*.

Le modèle de communication est composé de trois parties : Les sources de contexte (publieurs), le gestionnaire de confiance des dispositifs (GCD, souscripteur) de CASPaaS et le courtier de l'application. Les sources de contexte sont des dispositifs contraints qui collectent, chiffrent et envoient les informations contextuelles sécurisées au GCD à des

fréquences régulières. Le GCD reçoit, déchiffre, évalue la réputation et transfère les informations contextuelles fiables au module d'acquisition du Plan de Connaissance (PC) de CASPaaS (cf. Sections 3.4.3 et 3.4.4). Spécifiquement, lorsque le GCD reçoit l'information, il a pour rôle de vérifier cette information avant de la transférer au module d'acquisition de contexte du PC. Pour ce faire, il évalue la réputation de la source et ne la transfère au module d'acquisition du PC que si elle est sûre et fiable. Les évaluations de la réputation des sources de contexte et de la fiabilité des informations contextuelles sont décrites dans la section 4.3.3.

Le protocole MQTT propose plusieurs méthodes pour assurer la sécurité des communications : SSL/TLS et le chiffrement de la charge utile (*Payload*). Le mécanisme de sécurité le plus implémenté pour la sécurité des communications dans le protocole MQTT est SSL/TLS. Cela peut s'expliquer par les propriétés de sécurité et de confiance fournies par SSL/TLS. Cependant, la gestion des certificats et des clés de session dans SSL/TLS augmente la complexité de calcul pour les dispositifs IoT. Par conséquent, elle entraîne une consommation d'énergie importante et un impact non négligeable sur la durée de vie des batteries de ces dispositifs. C'est pourquoi nous proposons dans ce travail la sécurisation de la charge utile qui, comme nous pourrons le voir dans la section 4.4.2, a un impact plus faible sur la consommation de ressources (énergie, processeur et mémoire vive). La charge utile comprend l'information contextuelle (exemple : latitude et longitude) et son sujet (exemple : contexte).

Les contributions scientifiques de ce travail sont les suivantes. Premièrement, nous proposons une nouvelle approche de sécurisation des communications basée sur MQTT. En effet, cette approche consiste à sécuriser la charge utile du message MQTT. L'avantage principal de cette approche est sa capacité à sécuriser les communications. De plus, elle est moins gourmande en ressources que la mise en œuvre de TLS (le protocole le plus utilisé pour sécuriser les communications MQTT). Deuxièmement, nous montrons que cette approche a un faible impact sur le temps de transfert des informations contextuelles. Elle a également un faible impact sur l'énergie et les performances de calcul des dispositifs contraints. Ainsi, il est possible de mettre en œuvre un échange sécurisé d'informations contextuelles sans impact important sur la consommation de ressources pour un système de sécurité et de protection de la vie privée sensibles au contexte.

Le système proposé permet de mettre en œuvre la sécurité de l'échange des informations contextuelles, indispensable pour se protéger des menaces présentées dans la

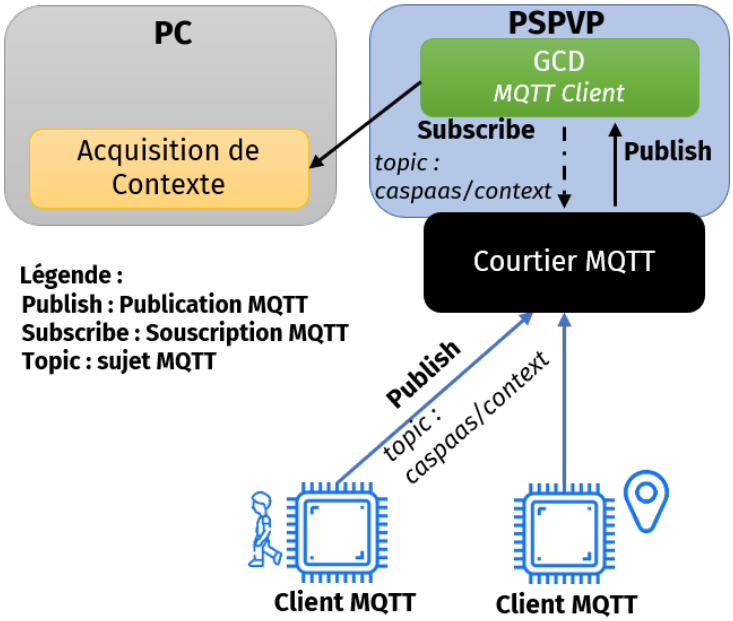



FIGURE 4.2 – Acquisition des informations contextuelles avec MQTT

section 4.3.1. Pour ce faire, nous avons besoin de mécanismes allégés en termes de consommation de ressource de calcul, de mémoire et d'énergie. Ce besoin est justifié par les contraintes liées aux sources de contexte. Ainsi, le système proposé permet d'assurer l'intégrité, la confidentialité, l'authentification des données et la protection contre le rejeu de la charge utile. Nous proposons également d'utiliser AES [DBN+01] pour le chiffrement des informations contextuelles. En effet, AES est un algorithme de chiffrement par bloc très utilisé pour assurer la sécurité des communications dans l'IoT. Cela s'explique par le fait que la plupart des dispositifs dispose de matériel d'accélération cryptographique spécifique à AES. Il offre également un chiffrement robuste avec une taille de clé réduite [JK16, SSMP17] permettant d'offrir le même niveau de sécurité que les algorithmes de chiffrement asymétriques. Il utilise des clés de 128, 192 et 256 bits. Chaque bloc de données chiffrées a une taille de 128 bits [DBN+01]. De plus, ces opérations de chiffrement nécessitent peu de ressources de la part des dispositifs.

Par ailleurs, AES a plusieurs modes d'opération : *Counter with CBC-MAC* (CCM), *Galois/Counter Mode* (GCM), *Electronic Codebook* (ECB), etc. [DR02]. Selon nos besoins, c'est-à-dire, un mécanisme cryptographique adapté aux dispositifs contraints et garantissant la confidentialité, l'intégrité et l'authentification, les modes d'opérations permettant de réaliser l'intégrité et l'authentification des données, en plus de la confi-

dentialité, sont CCM et GCM [Hou07]. Nous proposons que le système mette en œuvre le mode d'opération CCM. En effet, AES-CCM est un mode très utilisé car il présente de meilleures propriétés de sécurité par rapport à AES-GCM [Fer05, LXI17].

Cependant, AES étant un algorithme de chiffrement symétrique nécessitant l'utilisation de la même clé secrète pour le chiffrement et le déchiffrement des données, les deux parties doivent donc disposer de la même clé. L'échange de clé dans un réseau non sécurisé est confronté à plusieurs problèmes de sécurité. En effet, durant la phase d'échange de clés, il est possible qu'un adversaire puisse intercepter la clé. Par ailleurs, dans la plupart des systèmes, les clés sont stockées sur les dispositifs. Ces dispositifs peuvent être capturés ou clonés. Ainsi, pour se protéger de ces attaques, les clés utilisées pour chiffrer les informations contextuelles ne doivent pas être stockées sur ces derniers. Par conséquent, la mise en œuvre d'un algorithme permettant un échange sécurisé de clés à usage unique est nécessaire.

Pour résoudre les problèmes liés à l'échange de clés, nous proposons d'utiliser ECIES (*Elliptic Curve Integrated Encryption Scheme*) [CRT17]. C'est un système de cryptographie à clé publique authentifiée avec un certificat qui a pour objectif de générer une clé sécrète à usage unique par les deux parties d'une communication. Il combine la cryptographie à courbes elliptiques et la primitive de Diffie-Hellman [SNS16]. L'utilisation d'ECIES présente plusieurs avantages dans la gestion sécurisée de la sensibilité au contexte par rapport aux autres algorithmes de cryptographie à courbes elliptiques allégés pour l'IoT (par exemple Elliptic Curve Diffie-Hellman).

Premièrement, l'utilisation d'ECIES ne nécessite pas la mise en œuvre d'un tiers de confiance. En effet, un tiers de confiance dans cet environnement (sécurité et protection de la vie privée sensibles au contexte) ajouterait une couche supplémentaire. La collecte d'information contextuelle doit pouvoir se faire en temps réel ou quasiment en temps réel. Par conséquent, cela pourrait avoir des effets négatifs sur la collecte de ces informations. Deuxièmement, ECIES présente de bonnes performances et permet l'utilisation de clés de chiffrement à usage unique. Cela évite le stockage de clés sur les dispositifs, et élimine par conséquent la menace de réutilisation de clé lorsqu'un dispositif est capturé ou cloné. La clé sécrète générée est ensuite utilisée pour chiffrer les données en utilisant AES-CCM (*Confidentiality – Integrity – Availability* :CIA, Keyed-Hash Message Authentication Code : HMAC). Ce chiffrement des données permet d'assurer la confidentialité, l'intégrité et l'authentification des données.

Dans ce qui suit, nous décrivons la solution proposée qui comporte les trois étapes suivantes : authentification, initialisation des mécanismes de chiffrement et échange de données sécurisées.

#### 4.3.2.2 Authentification

Cette phase consiste à authentifier les différents clients, c'est-à-dire les sources de contexte, auprès du GCD. Le protocole MQTT prévoit trois modes d'authentification : identifiant du client, nom d'utilisateur/mot de passe ou certificat X.509. Nous proposons la génération et l'utilisation d'un identifiant unique par dispositif. Cet identifiant est issu du hachage cryptographique de la clé publique du dispositif et de son identifiant EUI-64. En effet, l'identifiant EUI-64 basé sur l'adresse MAC est unique et propre à chaque interface de communication d'un dispositif.

La clé publique est également unique et peut être générée à la demande. Grâce au hachage cryptographique robuste SHA256, l'identifiant unique généré sera très difficile à reproduire [GH04]. Une nouvelle clé publique sera périodiquement générée et l'identifiant recalculé. Cela permettra de prévenir les attaques pouvant viser la fonction de hachage et de deviner l'association clé publique/EUI-64. L'identifiant unique proposé est défini sous la forme suivante (équation 4.1) :

$$IDCS_i = Hash - SHA256(PkCS_i + EUI - 64CS_i) \tag{4.1}$$

Où SHA-256 est l'algorithme de hachage cryptographique, $IDCS_i$ est l'identifiant unique, $PkCS_i$ la clé publique du dispositif et $EUI - 64CSi$ l'identifiant EUI-64 du dispositif.

Les identifiants des sources de contexte autorisées sont préalablement enregistrés auprès du GCD. Si l'authentification réussit, les parties peuvent procéder à la phase d'initialisation. Si l'authentification échoue, le GCD le notifie au mécanisme de gestion de la confiance (voir Section 4.3.2.4) et redemande l'authentification. Après trois tentatives infructueuses, les demandes d'authentification sont rejetées durant une période de 30 secondes (période de rejet des authentifications). Si une nouvelle tentative échoue, un délai supplémentaire de 60 secondes est ajouté à la période de rejet des demandes d'authentification, et ainsi de suite. La période de rejet espace les demandes d'authentification et réduit le risque d'une attaque de déni de service.

#### 4.3.2.3 Initialisation

Dans le système proposé, l'initialisation consiste pour la source de contexte et le courtier à mettre en œuvre le processus de configuration d'ECIES. Cette configuration revient à générer les paramètres de sécurité de l'algorithme d'ECIES. Ces paramètres permettront de générer les couples de clés publique/privée qui serviront à l'agrément de clé. Dans ce qui suit, les clés privée et publique de la source de contexte i seront respectivement représentées par $SkCS_i$ et $PkCS_i$. De même, nous nous référerons aux clés privée et publique du GCD par, respectivement, $SkD$ et $PkD$.

Une fois l'authentification réussie, la source de contexte demande au GCD les paramètres de domaine permettant de spécifier une courbe elliptique et un point de référence : $E(Fp), G, p, n, a$ et $b$. Ces paramètres doivent être conformes aux recommandations édictées par le NIST (*National Institute of Standards and Technology*) [BCR+18] et l'ANSSI (*Agence Nationale de la Sécurité des Systèmes d'Information*) [Ref14] : taille ECC de 256 bits et nombre premier de 2048 bits. Pour ce faire, le courtier génère une clé privée $SkD$ et choisit un point $G$ de référence sur la courbe elliptique qu'il a défini suivant le niveau de sécurité. Ensuite, il détermine sa clé publique $PkD$ (équation 4.2).

$$PkD = SkD \cdot G, SkD \in [1, p] \tag{4.2}$$

Le GCD envoie $PkD$ et la courbe à la source de contexte. De la même manière, la source de contexte $CS_i$ génère $SkCS_i$et $PkCS_i$ et envoie $PkCS_i$ au GCD. $PkCS_i$ est défini par l'équation 4.3 :

$$PkCS_i = SkCS_i \cdot G, SkCS_i \in [1, p] \tag{4.3}$$

Où $G = (x_G,\ y_G)$ est le point de référence sur la courbe elliptique défini dans le corps fini $Fp$ [Bro]. Cet échange de clés publiques sécurisé basé sur l'échange de clé Diffie-Hellman est sécurisé et a lieu la première fois, c'est-à-dire, lors de l'établissement de l'échange sécurisé des informations contextuelles. Il a également lieu chaque fois qu'il nécessaire de réinitialiser les clés d'un dispositif.

#### 4.3.2.4 Échange de données sécurisées

Après la phase d'initialisation, l'envoi sécurisé de l'information contextuelle collectée prend place. Pour ce faire, la source de contexte $CS_i$ devra créer la clé sécrète partagée

$Ssk$qui servira à chiffrer la charge utile. Tout d'abord, cette $CS_i$ crée une valeur sécrète aléatoire partagée R, résultante de la multiplication scalaire prenant comme entrées la clé privée de $CS_i$ et la clé publique du courtier $PkD$ (équation 4.4).

$$R = SkCS_i \cdot PkD \tag{4.4}$$

Ensuite, la source de contexte $CS_i$ fournit la valeur sécrète aléatoire partagée $R$comme paramètre d'entrée de la fonction de dérivation de clé (*Key Derivation Function* - KDF). La fonction de dérivation de clé détermine la clé sécrète partagée $Ssk$et la clé de calcul du code d'authentification du message $K_MAC$. Nous proposons d'utiliser la fonction de dérivation de clé basée sur SHA-256. En effet, SHA-256 réduit fortement le risque de succès des attaques par force brute sur les clés générées. Cette fonction produit en sortie la concaténation de Ssk, la clé sécrète partagée, et de $K_MAC$, la clé du MAC, chacune ayant une taille de 128 bits. Pour ce faire, la $CS_i$ utilisera HMAC-SHA256, très sûre et très difficile à "craquer".

Nous proposons également d'horodater la charge utile pour pallier les attaques contre le rejeu. Nous supposons que chaque $CS_i$ dispose d'une horloge temps réel qui lui permet d'horodater les charges utiles de manière unique. Nous supposons également qu'à chaque démarrage, les $CS_i$ synchronisent leurs horloges avec le temps Internet. Le mécanisme anti-rejeu consiste à utiliser les horodates comme des nombres uniques non reproductibles générés au fur et à mesure des communications. La vérification suit le principe suivant. L'authentification de la $CS_i$ marque le début des échanges des informations contextuelles. Le GCD utilise l'horodate de l'authentification de $CS_i$ comme référence temporelle et initie une fenêtre temporelle glissante pour la $CS_i$ d'une durée de 300 secondes avec une temporisation (timeout) de 30 secondes. Cette temporisation, paramétrable selon les exigences de l'application à sécuriser, est utilisée par le GCD pour maintenir le glissement de la fenêtre temporelle ou l'arrêter si la $CS_i$ ne réalise aucune activité durant "ce temps". Dans ce dernier cas, la $CS_i$ devra s'authentifier pour effectuer une nouvelle opération.

La fenêtre temporelle glissante de 300 secondes peut être expliquée par la fréquence d'actualisation du contexte, une des caractéristiques majeures de la sécurité et de la protection de la vie privée sensibles au contexte. En effet, le contexte de l'utilisateur peut changer fréquemment, nécessitant une fréquence d'actualisation élevée. Avec une telle fréquence d'actualisation, l'authentification continue peut réduire l'autonomie énergé-

tique des dispositifs. Cette fenêtre glissante peut réduire la consommation d'énergie liée à l'authentification fréquente de la source de contexte. Ainsi, à chaque opération publish de la $CS_i$, le GCD valide l'horodate de la charge utile si elle respecte les conditions suivantes :

— L'horodate de la charge utile est inclus dans la fenêtre temporelle. Si ce n'est pas le cas, alors la charge utile est rejetée.

— S'il s'agit de la première opération publish après l'authentification, l'horodate ne doit pas être "supérieur" à celui de l'authentification de plus d'une demi-seconde. Sinon la charge utile sera rejetée car elle sera considérée comme "trop veille". Le délai d'une demi-seconde, soit 500 millisecondes, se justifie par la tolérance aux latences pouvant être engendrées par les perturbations des réseaux d'accès utilisés.

Lorsque le GCD valide une charge utile, il met à jour la fenêtre temporelle de la $CS_i$ en prenant la dernière horodate validée comme borne inférieure et rallonge la borne supérieure pour avoir les 300 secondes. Lorsque la $CS_i$ n'effectue aucune opération, après l'écoulement de la temporisation, elle doit s'authentifier et une nouvelle fenêtre temporelle doit être initiée. Ainsi, le GCD maintient une fenêtre temporelle glissante des horodates déjà validées par la $CS_i$ et rejette toutes les charges utiles qui ont une horodate déjà validées et ou en dehors de la fenêtre.

La source de contexte $CS_i$ utilise la clé secrète partagée $Ssk$pour chiffrer la charge utile horodatée en utilisant AES-CCM de 128 bits. Le résultat de cette opération est un message chiffré noté $M_ENC$. À partir de ce $M_ENC$ et de la clé $K_MAC$, la $CS_i$ utilise la fonction de HMAC-SHA256 pour calculer un *tag* MAC. Enfin, cette $CS_i$ envoie le couple ($M_ENC$, *tag*) au courtier. La figure 4.3 illustre la structure du paquet de données MQTT formé par une $CS_i$.

Lorsque le GCD reçoit la charge utile chiffrée, il extrait le couple $M_ENC$ et *tag* et fait le processus inverse en utilisant la clé publique de la $CS_i$. Pour ce faire, il recalcule la clé secrète partagée en utilisant la fonction de dérivation de clés et les paramètres précédemment établis avec la $CS_i$. Ainsi, il calcule un *tag* et le compare au tag envoyé par la $CS_i$. Si *tag* et tag sont différents, il abandonne le processus. Si *tag* et *tag* sont égaux, alors il procède au déchiffrement de $M_ENC$. Il vérifie l'horodatage de la charge utile pour en vérifier la validité. Si la vérification de l'horodatage échoue, il rejette le

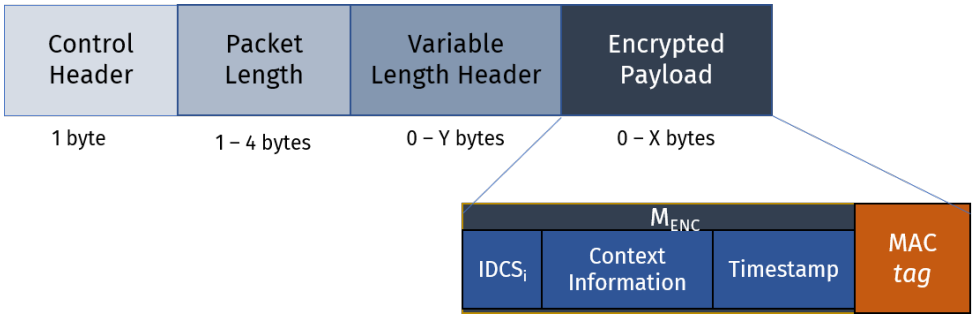



FIGURE 4.3 – Paquet MQTT avec information contextuelle sécurisée

paquet. L'opération PUBLISH est présentée dans l'équation 4.5.

$$PUBLISH(ECIES\_Key\_Exchange(G_i), Encrypt\_AES{-}CCM(CI, Ssk, Tp, ts, IDCS_i)) \tag{4.5}$$

Où $CI$ : information contextuelle, $Ssk$ : la clé secrète partagée, $Tp$ : sujet, $ts$ : horodatage

Une fois le message validé, il vérifie l'indice de confiance (*Trust Index* – TI) de la source de contexte grâce au mécanisme de gestion de la confiance. Si la source est sûre, il procède à la vérification de la fiabilité de l'information contextuelle avec le mécanisme de gestion de la fiabilité des informations contextuelles. En fonction du résultat de cette évaluation, l'information contextuelle reçue est transférée au module d'acquisition du contexte ou rejetée (Voir Section 4.3.3). Un résumé de l'opération d'envoi sécurisé d'une information contextuelle par une source de contexte est présenté dans la figure 4.4.

### 4.3.3 Mécanisme de gestion de la confiance des sources de contexte

Dans cette section, nous présentons le mécanisme de gestion de la confiance des sources de contexte. Ce mécanisme utilise la fiabilité des informations contextuelles et les comportements des sources de contexte pour évaluer la réputation de ces dernières. Grâce à l'évaluation de la fiabilité des informations contextuelles et du comportement des sources de contexte, le système peut détecter les fausses informations contextuelles.

#### 4.3.3.1 Généralités

La gestion de la confiance des sources consiste à définir comment établir, maintenir ou révoquer une relation de confiance avec un dispositif. Le but recherché est de

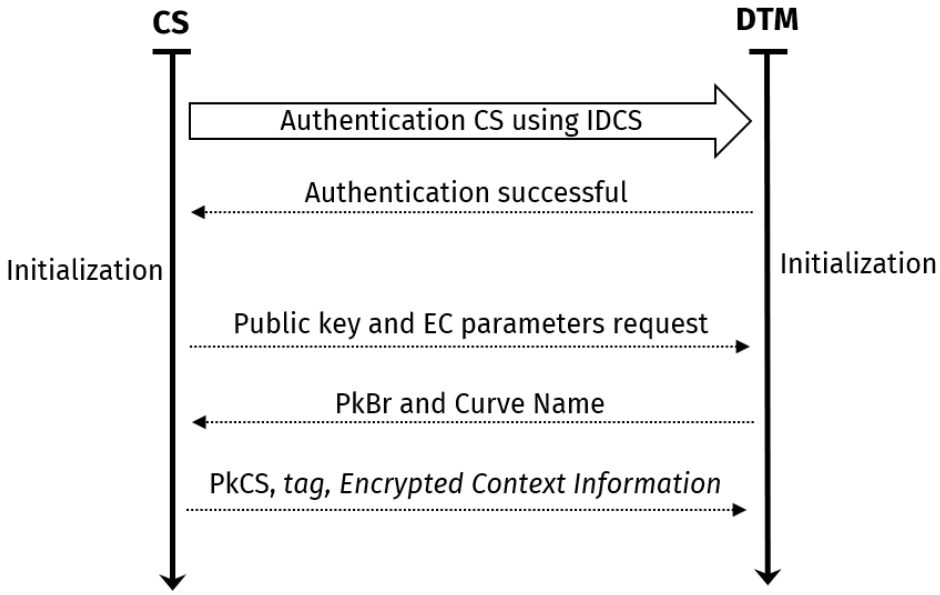



FIGURE 4.4 – Déroulement de l'échange sécurisé d'information contextuelle

permettre au système de gestion de la sensibilité au contexte de ne traiter que les informations contextuelles fiables et provenant de sources de contexte sûres. Cela protège l'architecture CASPaaS contre la prise de décision d'adaptation erronée ou inadaptée.

Il existe plusieurs modèles de gestion des relations de confiance dans l'IoT. Parmi ces modèles, nous pouvons citer la négociation, l'évaluation de la réputation et la décision par politique prédéfinie [YZV14]. Le choix d'un modèle est basé, d'une part, sur le modèle d'interaction des différents nœuds et, d'autre part, sur les données provenant de ces interactions. Selon ces éléments, l'approche basée sur le modèle de la réputation est la plus appropriée pour un environnement de sécurité et de protection de la vie privée sensibles au contexte dans l'IoT. En effet, dans cet environnement, nous pouvons disposer des données (fiabilité des informations contextuelles et comportement des dispositifs dans notre cas) permettant d'évaluer les expériences passées des sources de contexte. La réputation peut être considérée comme un moyen permettant de faire confiance ou pas sur la base des expériences et/ou observations passées, qu'elles soient bonnes ou mauvaises. Ainsi, notre système de gestion de la confiance sera basé sur le modèle d'évaluation de la réputation.

Cette évaluation de la réputation est fondée sur l'évaluation dynamique de la fiabilité des informations contextuelles et du comportement des sources de contexte. Pour

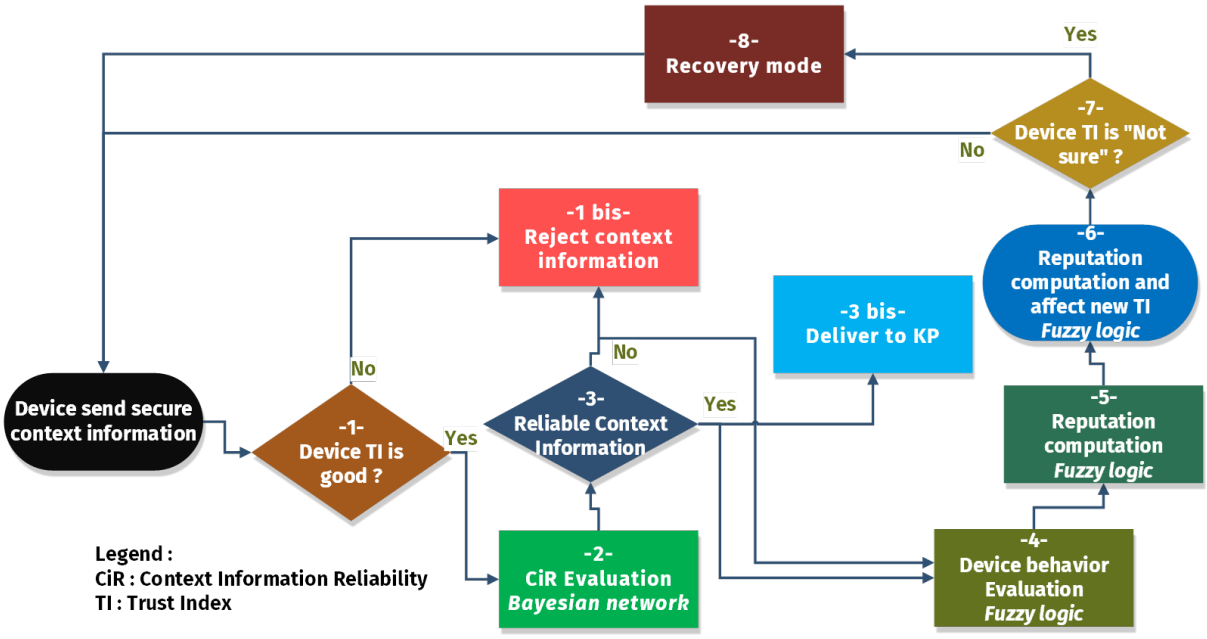



FIGURE 4.5 – Processus d'évaluation de la réputation proposée

ce faire, la fiabilité des informations contextuelles est évaluée en utilisant un réseau bayésien. Le comportement des sources de contexte est, quant à lui, évalué suivant les retours du Courtier MQTT (tentatives de connexion, tentatives d'authentification multiples, etc.) et du GCD (cf. Section 4.3.2.4) (par exemple : message non authentifié, etc.). La figure 4.5 illustre le processus d'évaluation de la réputation proposé.

### 4.3.3.2 Mécanisme d'évaluation de la réputation

Le mécanisme d'évaluation de la réputation des sources de contexte se base sur les comportements des dispositifs et la fiabilité des informations contextuelles délivrées par ces derniers (Fig. 4.5).

Spécifiquement, l'évaluation de la réputation d'une source de contexte se fait en trois étapes. Ces étapes sont décrites ci-dessous.

#### 4.3.3.2.1 Évaluation de la fiabilité des informations contextuelles

Comme introduit dans la section 4.3.3.1, le système de gestion de la réputation proposé se base en partie sur l'évaluation de la fiabilité d'une information contextuelle pour évaluer la réputation d'une source de contexte. L'évaluation de la fiabilité des informations contex-

tuelles dans un environnement de sécurité et de protection de la vie privée sensibles au contexte dans les applications IoT de la smart city consiste à établir la cohérence de cette information contextuelle avec le contexte réel de l'utilisateur. Elle permet de confirmer ou d'infirmer une information contextuelle en fonction de la qualité des capteurs, la présence d'autres capteurs et les informations sur l'utilisateur comme ses habitudes et son agenda.

On distingue plusieurs moyens de vérification, notamment la confrontation de l'information contextuelle provenant de la source considérée avec d'autres informations contextuelles provenant d'autres sources de contexte et avec les informations du profil utilisateur. Il existe plusieurs techniques d'intelligence artificielle pour la vérification et la validation : régression linéaire, machine à vecteurs de support, arbre de décision, réseaux de neurones, etc. [DGR+19]. Le choix d'une technique dépend du type d'apprentissage (supervisé, non supervisé), de la quantité de données d'entrée nécessaires et de la précision de ces données. D'une manière générale, il y a peu d'informations disponibles dans notre environnement (smart city) et ces informations sont souvent incertaines. Par exemple, en fournissant une information contextuelle (position géographique, date et heure), on doit être capable de savoir si cette position géographique est cohérente ou non. Ainsi, un réseau bayésien est bien adapté à ce cas de figure. En effet, l'avantage des réseaux bayésiens est qu'ils permettent de résoudre des problèmes avec un nombre de données limitées et comportant des incertitudes [**?**].

*Utilisation d'un réseau bayésien*

Un réseau bayésien est un graphe acyclique dirigé (DAG : Directed Acyclic Graph) dans lequel les nœuds représentent les variables aléatoires et les arcs représentent les probabilités de corrélations entre ces variables [SA09]. Il est utilisé pour modéliser les incertitudes et calculer ces incertitudes par le concept de probabilités. Le réseau obtenu suite à la modélisation d'un problème de prise de décision par un réseau bayésien représente une distribution de probabilité jointe.

Par ailleurs, la grande majorité des applications IoT de la smart city se base sur le profil de l'utilisateur pour lui offrir des services intelligents et personnalisés. Grâce aux informations provenant du profil de l'utilisateur disponible dans ces applications, il sera aisé de corréler les informations contextuelles avec ce profil pour en déterminer la fiabilité. Un contexte est caractérisé par des informations contextuelles permettant de déterminer un lieu (domicile, travail, centre commercial, salles de sport, etc.) et

une activité réalisée par l'utilisateur (repos, marche, sommeil, sport, conduite, etc.) à un moment bien précis (heure, jour, date, etc.). Ainsi, la présence d'un utilisateur sur un lieu peut être déterminée par son profil. En d'autres termes, cette présence peut être déterminée par les probabilités conditionnelles jointes de la géolocalisation, du temps (heure, jour de la semaine) et du réseau utilisé. La présence peut être confronté à l'agenda de l'utilisateur.

Notre approche consiste donc à utiliser un réseau bayésien (étape 2) pour combiner l'information contextuelle reçue avec le profil de l'utilisateur (le réseau, l'activité et l'agenda) afin de déterminer la probabilité de fiabilité de cette information. La figure 4.6 représente le réseau de connaissance du processus d'évaluation que nous avons défini. À partir de ces informations, des variables aléatoires déduites et les probabilités conditionnelles déterminées nous permettent de construire un réseau bayésien capable de déterminer la probabilité qu'une information contextuelle transmise à un moment donné (ex : Position GPS) soit fiable ou non. Au début, l'information contextuelle est considérée comme fiable (étape 3) si la probabilité déterminée par le réseau bayésien est supérieure ou égale à un certain seuil dont la valeur sera fixée à 80% suite aux résultats d'expérimentations détaillés dans la section 4.4.3. Elle n'est pas fiable dans le cas contraire. Ce seuil est défini afin de réduire au maximum le nombre de faux positifs résultant de cette évaluation. Le réseau bayésien apprendra au fur et à mesure de l'évolution du profil de l'utilisateur et des résultats de la détermination du contexte par le PC. Le module de gestion des préférences de l'utilisateur de l'architecture CASPaaS (cf. Section 3.4.3) est responsable de l'évolution du profil de l'utilisateur.

La distribution de probabilité jointe de l'ensemble des variables est définie dans l'équation 4.6 :

$$\begin{aligned} P(rci, cal, loc, act, cinf, d, t) = P(rci|cal, loc, act) \cdot P(cal) \cdot P(loc|cinf, d, t) \\ \cdot P(act) \cdot P(cinf) \cdot P(d) \cdot P(t) \end{aligned} \tag{4.6}$$

Avec $cal$ : l'agenda, $loc$ : la localisation, $act$ : l'activité de l'utilisateur, $d$ et $t$ : la date et le temps de l'observation de l'information contextuelle $cinf$. Quant à $rci$, il représente la probabilité de fiabilité de l'information contextuelle.

En évaluant les probabilités conditionnelles de chaque variable aléatoire, on obtient

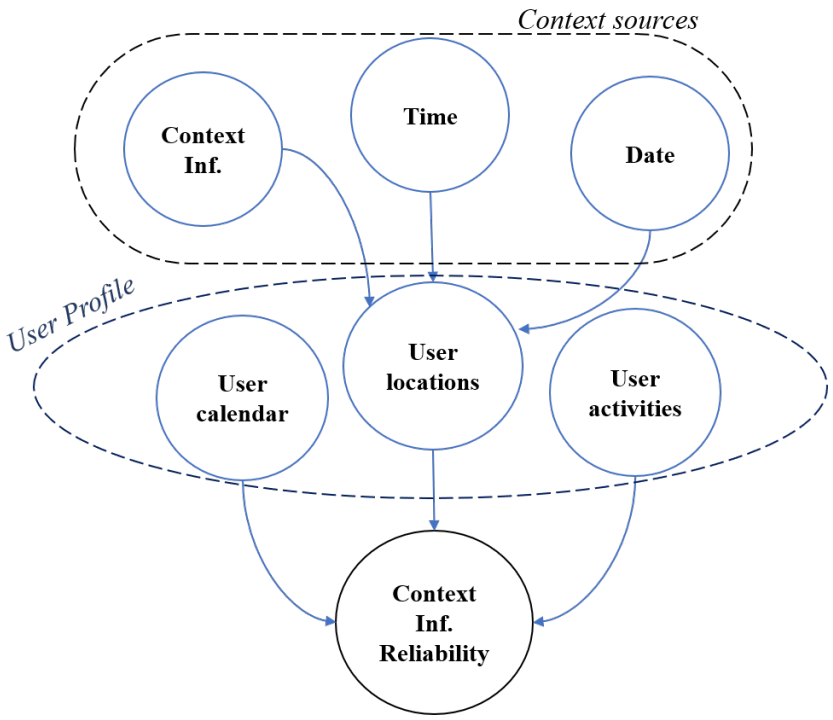



FIGURE 4.6 – Réseau bayésien d'évaluation de la fiabilité de l'information contextuelle

(équation 4.7) :

$$P(rci, cal, loc, act, cinf, d, t) = \frac{P(rci, cal, loc, act)}{P(cal, loc, act)} \cdot P(cal) \cdot \frac{P(loc, cinf, d, t)}{P(cinf, d, t)} \cdot P(act) \cdot P(cinf) \cdot P(d) \cdot P(t) \quad (4.7)$$

Si la probabilité de la fiabilité de l'information contextuelle $rci$ est en dessous du seuil de fiabilité, alors l'information contextuelle est rejetée (étape 1-bis). Cela peut être la résultante d'une fausse information contextuelle, éventuellement forgée.

Par exemple, il est fort probable que l'utilisateur Bob soit au bureau, un jeudi entre 8h et 17h, donc la probabilité jointe de la présence au bureau de l'utilisateur, sachant le temps (jour de la semaine, heure), la position GPS, le réseau WIFI sur lequel son téléphone est connecté, peut avoisiner 100%. L'accéléromètre et le pédomètre évoluent peu généralement durant cette plage horaire car Bob, étant assis à son bureau la majeure partie de son temps, effectue peu de mouvements. De plus, Bob a planifié une réunion de travail dans son agenda. Ces données confortent la probabilité que Bob soit effectivement au bureau à ce moment (jour, heure).

Cependant, Bob s'est fait volé sa montre connectée, il y a deux jours. Le système de sécurité et protection de la vie privée sensibles au contexte reçoit la position GPS provenant de la montre connectée de Bob. Cette position indique un lieu éloigné du domicile et du bureau de Bob. Lorsque cette position GPS, le jour de la semaine et l'heure sont passés au réseau bayésien, la probabilité de fiabilité de cette information peut être comprise entre 3 et 10%. Ainsi, avec un réseau bayésien bien construit et bien entraîné, il devient aisé de détecter de fausses informations contextuelles. Par conséquent, notre solution peut détecter les informations contextuelles forgées ou provenant de dispositifs clonés ou volés.

**4.3.3.2.2 Comportement des sources de contexte** Le protocole MQTT dispose de plusieurs paquets de contrôle des connexions : CONNECT, CONNACK, PUBACK, etc. À travers ces paquets, nous proposons d'évaluer (étape 4) les échanges afin de détecter les comportements suspects des différentes sources de contexte (essai multiple de connexion, essai multiple d'authentification, message anormalement grand, sujet (topic) non connu, etc.). Le mécanisme proposé utilise la logique floue pour détecter les comportements des sources de contexte au niveau du courtier, et cela en fonction de leurs activités.

*Utilisation de la logique floue*

La logique floue est une technique d'intelligence artificielle permettant de produire des raisonnements intéressants à partir de données incertaines et vagues [QKCK14]. Elle est flexible, ne nécessite que peu de données et tolère leur imprécision. En effet, les calculs sont basés sur les règles de type SI-ALORS.

D'une manière spécifique, le mécanisme proposé réalise des statistiques permettant d'avoir des indicateurs sur le comportement d'une source. Ce sont, entre autres, les taux de demandes de connexion, d'échec d'authentification et des messages ayant des tailles anormales. Ainsi, nous soutenons que ces indicateurs peuvent permettre de détecter le comportement d'une source de contexte. En effet, un taux de connexions avortées élevé peut indiquer qu'une source est compromise et qu'elle mène une attaque de déni de service contre le courtier. Un taux d'échec d'authentification élevé peut indiquer qu'une source malveillante essaie de s'authentifier sans succès. Une taille de message anormalement grand peut indiquer qu'une source essaie de saboter les performances du courtier. Un travail similaire a été proposé dans [APK19]. Cependant, les indicateurs et

les procédés utilisés dans ce travail diffèrent des nôtres. En effet, les auteurs ont utilisé les taux de demandes de connexion et d'acquittement de connexion pour un système de détection d'attaques par déni de service. En revanche, nos indicateurs permettent de rapidement déterminer le comportement d'une source de contexte.

Par conséquent, ces indicateurs proposés seront fuzzifiés et fournis en entrée à notre système d'inférence flou. Chaque variable (indicateur) floue comprend les sous-ensembles flous : low, high. Le sous-ensemble low contient les valeurs indiquant un ratio faible de l'indicateur considéré (taux de demandes de connexion, taux d'échec d'authentification et taux de messages de taille anormale). Le sous-ensemble high comprend les valeurs indiquant un ratio élevé de l'indicateur considéré. Par exemple, le taux d'échec d'authentification peut être considéré faible lorsqu'il est en dessous de 10% et élevé au-delà de ce seuil. Le taux de demandes de connexion est un indicateur différent du taux d'échec d'authentification et peut être considéré comme élevé lorsqu'il est au-delà de 40% et faible en dessous de cette valeur [BBBG19].

Généralement, en logique floue, les valeurs limites d'appartenance à un ensemble flou ne sont pas précisément définies [QKCK14]. Elles dépendent des objectifs recherchés. Ainsi, nous pouvons avoir différentes définitions du niveau "faible" ou du niveau "élevé". Nous estimons les possibilités dans la confiance pour les différents indicateurs. Un nœud qui scanne sporadiquement ou aléatoirement le courtier aura un taux d'échec de demande de connexion plus élevé par rapport à un nœud qui effectue des opérations légitimes. De même, un nœud qui a pour objectif de mettre le courtier hors service aura un taux d'échec de demande de connexion plus élevé. Ainsi, le taux d'échec de demande de connexion est 100% faible quand le ratio est compris entre 0 et 30%. De 25 à 100%, la probabilité que le taux soit élevé augmente tandis que la probabilité que le taux soit faible diminue.

Nous soutenons que le taux d'échec d'authentification d'une source de contexte est faible lorsqu'il est inférieur à 20%. En effet, un ou deux échecs pour dix authentifications n'est pas un révélateur d'un mauvais comportement. Par contre, un taux d'échec supérieur à 30% peut indiquer que la source de contexte mène une attaque par force brute ou une attaque de l'homme-du-milieu (MITM). En ce qui concerne le taux de messages de taille anormale, il en va de même que pour le taux d'échec d'authentification. Un nombre de messages de taille anormale élevé démontre qu'il s'agit d'une source de contexte malveillante qui envoie des paquets de données forgés, soit pour tromper la

perception de la sensibilité au contexte, soit pour mener une attaque de déni de service [BBBG19]. Les expressions d'évaluation des différents ratios et leurs fonctions d'appartenance sont présentées ci-dessous.

$ConnRateCS_i$ indique le taux de demandes de connexion pour une source de contexte $CS_i$ (Équation 4.8).

$$ConnRateCS_i = \frac{ConnectPacketCS_i}{TotalConnectPackets} \tag{4.8}$$

Avec $ConnectPacketCS_i$, le nombre de paquets CONNECT envoyés par la source $CS_i$ et TotalConnectPackets, le nombre total de paquets CONNECT MQTT observé sur une période. Les fonctions d'appartenance de $ConnRateCS_i$ sont :

$$\mu_{low}(ConnRateS_i) = \begin{cases} 0, & ConnRateS_i \leq 0 \\ \frac{ConnRateS_i}{15\%}, & 0 < ConnRateS_i \leq 15\% \\ \frac{30\% - ConnRateS_i}{30\% - 15\%}, & 15\% < ConnRateS_i \leq 30\% \\ 0, & ConnRateS_i \geq 30\% \end{cases} \tag{4.9}$$

$$\mu_{high}(ConnRateS_i) = \begin{cases} 0, & ConnRateS_i \leq 25\% \\ \frac{ConnRateS_i - 25\%}{50\% - 25\%}, & 25\% < ConnRateS_i \leq 50\% \\ \frac{100\% - ConnRateS_i}{100\% - 50\%}, & 50\% < ConnRateS_i \leq 100\% \\ 0, & ConnRateS_i \geq 100\% \end{cases} \tag{4.10}$$

$\mu_{low}\left(ConnRateS_i\right)$ et $\mu_{high}\left(ConnRateS_i\right)$ sont respectivement les fonctions d'appartenance des taux d'échecs de connexion *low* et *high* d'une source de contexte.

La variable $FARateCS_i$(Équation 4.11) représente le taux d'authentifications échouées pour une source de contexte $CS_i$.

$$FARateS_i = \frac{NFailedAuthS_i}{NAuth} \tag{4.11}$$

$NFailedAuthCS_i$ est le nombre d'authentifications échouées d'une source de contexte et $NAuth$ est le nombre total d'authentifications, c'est-à-dire, de toutes les sources observées sur une période.

$$\mu_{low}(FARateS_i) = \begin{cases} 0, & FARateS_i \leq 0 \\ \frac{FARateS_i}{10\%}, & 0 < FARateS_i \leq 10\% \\ \frac{20\% - FARateS_i}{20\% - 10\%}, & 15\% < FARateS_i \leq 30\% \\ 0, & FARateS_i \geq 20\% \end{cases} \tag{4.12}$$

$$\mu_{high}(FARateS_i) = \begin{cases} 0, & FARateS_i \leq 18\% \\ \frac{FARateS_i - 18\%}{40\% - 18\%}, & 18\% < FARateS_i \leq 40\% \\ \frac{100\% - FARateS_i}{100\% - 40\%}, & 40\% < FARateS_i \leq 100\% \\ 0, & FARateS_i \geq 100\% \end{cases} \tag{4.13}$$

$\mu_{low}\left(FARateS_i\right)$ et $\mu_{low}\left(FARateS_i\right)$ sont respectivement les fonctions d'appartenance des taux d'échec d'authentification *low* et *high* d'une source de contexte.

Finalement, $AMSRCS_i$ (Équation 4.14) indique le taux de messages de taille anormale envoyés par une source de contexte $CS_i$.

$$AMSRS_i = \frac{AMSS_i}{NMS} \tag{4.14}$$

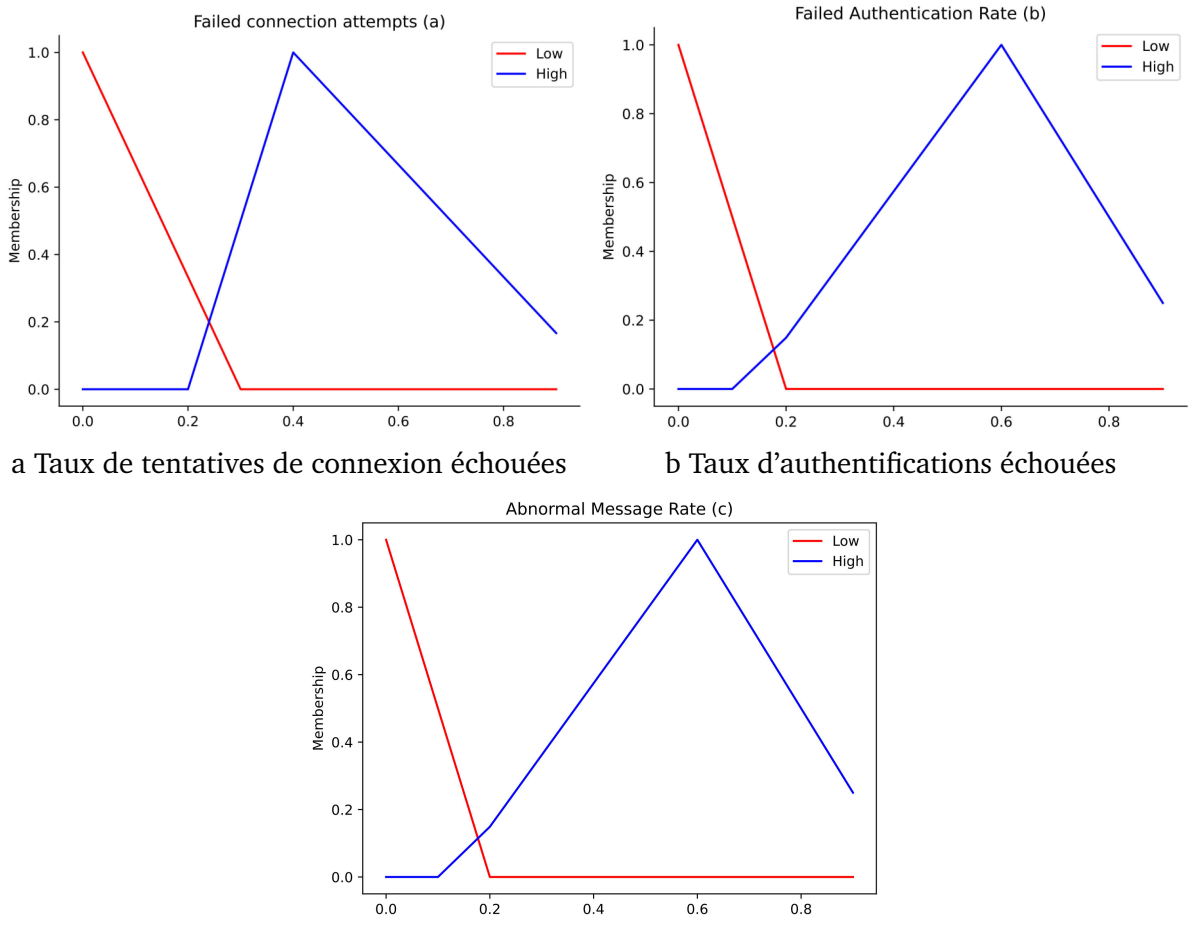
Avec $AMSRCS_i$, le nombre de messages de taille anormale envoyés par une source de contexte $CS_i$ et $NMS$, le nombre de total de messages de taille normale observés sur la période.

$$\mu_{low}(AMSRS_i) = \begin{cases} 0, & AMSRS_i \leq 0 \\ \frac{AMSRS_i}{10\%}, & 0 < AMSRS_i \leq 10\% \\ \frac{20\% - AMSRS_i}{20\% - 10\%}, & 15\% < AMSRS_i \leq 30\% \\ 0, & AMSRS_i \geq 20\% \end{cases} \tag{4.15}$$

$$\mu_{high}(AMSRS_i) = \begin{cases} 0, & AMSRS_i \leq 18\% \\ \frac{AMSRS_i - 18\%}{40\% - 18\%}, & 18\% < AMSRS_i \leq 40\% \\ \frac{100\% - AMSRS_i}{100\% - 40\%}, & 40\% < AMSRS_i \leq 100\% \\ 0, & AMSRS_i \geq 100\% \end{cases} \tag{4.16}$$

$\mu_{low}\left(AMSRS_i\right)$ et $\mu_{low}\left(AMSRS_i\right)$ sont respectivement les fonctions d'appartenance des taux de messages de taille anormale envoyés par une source de contexte. Les seuils des fonctions d'appartenance fixés dans les équations 4.9, 4.10, 4.12, 4.13, refeq :415 et 4.16 sont expliqués dans la section 4.3.3.2.2. Les fonctions d'appartenance respectives sont représentées sur les figures 4.7a, 4.7b et 4.7c.

Pour chaque règle de la base des règles, une implication appropriée devra être appliquée. Chaque implication est composée d'une prémisse et d'une conclusion. Le résultat de la règle d'implication est par la suite agrégé et défuzzifié pour obtenir le résultat final.




a Taux de tentatives de connexion échouées

b Taux d'authentifications échouées

c Taux de message de taille anormale

FIGURE 4.7 – Fonctions d'appartenance des indicateurs de comportement

Ce résultat sera utilisé par le mécanisme d'évaluation de la réputation des sources de contexte pour déterminer l'indice de confiance de la source de contexte. Ce mécanisme est décrit dans la section 4.3.3.2.3. Les variables floues d'entrées sont : $ConnRateCS_i$, $FARateCS_i$et $AMSRCS_i$. Le système d'inférence flou basé sur le modèle Mamdani utilise les règles floues pour déterminer le comportement de chaque source de contexte. Le tableau 4.2 représente les règles pour le comportement des dispositifs. La sortie du système d'inférence est également une variable floue. Elle est défuzzifiée afin d'obtenir les valeurs non floues représentant la décision résultante du processus. Comme décision, nous avons les valeurs suivantes : *good* (bon), *doubtfull* (douteux) et *malicious* (malveillant).

Tableau 4.2 – Table des règles de comportement des sources de contexte

| Index | TE de connexions | TE d'authentifications | TM anormaux | Comp. du disp. |
|---|---|---|---|---|
| 1 | Low | Low | Low | Good |
| 2 | Low | High | Low | Malicious |
| 3 | Low | Low | High | Doubtful |
| 4 | Low | High | High | Malicious |
| 5 | High | Low | Low | Doubtful |
| 6 | High | High | Low | Malicious |
| 7 | High | Low | High | Malicious |
| 8 | High | High | High | Malicious |

Légende : TE :Taux d'échec, TM : Taux de messages, Comp. :Comportement, disp. :dispositif.

**4.3.3.2.3 Réputation des sources de contexte** À l'initialisation du système, les sources de contexte ont l'indice de confiance (TI) maximal (Figure 4.5 - étape 1). A chaque remonté des informations contextuelles, cet indice est évalué. En effet, lorsque le GCD reçoit et déchiffre l'information contextuelle, il démarre le cycle d'évaluation de la réputation de la source de contexte. Pour ce faire, la fiabilité de l'information reçue et le comportement de la source de contexte sont successivement évalués. Si l'information contextuelle n'est pas fiable, alors elle est rejetée (étape 1-bis) et le processus d'évaluation de la réputation continue pour déterminer l'indice de confiance (TI). L'étape suivante évalue le comportement du dispositif (cf. Section 4.3.3.2.2). Une fois le comportement du dispositif déterminé et la fiabilité de l'information contextuelle évaluée, son indice de confiance est calculé par le mécanisme de gestion de la réputation (étape 5 et 6).

L'indice de confiance permet de directement savoir si une source de contexte est sûre ou non, sur une échelle variant de 0 à 1. Le tableau 4.3 résume les plages de valeurs de l'indice de confiance. À chaque évaluation de la réputation (étape 7), la valeur de l'indice de confiance est augmentée, réduite ou reste inchangée. Lorsque la valeur de l'indice de confiance TI atteint le niveau "Pas sûr", le système envoie une notification à l'utilisateur et met provisoirement le dispositif en mode récupération (étape 8). Dans ce mode provisoire, les informations contextuelles sont évaluées mais ne sont pas délivrées au PC. Il a pour objectif de réinitialiser le niveau de confiance d'un dispositif avec la prise en charge de l'avis de l'utilisateur. La mise en mode récupération peut intervenir dans

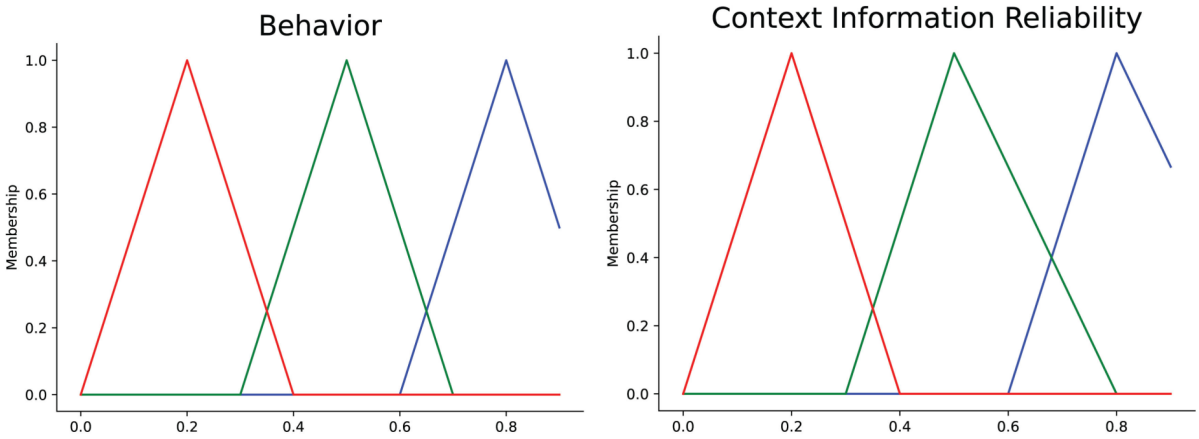
les cas où un dispositif a été retrouvé après un vol ou lorsqu'un dispositif a des capteurs qui doivent être recalibrés. Cependant, lorsqu'un dispositif est mis plus de deux fois en mode récupération, cela peut être le signe qu'il s'agit d'un dispositif compromis. Dans ce cas, le système notifie à l'utilisateur que ce dispositif n'est plus sûr et qu'il devra être retiré des sources de contexte. À ce niveau, l'utilisateur peut, dans la mesure du possible, effectuer la réinitialisation matérielle du dispositif concerné et l'ajouter à nouveau au système.

Tableau 4.3 – Plage de valeurs de l'Indice de confiance

| **Niveau de confiance** | **Indice de confiance** |
|---|---|
| Sûr | $TI \geq 0.8$ |
| En faute (Défection temporaire) | $0.6 \leq TI < 0.8$ |
| Douteux | $0.4 \leq TI \leq 0.6$ |
| Pas sûr | $TI < 0.4$ |

Le mécanisme proposé calcule l'indice de confiance en utilisant la logique floue. L'évaluation de l'indice de confiance comporte de l'incertitude car elle est basée en partie sur un élément incertain et flou qu'est le comportement du dispositif. Ainsi, la logique floue est très bien adaptée à ce cas de figure car elle permet de gérer les incertitudes qui ne peuvent pas être strictement traitées avec la vraisemblance de la probabilité. De ce fait, les variables floues d'entrées sont la fiabilité de l'information contextuelle et le comportement de la source de contexte. La fiabilité de l'information contextuelle a pour sous-ensembles flous : *not reliable, doubtful* et *reliable*. Ces éléments sont les résultats possibles du mécanisme d'évaluation de la fiabilité des informations contextuelles. Le comportement de la source de contexte a pour sous-ensembles flous la sortie du mécanisme d'évaluation du comportement des sources de contexte, c'est-à-dire, *good*, *doubtful* et *malicious*. Les fonctions d'appartenance caractérisant la fiabilité de l'information contextuelle et le comportement de la source de contexte sont représentées sur les figures 4.8a et 4.8b.

Bien que la fiabilité des informations contextuelles et le comportement des dispositifs sources de contexte soient deux informations de natures différentes, nous les combinons pour avoir l'indice de confiance. Cela est fait à travers l'utilisation d'une base de règles en suivant un système d'inférence flou Mamdani. Le tableau 4.4 représente les règles utilisées par le système d'inférence flou. Le résultat de cette combinaison représente l'in-




a b

FIGURE 4.8 – Fonctions d'appartenance de la fiabilité et du comportement

Tableau 4.4 – Règles d'inférence pour l'évaluation du niveau de confiance

| **Index** | **Fiabilité I.C.** | **Comp. de la source** | **Niveau de confiance** |
|---|---|---|---|
| 1 | Reliable | Good | Sure |
| 2 | Reliable | Doubtful | Faulty |
| 3 | Reliable | Low | Doubtful |
| 4 | Doubtful | Malicious | Faulty |
| 5 | Doubtful | Doubtful | Doubtful |
| 6 | Doubtful | Malicious | Not sure |
| 7 | Not reliable | Good | Faulty |
| 8 | Not reliable | Doubtful | Not sure |
| 9 | Not reliable | Malicious | Not sure |

Légende :I.C. Information Contextuelle, Comp. : Comportement,

dice de confiance, et par conséquent le niveau de confiance de la source de contexte : *sure* (sûr), *doubtful* (douteux), *faulty* (en faute), et *not reliable* (pas sûre (compromis)). Comme pour le mécanisme d'évaluation du comportement, la base de règles est exprimée sous la forme Si - Alors, Sinon. Par exemple, si l'information contextuelle est fiable et le comportement bon, alors le niveau de confiance est sûr. Par contre, si l'information contextuelle est douteuse et le comportement bon alors le niveau de confiance est en faute. Dans tous les cas, la réputation de la source de contexte est évaluée à une action précédente près. Cela permettra au système de sécurité et de protection de la vie privée sensibles au contexte de rapidement écarter les sources de contexte compromises.

Le système étant centralisé et les sources ne s'évaluant pas mutuellement, il est

résistant aux attaques de *Ballot stuffing* et de *Bad mouthing*. Les identités des sources de contexte ne peuvent pas être usurpées (voir Section 4.3.2.1). Ainsi, le système proposé est résistant aux attaques de changement d'identité.

## 4.4 Évaluation des performances

Dans cette section, nous analysons les performances de la solution d'échange sécurisé des informations contextuelles dans un environnement de confiance. Après la description de l'implémentation et de la configuration de l'expérimentation, nous évaluons, dans un premier temps, l'impact de notre solution d'échange sécurisé d'information contextuelle, par rapport à SSL/TLS, sur les performances : temps d'exécution (chiffrement, déchiffrement, envoi et acquittement), espace mémoire, surcharge protocolaire (*overhead*) et consommation énergétique. Dans un second temps, nous analysons les performances de notre système de gestion de la confiance en termes de temps d'évaluation de la fiabilité de l'information contextuelle, de comportement et de détection d'un nœud malveillant. Nous évaluons, par la suite, le taux de détection de nœuds malveillants.

### 4.4.1 Configuration de l'expérimentation

Pour mener à bien nos expérimentations , nous avons considéré un utilisateur d'une application de maison intelligente possédant trois dispositifs IoT comme source de contexte : une montre connectée, un pédomètre (compteur de pas) connecté et un smartphone. Ces dispositifs IoT sont basés sur Raspberry Pi Zero w ayant une puce Broadcom BCM2835 basé sur le processeur ARM1176 de 700 MHz et une mémoire vive de 256 Mo. Ils sont configurés avec le système Raspbian Buster et possèdent une connectivité Wi-Fi. Le GCD est hébergé comme un service sur un ordinateur Dell configuré avec Ubuntu 18.04 LTS (64-bit), Intel Core i7 vPro 5ème génération Dual Core 2.60 GHz et 12 Go de RAM.

Le client MQTT utilisé au niveau des sources de contexte est basé sur la librairie Paho MQTT Embedded C, une bibliothèque open source légère et largement utilisée dans la recherche et l'industrie de l'IoT [Ecl20]. Le client MQTT a été écrit en C++. Ainsi, la librairie CryptoPP [Dai19] a été utilisée pour l'écriture de l'algorithme cryptographique mis en œuvre pour assurer la sécurité de l'échange des informations contextuelles. Le Broker MQTT utilisé est la version 3.1.1 de Mosquitto, un broker open source, largement

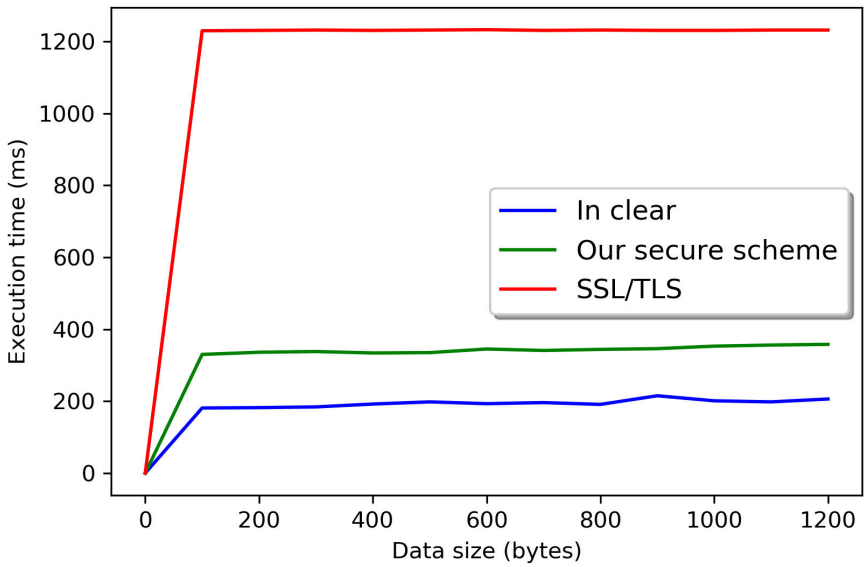



FIGURE 4.9 – Temps moyen d'un envoi d'une information contextuelle

utilisé [57] et écrit en C++/Python. Nous avons utilisé la librairie PyAgrum pour développer le réseau bayésien pour la détection de la fiabilité de l'information contextuelle [tea]. C'est une librairie puissante écrite à la base en C++ puis adaptée en Python. Le langage Python a également été utilisé pour écrire les algorithmes de logique floue à travers la librairie SkFuzzy [Sci]. SkFuzzy est une bibliothèque open source performante écrite en Python et permettant de créer des algorithmes de logique floue complexes.

### 4.4.2 Comparaison de la surcharge de SSL/TLS et de notre solution

Pour démontrer la faisabilité et les avantages de notre solution en termes de temps d'exécution et de consommation de mémoire dans les dispositifs contraints, nous réalisons une comparaison avec un système de gestion de la sensibilité au contexte n'implémentant pas de sécurisation des échanges des informations contextuelles et un système basé sur l'utilisation de SSL/TLS pour la sécurisation de ces échanges. Pour cela, nous utilisons le client Paho MQTT C pour l'IoT avec une authentification mutuelle TLS et un certificat de 2048 bits.

La surcharge requise pour chaque charge utile, par conséquent par chaque paquet envoyé avec le système proposé, est de 40 octets, comparé au même paquet envoyé sans sécurité et ayant une taille moyenne d'environ 100 octets. En effet, cela est justifié par

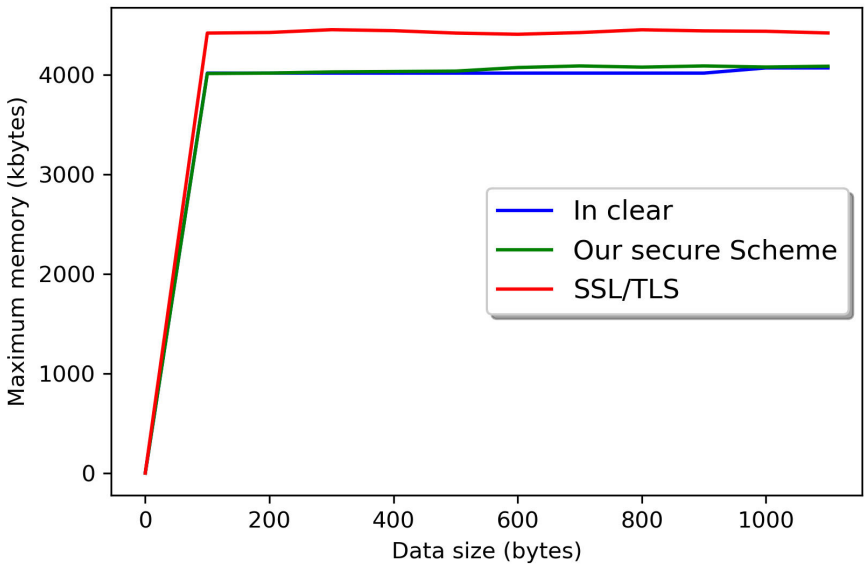



FIGURE 4.10 – Comparaison des systèmes sur le plan de l'usage maximal de la mémoire

l'ajout des données d'authentification par AES-CCM d'une taille fixe de 8 octets et du tag (HMAC) authentifiant le message ayant également une taille fixe de 32 octets. La grande partie de la taille de la surcharge étant proportionnelle à celle de la clé publique, la surcharge aura sensiblement la même taille, quel que soit la taille des informations contextuelles à envoyer. Elle est acceptable et ne nécessite que 40 octets de bande passante supplémentaire.

La figure 4.9 illustre le temps de traitement des messages envoyés avec le système proposé, comparé à un système sans sécurité et à un système sécurisé avec SSL/TLS. La surcharge du système proposé est acceptable par rapport à celle induite par la sécurité SSL/TLS. L'utilisation du protocole SSL/TLS nécessite un établissement de session, c'est-à-dire, la mise en œuvre du handshake TLS à chaque opération publish. Cela comprend les messages *ClientHello* et *ServerHello* d'en moyenne 250 octets, l'échange et la vérification des certificats, en moyenne 3000 octets, l'échange de clé, etc. Ainsi, ces échanges initiaux génèrent une surcharge importante que notre solution permet de réduire (nous avons l'envoi unique de la clé publique lors du premier échange).

Par ailleurs, le Handshake du protocole TLS est gourmand en ressources et est, par conséquent, consommateur d'énergie, contrairement au système proposé. Ainsi, un envoi d'une information contextuelle avec SSL/TLS nécessite en moyenne 1230 ms. Par

rapport à SSL/TLS, le système que nous proposons permet un gain d'environ 900 ms. Le système proposé, comparé à un système sans sécurité, présente un temps d'exécution supplémentaire d'environ 160 ms. Comparé à SSL/TLS, ce délai est acceptable parce qu'il a un faible impact sur le temps global de la gestion de la sensibilité au contexte, comprenant le temps de la collecte et de la transmission et le temps de traitement des informations contextuelles. Un système de sécurité sensible au contexte doit traiter très rapidement les informations contextuelles.

En ce qui concerne les performances en termes de consommation de la mémoire, notre système utilise au maximum 4000 Ko. Il en est de même pour un programme de collecte d'informations contextuelles ne mettant pas en œuvre la sécurité de ces échanges. En revanche, le client Paho MQTT C utilisant SSL/TLS consomme près de 10% de mémoire en plus, comparé à notre solution. La figure 4.10 illustre ces comparaisons. La consommation supplémentaire de mémoire du client Paho MQTT C par rapport au système proposé est due à la consommation excessive de ressources nécessaires pour l'initialisation des échanges basés sur SSL/TLS. En effet, SSL/TLS affecte les performances de manière significative, notamment au niveau de l'usage du processeur durant la phase d'échange. De plus, il impacte négativement les performances énergétiques des dispositifs. Ainsi, le système proposé a une efficacité sur le plan de l'utilisation des ressources (processeur et mémoire) des dispositifs contraints. Par conséquent, il a un faible impact énergétique. La figure 4.11 illustre la consommation énergétique moyenne des trois systèmes pendant leur temps d'exécution moyen respectif de 186 ms, 300 ms et 1210 ms.

### 4.4.3 Gestion de la confiance des sources de contexte

La première étape du système de gestion de la confiance proposé est l'évaluation de la fiabilité des informations contextuelles. Ce mécanisme basé sur un réseau bayésien (figure 4.6) n'a besoin que de quelques informations pour une bonne évaluation de la fiabilité. Dans notre simulation, nous avons considéré la position géographique et la vitesse de mouvement comme informations contextuelles primaires fournies par les dispositifs de l'utilisateur. Selon certains cas, les informations provenant du profil comme l'agenda et/ou les activités habituelles de l'utilisateur peuvent être utilisés pour augmenter la précision de l'évaluation. Pour mener à bien cette expérimentation, le réseau a été entraîné avec plus de 1400 conditions de probabilités jointes construites à partir

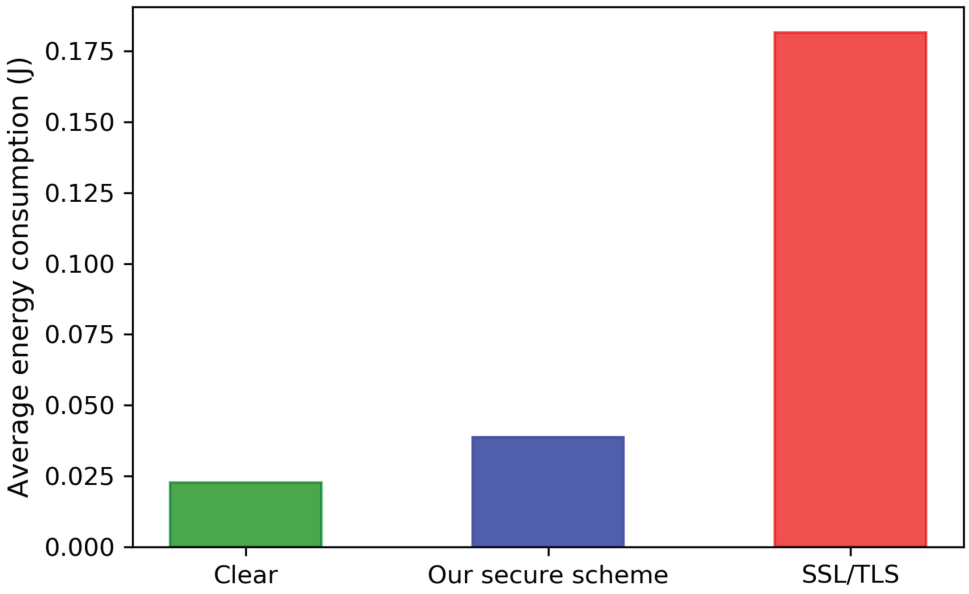



FIGURE 4.11 – Comparaison des systèmes sur le plan de l'usage maximal de la mémoire

des habitudes simplifiées de Bob.

Par exemple, nous considérons le cas de Bob (cf. section 4.3.3.2.1). Il est jeudi 10h, selon son profil, Bob est censé être au travail dans son bureau. Lorsque Bob est dans son bureau, il ne fait pratiquement pas de mouvement car il reste assis la plupart du temps. La montre connectée de Bob, source de contexte, envoie la position géographique et la vitesse de mouvement de ce dernier. Après l'extraction de la zone géographique par la technique de geofencing, la zone déterminée est le lieu de travail. La vitesse de mouvement notée m, la zone géographique notée p sont fournis au système. Les données de l'agenda notées c et de l'activité habituelle notée a ne sont pas connues dans cet exemple. Par conséquent, elles ont comme valeurs les valeurs par défaut : respectivement rien et inconnue. La figure 5.12 illustre le résultat de l'évaluation de la fiabilité des informations contextuelles transmises. L'inférence a été effectuée en moins d'une milliseconde, ce qui prouve la rapidité de l'évaluation de la fiabilité de notre réseau bayésien. Le résultat de l'évaluation indique que les informations contextuelles fournies sont fiables à plus de 83%.

L'évaluation de la fiabilité de l'information contextuelle précède l'évaluation du comportement du dispositif source de contexte. Comme expliqué précédemment (Section 4.3.3.2.2), le système proposé détermine le comportement d'une source de contexte par

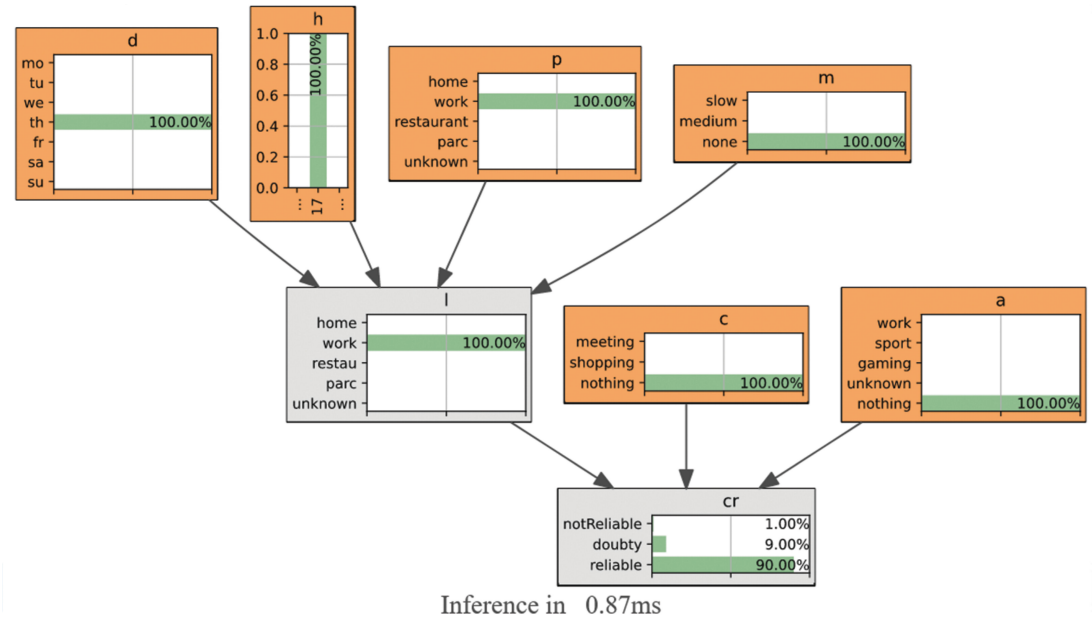



FIGURE 4.12 – Résultat de l'inférence du réseau bayésien pour l'évaluation des informations contextuelles

le calcul des indicateurs de comportement, c'est-à-dire, le taux de connexions échouées ($ConnRateCS_i$), le taux d'authentifications échouées ($FARateCS_i$) ainsi que le taux de messages de taille anormale ($AMSRCS_i$) envoyés par le dispositif. $ConnRateCS_i$, $FARateCS_i$et $AMSRCS_i$ sont détaillés dans la section 4.3.3.2.2. La figure 4.13a illustre un cas pratique d'évaluation du comportement d'un dispositif source de contexte. Dans cette expérimentation, les taux suivants ont été recueillis sur le comportement du dispositif : $ConnRateCS_i$ :0, $FARateCS_i$ :0 et $AMSRCS_i$ :0. $ConnRateCS_i$ est calculé comme suit : les tentatives de connexions échouées de la $CS_i$ (égale à 0) divisées par le nombre total de paquet connect (égal à 20). $FARateCS_i$est égale à : les tentatives d'authentification échouées de la $CS_i$ (égale à 0) divisées par le nombre total d'authentifications. Enfin, $AMSRCS_i$ est obtenu en divisant le nombre de messages de taille anormale envoyés par la $CS_i$ (égal à 0) par le nombre total de messages reçus (égal à 50). Cela démontre que cette source de contexte n'a pas eu d'échec de connexion ni d'échec d'authentification et n'a également pas envoyé de message anormal. Le résultat de l'évaluation montre que le dispositif a un bon comportement.

La figure 4.13b illustre le cas d'une source de contexte malveillante. Les valeurs fournies à l'algorithme sont : $ConnRateCS_i$ : 0,5, $FARateCS_i$ : 0,4 et $AMSRCS_i$ :

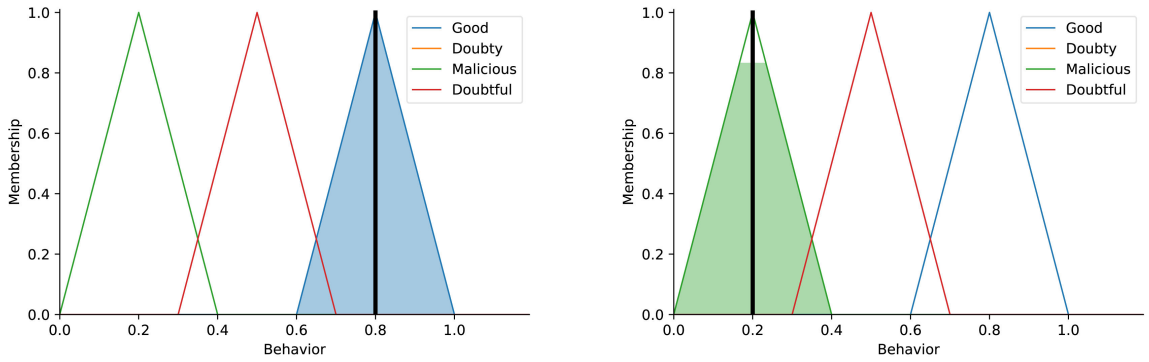
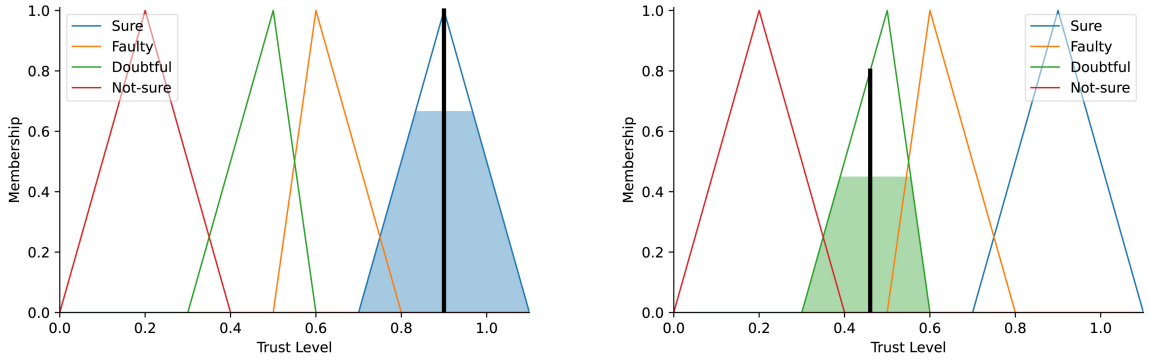



FIGURE 4.13 – Résultats de l'évaluation du comportement de deux sources de contexte

0,9. Ces valeurs témoignent des activités malveillantes de cette source de contexte. En effet, ces activités peuvent être expliquées par les paramètres suivants. Les tentatives de connexion échouées de la $CS_i$ sont au nombre de 50 sur un total de 100 tentatives de connexion. Les tentatives d'authentification échouées sont au nombre de 20 sur un total de 50 authentifications. Les messages de taille anormale détectés sont au nombre de 27 sur un total de 30 messages. Ainsi, le résultat de l'évaluation du comportement indique que la source de contexte a un comportement malveillant. Le système d'évaluation de la réputation utilise les sorties des mécanismes précédents pour estimer la réputation du dispositif. C'est la dernière étape du système de gestion de la confiance proposé (GCD). La figure 16 montre quelques résultats d'évaluation pour un nœud douteux (fig. 4.14b) et un nœud ayant un bon indice de confiance TI (fig. 4.14a).

Le ratio de faux négatifs représente le taux d'informations contextuelles non fiables détectées comme fiables (4.17). Ce ratio diminue avec l'enrichissement du profil de l'utilisateur, augmentant par la même la précision du mécanisme. En considérant la rapidité et l'efficacité de détection du mécanisme, nous pouvons conclure qu'il a une faible surcharge sur la mise en œuvre de la gestion sécurisée de la sensibilité au contexte dans les applications IoT susmentionnées.

La fiabilité des informations contextuelles provenant de la source de contexte ayant obtenu une bonne réputation est de 90% et elle a un bon comportement. La source de contexte avec une réputation douteuse présente des informations contextuelles fiable à 80%. Par contre, son comportement est estimé comme malveillant car elle présente un




a b

FIGURE 4.14 – Résultats de l'évaluation de la réputation de deux sources de contexte

Tableau 4.5 – Caractéristiques du mécanisme de gestion de la confiance

| **Propriétés** | **Valeurs** |
|---|---|
| Temps de détection | $\sim= 2millisecondes$ |
| Précision | $> 81\%$ |
| Vrai Positif (VP) | $\geq 80\%$ |
| Faux Positif (FP) | $< 4\%$ |
| Vrai Négatif (VN) | $\geq 99\%$ |
| Faux Négatif (FN) | $\leq 19\%$ |

taux d'échec de connexion élevé. Ainsi, notre système a une précision de détection élevée. Le système proposé présente un taux de faux positifs inférieur à 4% et une rapidité de détection proche de 2 millisecondes. L'évaluation de la précision a été effectuée sur plusieurs scénarios tels que : tout est cohérent, vol du smartphone et de la montre. Il faut également noter que le temps de détection n'est pas proportionnel au nombre de sources de contexte. En d'autres termes, le mécanisme a le même temps de détection avec une source de contexte et 10 sources de contexte. Ainsi, le mécanisme proposé permet de passer à l'échelle. Le tableau 4.5 résume les propriétés du mécanisme proposé sur 100 échantillons.

$$Précision = \frac{VP + VN}{VP + VN + FP + FN} [\text{BHNEA15}] \tag{4.17}$$

Le ratio de faux négatifs représente le taux d'informations contextuelles non fiables détectées comme fiables. Ce ratio diminue avec l'enrichissement du profil de l'utilisateur, augmentant par la même la précision du mécanisme. En considérant la rapidité

et l'efficacité de détection du mécanisme, nous pouvons conclure qu'il a une faible surcharge sur la mise en œuvre de la gestion sécurisée de la sensibilité au contexte dans les applications IoT susmentionnées.

### 4.4.4 Analyse de la sécurité et de la confiance

La mise en œuvre de la sécurité et de la protection de la vie privée sensibles au contexte dans les applications IoT de la ville intelligente donne lieu à plusieurs menaces de sécurité (voir Section 4.3.1). Dans cette section, nous analysons les propriétés de sécurité, de protection de la vie privée et de confiance de notre proposition.

#### 4.4.4.1 Sécurité et protection de la vie privée

Le système d'échange sécurisé des informations contextuelles proposé offre cinq services de sécurité, d'une part, et préserve la vie privée de l'utilisateur, d'autre part. Premièrement, il limite l'accès au GCD aux dispositifs autorisés avec authentification obligatoire de ces derniers avant toute remontée d'information contextuelle. Deuxièmement, avec l'utilisation du mode CCM d'AES, notre système assure l'authentification de l'origine des données en permettant à la source de générer un *tag* d'authentification chiffré, décrit dans la section 4.3.2.4, que seul le GCD est capable de déchiffrer avec sa clé privée. Troisièmement, ce mécanisme permet d'assurer l'intégrité des données échangées, et permet donc de pallier les attaques de modification de données. Quatrièmement, le chiffrement des informations contextuelles avec AES et l'utilisation d'une clé sécrète partagée et éphémère garantit leur confidentialité. Grâce à l'échange de clé intégré à ECIES, seul le GCD peut déchiffrer les informations contextuelles chiffrées avec sa clé publique et envoyées par les sources de contexte autorisées. Ainsi, les informations contextuelles sont protégées contre l'écoute clandestine. L'identification proposée permet d'éviter notamment les attaques d'usurpation d'identité.

Cinquièmement, le système proposé garantit la protection contre l'attaque de rejeu en empêchant les informations contextuelles ayant été précédemment reçues et traitées d'être traitées à nouveau, même si elles proviennent de sources de contexte de confiance. Cette propriété est assurée grâce au mécanisme anti-rejeu intégré (cf. Section 4.3.2.4). De plus, l'association au mécanisme de gestion de la réputation permet d'assurer la protection contre les attaques de déni de service (DoS). En effet, lorsqu'un dispositif a un comportement suspect qui peut être proche d'une attaque de déni de ser-

vice, il est rapidement détecté et ses communications seront rejetées. Ainsi, le système proposé permet d'assurer la disponibilité.

La vie privée de l'utilisateur doit être préservée dans un système de sécurité et de protection de la vie privée sensibles au contexte car les informations contextuelles comprennent en grande partie des informations sensibles sur l'utilisateur (localisation géographique, activité en cours, etc.). Le système proposé permet l'anonymisation des informations contextuelles en assurant leur envoi sans identification des utilisateurs. En outre, en garantissant la confidentialité de ces informations et en garantissant que seul le GCD peut avoir accès à ces informations, le système proposé préserve la vie privée de l'utilisateur. Un autre aspect de la préservation de la vie privée est la protection des données lors du stockage. Le système proposé ne stockant que temporairement les données et sous forme chiffrée, il limite ainsi les risques de divulgation de ces données en cas d'attaque et préserve par la même la vie privée de l'utilisateur.

Enfin, la sécurité matérielle doit être considérée. Notre système ne stocke pas les clés de chiffrement symétriques utilisés par AES. Cependant, il stocke les clés privées utilisées pour l'essentiel des opérations d'échanges sécurisés. En effet, le système cryptographique ECIES est un système hybride, c'est-à-dire qu'il met en œuvre la cryptographie asymétrique et la cryptographie symétrique. D'une part, il utilise un couple de clés privée/publique (échange Diffie-Hellman). La clé publique est dérivée à partir de la clé privée. D'autre part, la clé publique est par la suite utilisée par les différentes parties pour générer la clé secrète partagée, appelée aussi clé de session. Cette clé secrète est utilisée pour chiffrer les données avec un système cryptographique symétrique (par exemple AES-CCM). Ainsi, d'autres travaux sont nécessaires afin de protéger les clés privées des dispositifs contre les attaques d'extraction de clé. Une piste possible est l'utilisation de module de racine de confiance (RoT – Root of Trust), composé d'un élément sécurisé (SE – Secure Element) pour le stockage sécurisé des clés, d'une plateforme matérielle sécurisée (TPM – Trusted Platform Module) ou d'un environnement d'exécution privilégié (TEE – Trusted Execution Environment) pour les opérations de génération et/ou de dérivation de clés (publique, privée, sécrète éphémère, etc.) [SCA16].

#### 4.4.4.2 Gestion de la confiance

D'une manière générale, un système de gestion de la confiance basé sur la réputation est vulnérable aux attaques sur le score de confiance (*ballot stuffing, bad mouthing*), aux attaques de changement d'identité, etc. [FBM12, Suc13] (cf. Section 4.3.1). Notre proposition permet de pallier les attaques de changement d'identité grâce à la gestion des identités intégrée dans le mécanisme d'échange sécurisé des informations contextuelles. Elle permet également de pallier les attaques sur le score de confiance grâce à l'évaluation du comportement des sources de contexte et de la fiabilité des informations contextuelles qu'elles transmettent.

Par ailleurs, les données utilisées par le réseau bayésien pour l'évaluation de la fiabilité des informations contextuelles ont été fournies manuellement. Comme introduit dans la section 4.3.3.2, le module de gestion des préférences de l'utilisateur peut continuellement entrainer le réseau bayésien avec les nouvelles données du profil de l'utilisateur. Ainsi, plus il y aura de données sur le profil de l'utilisateur, plus le réseau bayésien sera efficace dans la détermination de la fiabilité de l'information contextuelle.

## 4.5 Conclusion

Dans ce chapitre, nous avons introduit un système de gestion sécurisée de la sensibilité au contexte dans un environnement IoT. Ce système assure la sécurité de l'échange des informations contextuelles et permet la détection des sources de contexte malveillantes ou compromises. Il permet également au système de sécurité et de protection de la vie privée sensibles au contexte, CASPaaS, de ne pas prendre de décisions d'adaptation de mécanismes de sécurité et de protection de la vie privée erronées. En effet, grâce au mécanisme anti-rejeu et à la fiabilité des informations contextuelles, CASPaaS ne traitera que les informations contextuelles fiables. La solution proposée a été implémentée et évaluée avec les communications basées sur MQTT. L'évaluation a prouvé son efficacité comparée au protocole SSL/TLS. L'évaluation a également prouvé l'efficacité du mécanisme de gestion de la confiance en termes de précision et de rapidité de détection. L'impact global de la solution dans un système de sécurité et de protection de la vie privée sensibles au contexte est acceptable en considérant le nombre de nœuds qu'un utilisateur peut posséder dans le cadre des applications IoT de la ville intelligente. Les résultats obtenus montrent que l'utilisation du profil de l'utilisateur pour détecter

les informations contextuelles non fiables peut donner de bons résultats.

Cependant, cette contribution présente quelques limites. Par exemple, les informations du profil de l'utilisateur utilisées sont statiques et fournies manuellement. Il pourrait donc être intéressant de rendre les informations du profil de l'utilisateur dynamiques et enrichies automatiquement. De plus, notre proposition a été validée en utilisant des données fournies en laboratoire. Nous devons donc évaluer son efficacité réelle pour des cas concrets. Ainsi, nous prévoyons d'effectuer des tests avec des données réelles que nous générerons avec des dispositifs réels (smartphone, montre connectée, etc.) dans un futur proche.

Une autre perspective intéressante consiste à implémenter la solution avec la sécurité matérielle et évaluer son impact sur l'ensemble des mécanismes proposés dans ce chapitre. Le chapitre suivant est consacré à la suite de ce travail et concerne la gestion décentralisée des autorisations sensibles au contexte dans l'IoT.

# Chapitre 5

# Gestion décentralisée des autorisations sensibles au contexte en tant que service dans l'IoT *Edge*

## 5.1 Introduction

L'émergence de l'Internet des Objets (IoT) a permis de développer de nouveaux services et applications impliquant les données et les dispositifs des utilisateurs. Cependant, ces services et applications peuvent présenter des risques pour la sécurité et la vie privée de ces utilisateurs. Récemment, des travaux de recherche ont permis de définir des solutions mettant en œuvre des mécanismes de sécurité et de protection de la vie privée sensibles au contexte de l'utilisateur [dTAH18b, SCKS20]. Ces solutions utilisent les informations contextuelles pertinentes (en partie collectées par les dispositifs IoT) afin de déployer des mécanismes de sécurité et de protection de la vie privée adaptés à la situation de l'utilisateur [SCFS21].

La sécurité des systèmes IoT concerne essentiellement les communications, le stockage des données et les accès à ces données [SCKS21]. Une architecture de sécurité et de protection de la vie privée a été proposée dans le chapitre 3 de cette thèse. Dans ce chapitre, nous nous intéressons à la sécurité de l'accès aux données stockées dans le cadre des applications et services IoT, plus spécifiquement à la mise en œuvre du module de gestion des autorisations proposées dans le chapitre 3. En effet, les mécanismes de gestion des autorisations permettent d'assurer le contrôle d'accès aux ressources protégées. Bien que la gestion des autorisations dans l'IoT ait fait l'objet de plusieurs travaux et standards, peu de ces travaux prennent en considération la sensibilité au contexte dans la mise en œuvre de ces mécanismes [LSW+18, SPC+17] et la décentralisation des serveurs d'autorisation.

Par ailleurs, l'IETF a proposé un cadre générique pour l'authentification et le contrôle

d'accès pour l'IoT, appelé Authentication and *Authorization for Constrained Environments* (ACE) et utilisant le *framework* Open Authorization (OAuth) 2.0 [LSW+18]. ACE-OAuth est basé sur OAuth 2.0 [Har12] et a pour objectif de permettre à une tierce partie d'avoir accès aux ressources protégées d'un dispositif appelé serveur de ressources à travers *Constrained Application Protocol* (CoAP) [SHB14]. En effet, la tierce partie, appelée client, doit posséder un jeton d'autorisation délivré au préalable par le serveur d'autorisation. Le client peut utiliser son jeton d'autorisation pour demander l'accès aux ressources protégées du serveur de ressources en utilisant, par exemple, *Datagram Transport Layer Security* (DTLS) ou *Object Security for Constrained RESTful Environments* (OSCORE) comme moyens de communications sécurisés.

DTLS [TF16] et OSCORE [SMPS19] sont des mécanismes de sécurité recommandés dans le *framework* ACE. Ils permettent aux dispositifs contraints de s'authentifier mutuellement et d'effectuer des communications sécurisées. Le protocole DTLS peut être utilisé pour sécuriser les communications CoAP, dont les échanges sont basés sur le protocole UDP. Le protocole OSCORE a été proposé pour sécuriser de bout en bout les échanges de CoAP qui ont lieu à travers des proxys intermédiaires. En effet, les informations ne sont pas sécurisées par DTLS lors du transit par les proxys intermédiaires. Un proxy malveillant peut espionner ou modifier les données en transit. La figure 5.1 illustre la mise en œuvre d'ACE avec un proxy malveillant.

Cependant, ACE présente plusieurs limites pour la sécurité d'une application IoT. D'une part, ces applications sont caractérisées par leur environnement dynamique. Ce qui rend indispensable l'adaptation dynamique des autorisations suite à l'évolution des risques (causée, à titre d'exemple, par le changement du contexte de l'utilisateur). Par exemple, un médecin peut avoir des autorisations contextuelles, lui permettant de consulter les données de santé d'un patient depuis l'hôpital, mais peut ne pas avoir l'autorisation d'accéder aux mêmes données quand il est hors de l'hôpital (par exemple depuis son domicile). Un autre exemple est de donner la possibilité à un patient d'attribuer des droits d'accès exceptionnels et contextuels aux données de son glucomètre intelligent à un infirmier (par exemple dans le cas d'une situation critique à l'hôpital). En revanche, en temps normal (une situation normale à l'hôpital), l'infirmier n'a qu'un droit d'accès limité aux données du glucomètre. D'autre part, la sécurité d'ACE dépend de la sécurité des communications (DTLS ou OSCORE) et du serveur d'autorisation. Ce dernier, centralisant toutes les opérations, peut représenter un point de défaillance

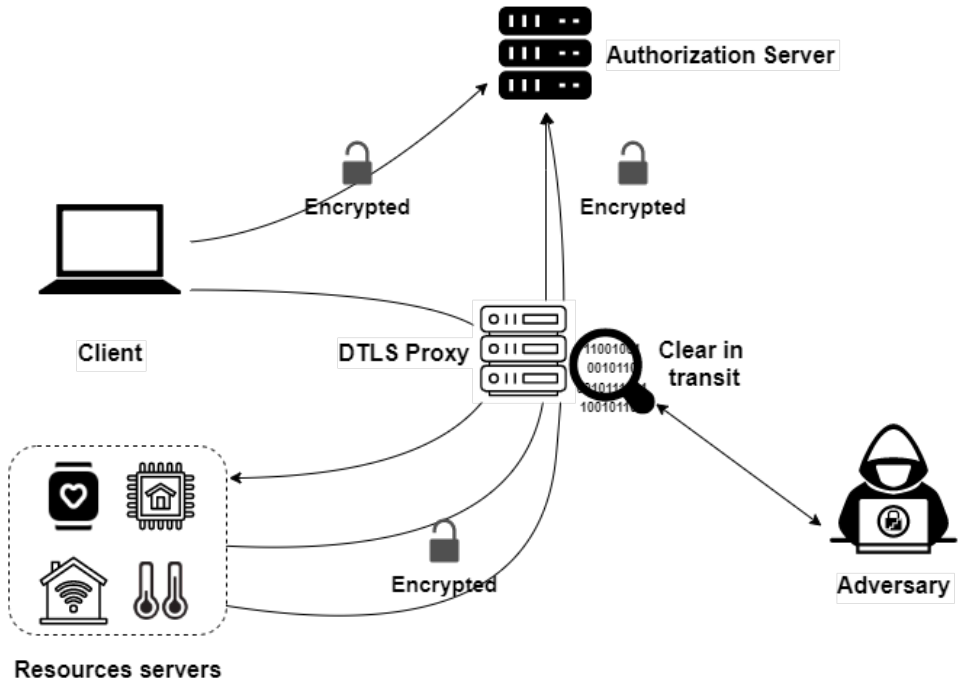



FIGURE 5.1 – Cadre d'autorisation ACE avec le profil de sécurité DTLS

unique parce qu'il peut tomber en panne ou être compromis.

Dans ce chapitre, nous proposons un système de gestion décentralisée des autorisations sensibles au contexte basé sur le *framework* ACE et pouvant s'intégrer dans l'architecture CASPaaS en tant que module de gestion des autorisations. A notre connaissance, un système de gestion décentralisée des autorisations sensibles au contexte n'a pas encore été proposé pour la gestion efficace des autorisations dans les environnements IoT hautement dynamiques. Ce système pourra donc s'intégrer à tout type de service de sécurité sensible au contexte pour l'IoT (par exemple avec la proposition effectuée dans [dTAH18b]). Dans ACE, les clients doivent présenter au serveur de ressources un jeton d'accès statique contenant les autorisations. Pour la prise en charge des autorisations basées sur des jetons dynamiques et contextuels, nous proposons de conditionner la génération de jeton d'accès à la validité du contexte de l'utilisateur. Pour pallier les risques de sécurité liés au transport des jetons dans le cadre d'ACE, nous proposons l'utilisation de jetons d'autorisation contextuels sécurisés.

Par ailleurs, comme nous l'avons mentionné précédemment, un serveur d'autorisation central peut représenter un point de défaillance unique et peut être compromis. Ainsi, pour renforcer la sécurité de notre proposition et pallier ce problème, nous

allons définir un système d'autorisation décentralisé en faisant appel à la blockchain [XTH+19]. En effet, la blockchain, dont la confiance et la sécurité ne sont plus à prouver, permettra à l'utilisateur de gérer les autorisations de manière flexible, dynamique et sûre sans se fier à un tiers de confiance. Grâce au système proposé, l'utilisateur pourra dynamiquement définir les droits d'accès contextuels de chaque application IoT dans un *smart contract* [CD16]. Le *smart contract* génèrera alors les jetons d'accès contextuels à la demande des clients autorisés. En outre, notre proposition met en œuvre l'approche *as a service* et peut ainsi être intégrée dans les nouvelles architectures réseaux. Ainsi, notre système permet un déploiement du service de sécurité sensible au contexte et de la blockchain dans une infrastructure de bordure (*Edge Computing* infrastructure). Cette dernière a fait l'objet d'une contribution et est discutée dans le chapitre suivant. Cela présente plusieurs avantages. Premièrement, le service peut être déployé dans n'importe quel point du réseau et ainsi être placé au plus proche de l'utilisateur. Ceci permet des communications temps réel et une gestion efficace de la mobilité de l'utilisateur. Deuxièmement, l'architecture proposée est dynamique, flexible et permet le passage à l'échelle (cf. Section 5.5.1). Les contributions majeures peuvent être résumées comme suit :

- Un nouveau système de gestion décentralisée des autorisations sensibles au contexte basé sur une extension du *framework* ACE ;
- Une sécurité et une confiance renforcées grâce à l'ajout de la technologie blockchain ;
- Une intégration « *as a service* » au sein de tout service de sécurité sensible au contexte dans l'IoT pour permettre une gestion dynamique et flexible des autorisations ;
- Une intégration des nouvelles architectures réseaux, notamment les infrastructures *Edge Computing* afin de mieux gérer la mobilité des utilisateurs, réduire la latence et ainsi permettre la mise en œuvre d'une sécurité dynamique et permanente des utilisateurs de l'IoT.

La suite de ce chapitre est organisée comme suit. La section 5.2 compare les principaux travaux portant sur la gestion des autorisations dans l'IoT. La section 5.3 introduit le background des principaux éléments abordés dans ce chapitre. La section 5.4 présente l'architecture du système proposé ainsi que ses principaux avantages. La section

5.5 analyse et discute les résultats des évaluations et la sécurité de cette architecture. Enfin, la section 5.6 conclut le papier et présente les futurs travaux.

## 5.2 Travaux connexes

La gestion des autorisations dans un environnement IoT a fait l'objet de plusieurs travaux. Dans cette section, nous analysons ces travaux et nous identifions leurs limites.

Un environnement IoT est un environnement dynamique, caractérisé par des changements fréquents dans le contexte global de l'utilisateur. C'est pourquoi Ramos et al. [RBS15] ont proposé une architecture de sécurité sensible au contexte pour l'IoT. Cette architecture comprend un module de gestion des autorisations sensibles au contexte. Le module proposé est chargé de gérer le contrôle d'accès ainsi que la gestion de jetons d'autorisation basés sur les capacités (émission, révocation, etc.). Dans l'architecture proposée, la gestion des autorisations est centralisée au niveau du système de gestion de la sensibilité au contexte. Cependant, les jetons d'autorisation proposés ne sont pas sécurisés. Ainsi, il est possible pour un adversaire d'intercepter et d'utiliser un jeton délivré à un client légitime.

Sciancalepore et al. [SPC+17] ont proposé une adaptation du standard OAuth 2.0 [HJ12] pour la gestion des autorisations et le contrôle de l'accès aux ressources dans un réseau IoT composé de dispositifs contraints. Les ressources des dispositifs du réseau considéré sont mises en cache au niveau de la passerelle. La passerelle agit en tant que serveur de ressources et vérifie ainsi les autorisations d'accès des applications clientes. Les jetons d'autorisation sont délivrés par le serveur d'autorisation après authentification des applications clientes. Cependant, le système proposé n'est pas sensible au contexte. Dans un environnement IoT, le contexte de l'utilisateur change fréquemment, les risques de sécurité évoluent et les autorisations doivent s'adapter en conséquence.

Dans [CRT17], un système d'authentification et de contrôle d'accès basé sur les cadres génériques d'ACE et d'OAuth1.0a a été proposé. Ce système a pour objectif de gérer et de transporter les jetons d'autorisation pour les dispositifs contraints dans un réseau non sécurisé. Il s'agit de jetons de Preuve de Possession (Proof of Possession) qui sont sécurisés par COSE (CBOR (*Concise Binary Object Representation*) *Object Signing and Encryption*). Comme dans [SPC+17], les jetons sont sécurisés mais ils ne sont pas sensibles au contexte.

De Matos et al. [dTAH18b] ont proposé une architecture de sécurité sensible au

contexte basée sur le partage de contexte dans un environnement distribué. L'architecture intègre un module chargé du contrôle d'accès selon les règles de sécurité sensible au contexte. Selon les auteurs, cette architecture devra être déployée dans une infrastructure Edge. Cependant, la gestion des règles peut devenir fastidieuse si le nombre d'applications clientes et/ou de dispositifs augmente. De plus, à l'instar des travaux de [RBS15], [SPC+17] et [CRT17], les jetons ne sont pas gérés dynamiquement, c'est-à-dire que la modification des accès d'un jeton et la révocation d'un jeton ne sont pas effectuées dynamiquement.

Les travaux de [RBS15], [SPC+17], [CRT17] et [dTAH18b] ont porté sur la mise en œuvre d'une gestion centralisée des autorisations. Cette gestion présente plusieurs limites. Le serveur d'autorisation peut constituer un goulot d'étranglement. Il est également possible qu'un serveur d'autorisation illégitime se fasse passer pour le serveur légitime. En outre, une gestion centralisée des autorisations dans les applications IoT pose le problème de la confiance des utilisateurs en une tierce partie.

Les auteurs de [AAC+18] ont proposé un système de gestion des autorisations décentralisé et sécurisé de bout en bout basé sur l'architecture OSCAR (*Object Security Architecture*) [VTR+15] et ACE, appelé IoTChain. Le serveur d'autorisation d'ACE est remplacé par la blockchain. L'architecture OSCAR, une amélioration de DTLS, est utilisée pour la gestion des clés et la sécurisation des échanges de jetons d'autorisation et des données. Cependant, à l'instar de [SPC+17] et [CRT17], ce système n'est pas sensible au contexte.

Zhang et al. [ZKS+19] ont proposé un système de contrôle d'accès distribué et de confiance basé sur la blockchain. Le système proposé comprend deux *smart contracts* de contrôle d'accès : un *smart contract* juge et un *smart contract* d'enregistrement. Ce système peut valider statiquement ou dynamiquement les accès et peut pénaliser les nœuds responsables de mauvais comportement. Cependant, ce système n'est pas sensible au contexte. En effet, la validation des accès ne prend pas en compte le contexte des utilisateurs. De plus, la solution proposée n'est pas assez flexible, n'est pas assez performante (latence) et supporte mal la mobilité des utilisateurs.

Arfaoui et al. [ACK+19] ont proposé une approche de contrôle d'accès sensible au contexte et basée sur les attributs (CAABAC – *Context-Aware Attribute-Based Access Control*). La solution proposée prend en compte les informations contextuelles ainsi que les attributs des utilisateurs dans le contrôle d'accès. Elle intègre, également, le

Tableau 5.1 – Comparaison des systèmes de gestion d'autorisation étudiés

| Travaux | J. Séc. | J. A. .C | G. D. J. | Authz D. | I. AS. | E. C. | Evalué |
|---|---|---|---|---|---|---|---|
| [RBS15] | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| [SPC+17] | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| [CRT17] | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| [dTAH18b] | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| [AAC+18] | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ |
| [ZKS+19] | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ |
| [ACK+19] | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |
| Notre travail | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

*Légende : J. Séc. : Jeton Sécurisé, J. A. C. :Jeton d'Accès Contextuel, G. D. .J : Gestion dynamique des jetons, Authz D. : Gestion décentralisée des autorisations, I. AS. : Intégration as a service, E.C. : Edge computing.*

chiffrement basé sur les attributs (*Ciphertext-Policy Attribute Based Encryption*) pour assurer la confidentialité des données. Cette solution a plusieurs avantages. Elle utilise le contexte pour adapter dynamiquement le contrôle d'accès aux ressources des dispositifs contraints. Elle résout le problème du tiers de confiance du CP-ABE. Enfin, elle permet de protéger les données de l'utilisateur. Cependant, cette solution n'est pas décentralisée et repose sur deux systèmes (le Centre de Génération de Clé (KGC) et l'Autorité des Attributs (AA)) qui peuvent être complexes à gérer pour un utilisateur lambda. En outre, à l'instar de [ZKS+19], la mobilité de l'utilisateur n'est pas prise en charge et le passage à l'échelle n'est pas garanti.

Cependant, comme vu dans la section 5.1 et mentionné dans [SCKS20] et [SCFS21], la sécurité sensible au contexte peut être une solution pour beaucoup d'applications IoT comme la e-santé. En outre, les préférences de l'utilisateur, son contexte et les risques liés à son contexte doivent être considérés dans la gestion des autorisations et dans le contrôle d'accès.

Le tableau 5.1 résume la comparaison des travaux étudiés portant sur la gestion des autorisations dans un environnement IoT. A notre connaissance, l'idée d'une gestion

dynamique et décentralisée des autorisations sensibles au contexte dans l'IoT basée sur OAuth2.0 et la blockchain n'a pas encore été proposée. C'est pourquoi, dans ce chapitre nous proposons une nouvelle architecture de gestion décentralisée des autorisations sensible au contexte pour l'IoT. En effet, d'une part, cette architecture permet de prendre en charge l'aspect dynamique de la plupart des environnements IoT, notamment la considération de la situation des utilisateurs dans la gestion des autorisations. D'autre part, l'architecture proposée permet aux utilisateurs de gérer directement les autorisations de manière décentralisée sans passer par une autorité de confiance tierce. Ainsi, dans ce chapitre, notre objectif est de démontrer l'avantage d'une telle approche (sécurité, sensibilité au contexte, dynamicité et passage à l'échelle) pour la gestion des autorisations dans les applications IoT.

Par ailleurs, l'architecture est conçue suivant l'approche '*as a service*'. Ce qui permet de relever les défis de la prise en charge de nombreuses applications IoT et de l'intégration nécessaire dans les nouvelles architectures réseaux, notamment les infrastructures *Edge Computing* [SCKS20]. L'architecture proposée est présentée dans la section 5.4. Comme cette architecture met en œuvre la sécurité des jetons et fait appel à la blockchain pour garantir la scalabilité, la section suivante décrit ces deux concepts.

## 5.3 Concepts fondamentaux

Dans cette section, nous décrivons les concepts fondamentaux abordés dans ce chapitre. Il s'agit de la sécurité des jetons dans le *framework* ACE et de la blockchain.

### 5.3.1 Sécurité des jetons dans ACE

OAuth2.0 est un protocole qui permet à un utilisateur d'accorder à une application web tierce partie l'autorisation d'accès à ses ressources protégées sans forcément révéler ses identifiants [Har12]. Les autorisations sont accordées aux applications web tierces parties, dénommées clients, sous forme de jetons. Les jetons d'accès dans OAuth 2.0 sont échangés sous le format *JSON Web Token* (JWT), dans lequel les informations nécessaires à la demande d'autorisation (identité du client, les accès autorisés, etc.) sont spécifiées et encodées. Pour sécuriser les jetons, le format *JSON Object Signing and Encryption* (JOSE) a été proposé. Ce standard comprend plusieurs spécifications pour le chiffrement, la signature et l'intégrité des jetons JWT [JBS15, JH15, Jon15b, Jon15a]. Cependant, OAuth 2.0 n'étant pas adapté pour les dispositifs contraints composant l'IoT,


FIGURE 5.2 – Format d'un jeton Preuve de Possession (PoP) COSE

le *framework* ACE a été proposé.

Pour répondre aux exigences des dispositifs contraints, notamment ceux de l'IoT dans ACE, le format *Consise Binary Object Representation* (CBOR) a été proposé [BH13]. L'objectif de ce format est de faciliter la définition et le transport des messages de petites tailles pour les applications de l'IoT. Les jetons sécurisés générés selon le *framework* ACE sont ainsi représentés sous le format *CBOR Web Token* (CWT) [JET18]. Le standard COSE a été défini dans le but de sécuriser les jetons CWT, tout comme JOSE pour les jetons JWT. La différence principale entre COSE et JOSE est l'utilisation du format concis et compact CBOR pour le premier alors que le second utilise le format JWT. Ainsi, dans notre système, nous proposons d'utiliser COSE, car il est sécurisé et adapté aux applications IoT. La mise en œuvre de COSE est détaillée dans la section 5.4.4.
La preuve de possession (*Proof of Possession* - PoP) est un mécanisme d'extension de jeton d'accès prévu pour permettre une association sécurisée des jetons et des requêtes permises par ces jetons [BHJ+18]. Une clé cryptographique est associée au jeton et permet au client de démontrer qu'il possède le secret associé lors de l'accès à une ressource. Le serveur de ressources vérifie que la clé utilisée par le client est bien celle associée au jeton. La clé associée au jeton peut être une clé mettant en œuvre une cryptographie symétrique ou asymétrique. La figure 5.2 illustre le format d'un jeton PoP. La charge utile comprend quatre champs. ISS (*issuer*) identifie l'émetteur du jeton. AUD (*audience*) identifie les destinataires du jeton. Ainsi, chaque client doit s'identifier obligatoirement par rapport à l'AUD avant de pouvoir traiter le jeton. EXP (*expiration time*) représente la durée de validité du jeton, au-delà de laquelle le jeton ne doit plus être accepté et traité. CNF (*confirmation*) est utilisée pour établir la PoP. La confirmation CNF a pour valeur un objet JSON, dont les membres identifient la clé PoP.

### 5.3.2 Blockchain

La blockchain est un registre distribué, répliqué et permanent dans lequel les enregistrements sont des blocs horodatés et signés [CD16]. Un bloc est identifié par une

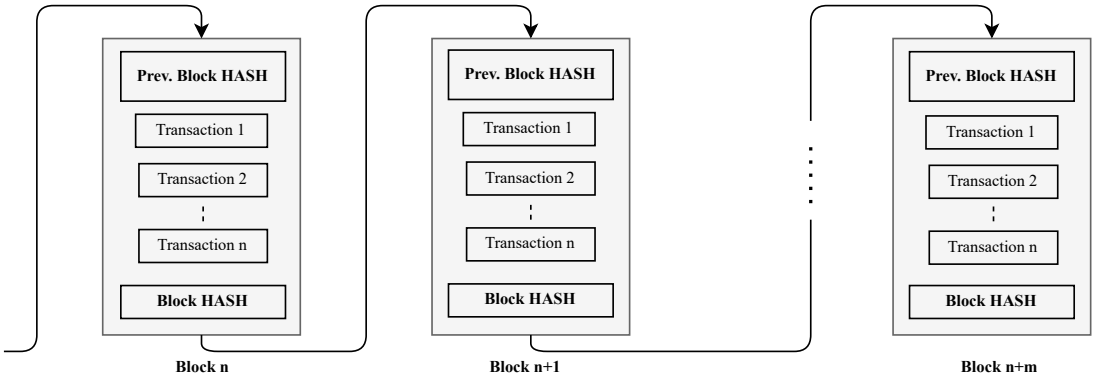



FIGURE 5.3 – Structure d'une chaine de blocs

empreinte numérique résultante d'un hachage cryptographique. L'empreinte numérique de chaque bloc pointe vers celle du bloc précédent, et ainsi de suite. Ce référencement des empreintes numériques des blocs forme une chaine de blocs, d'où le nom de *chaine de blocs* ou *blockchain*. La figure 5.3 illustre une chaine de blocs.

L'arbre de Merkle est une structure de données contenant les empreintes numériques de l'ensemble des transactions d'un bloc [NGHS17]. Le Merkle Root d'un bloc est créé à partir de l'empreinte numérique de chaque paire de transactions. Un seul bit d'une feuille de l'arbre de Merkle ne peut être modifié sans altérer le Merkle Root. Ainsi, le bloc est inaltérable. Chaque membre du réseau, appelé nœud, possède une copie complète de la chaine. A chaque fois qu'un nœud effectue une transaction, un mineur ajoute la transaction à un bloc, puis valide cette transaction. Un nouveau bloc n'est ajouté à la chaine qu'à condition qu'il ait été certifié par un algorithme de validation distribué appelé consensus. Les algorithmes de consensus les plus connus sont *Proof of Work* (PoW), *Proof of Stake* (PoS), *Pratical Byzantine Fault Tolerance* (PBFT) [MCK20c, MXZ+17, NGHS17, XTH+19].

La sécurité des échanges est assurée en utilisant la cryptographie asymétrique. Chaque utilisateur interagit avec la blockchain à travers une paire de clés publique/privée. La clé publique est utilisée pour identifier un utilisateur, tandis que la clé privée est utilisée pour signer les transactions. Ainsi, un réseau blockchain met en œuvre l'authentification, l'intégrité et la non répudiation. Avec l'utilisation de la cryptographie asymétrique et un mécanisme de consensus distribué, la blockchain met en œuvre un environnement transparent, fiable, sécurisé et évolutif, sans besoin d'un tiers de confiance central.

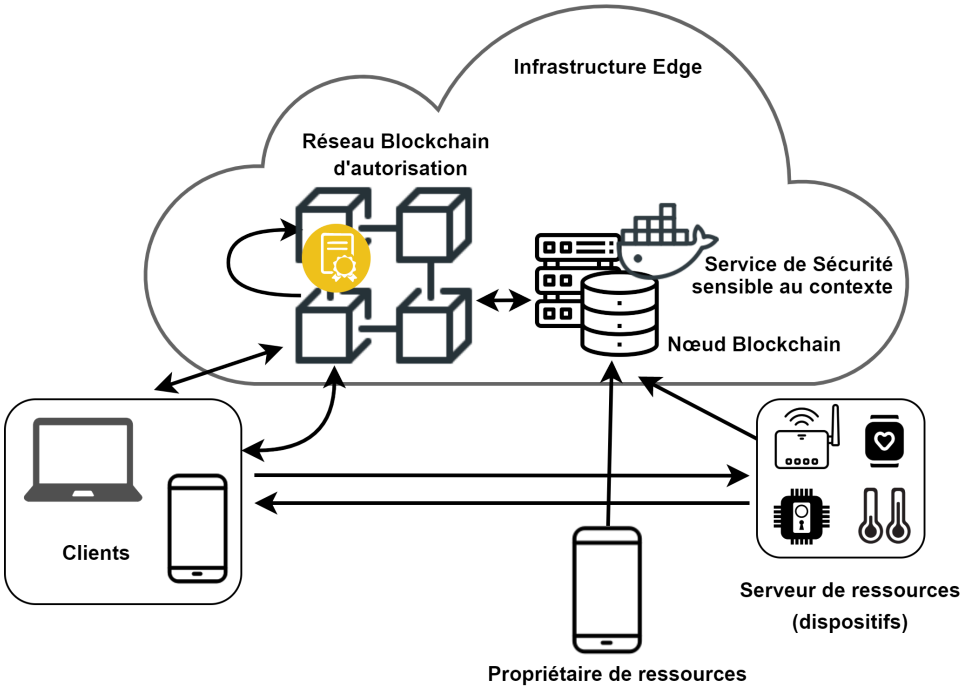



FIGURE 5.4 – Architecture pour la gestion décentralisée des autorisations sensibles au contexte dans l'IoT

Un autre élément caractérisant la blockchain est le *smart contract*. Le *smart contract* est un programme informatique stocké dans la blockchain qui permet d'exécuter du code à la demande et sous certaines conditions (analogie aux clauses d'un contrat écrit) [CD16]. Par exemple, la condition d'exécution d'une fonction d'un *smart contract* peut être le contexte, par exemple "l'utilisateur est à l'hôpital". Il aide à l'automatisation des échanges transparents à bas coût pour les parties prenantes sans l'intervention d'un tiers de confiance. Pour cela, un *smart contract* fournit plusieurs fonctions, appelées interface binaire d'application (*Application Binary Interface* – ABI), pour interagir avec lui.

## 5.4 Architecture du système proposé

Dans cette section, nous introduisons le système proposé, nous détaillons son fonctionnement et nous présentons ses principaux avantages. La figure 5.4 illustre l'architecture de ce système ainsi que les flux des échanges pour l'accès aux ressources IoT sécurisées des utilisateurs.

Nous proposons un nouveau système de contrôle d'accès sensible au contexte basé sur les jetons d'accès contextuels. L'architecture est composée des éléments du *frame-*

*work* ACE, du réseau blockchain d'autorisation décentralisée et d'un service de sécurité sensible au contexte pour l'IoT [SCKS20]. En effet, ce service de sécurité sensible au contexte dynamique et flexible, et conçu suivant l'approche '*as a service*', peut être dynamiquement placé dans n'importe quel point d'une infrastructure *Edge Computing*. L'intégration comme un service Edge permet une communication à très faible latence (indispensable pour les services temps réel) et la prise en charge de la mobilité des utilisateurs. Selon la terminologie d'ACE, les éléments de l'architecture proposée sont :

— **Propriétaire de ressources :** propriétaire légal des ressources protégées par les serveurs de ressources. C'est l'utilisateur de l'application IoT (application de e-santé, maison connectée, etc.).

— **Serveurs de ressources :** stockent et gèrent les ressources protégées. Dans notre contexte, il s'agit des dispositifs IoT, comme par exemple un capteur de température, un glucomètre, une montre intelligente, une serrure, etc.

— **Clients :** parties ayant besoin d'accéder aux ressources protégées par le serveur de ressources. Il s'agit généralement d'une application derrière laquelle il y a un utilisateur : un médecin, un électricien, etc.

— **Service de sécurité sensible au contexte :** reçoit les informations contextuelles et détermine le contexte de l'utilisateur. Il est responsable de l'adaptation dynamique des mécanismes de sécurité en fonction du contexte de l'utilisateur.

— **Réseau blockchain d'autorisation :** responsable de la génération des jetons d'autorisation contextuels pour l'attribution des autorisations aux clients sur ordre du propriétaire des ressources. Ainsi, les clients possédant les jetons d'accès contextuels valides peuvent accéder aux ressources protégées.

— **Nœud blockchain :** membre et partie prenante du réseau blockchain.

Nous supposons que les dispositifs contraints sont capables de réaliser les opérations de cryptographie asymétrique allégées, possèdent des certificats racines et sont capables de contacter une autorité de certification. Les dispositifs contraints qui n'en sont pas capables doivent passer par un proxy qui doit être capable de réaliser ces opérations (cryptographie asymétrique). Ainsi, les dispositifs concernés (serveurs de ressources) seront capables de signer et de vérifier les identités liées aux signatures.

### 5.4.1 Modèle de menaces

Un modèle de menace permet de modéliser les menaces et les adversaires auxquels un système peut être exposés. L'analyse des différentes menaces et adversaires est une étape primordiale dans la conception d'un système de sécurité et de protection de la vie privée. Le système proposé est exposé à plusieurs menaces :

— **Intégrité des jetons d'accès contextuels :** les jetons d'accès contextuels doivent être protégés contre la forge (création illégitime) et les modifications non autorisées. Leurs contextes de validité doivent également être protégés contre la forge et la modification (cf. Section 5.4.3). Un mécanisme doit être mis en place pour détecter les jetons d'accès contextuels volés ou illégitimes.

— **Service de sécurité sensible au contexte :** il génère et gère les contextes de validité selon les préférences du propriétaire de ressources. Il doit être protégé contre les attaques visant sa disponibilité. Les contextes de validité doivent rester protégés en cas de compromission du service de sécurité sensible au contexte.

— **Serveur de ressources :** ils sont contraints et sujets à plusieurs attaques physiques et réseau. Ils doivent être sécurisés afin de protéger les jetons d'accès contextuels lors du traitement et du stockage.

— **Propriétaire de ressources :** il peut être un utilisateur averti ou non sur la sécurité. Il doit être informé de l'importance de la sécurité de ses interactions avec le service de sécurité sensible au contexte. Chacune de ses interactions avec ce service doit être effectuée sur un canal de communication authentifiée et sécurisée. Il doit également être protégé contre l'usurpation d'identité.

— **Clients :** ils sont hétérogènes (ordinateur portable, smartphone, tablette, etc.) et stockent les jetons d'accès contextuels. Ils doivent utiliser des communications et sécurisées. Les jetons d'accès contextuels créés doivent être protégés contre le vol et les clients empêchés de forger des jetons valides.

### 5.4.2 Principaux avantages de ce nouveau système

La blockchain et les *smart contracts* offrent plusieurs avantages en termes de sécurité, de confiance et de performances (architecture décentralisée) [MCK20b]. Le déploiement d'un service de sécurité sensible au contexte dans une infrastructure de bordure offre également plusieurs avantages et possibilités pour la sécurité et la protection de

la vie privée sensibles au contexte dans l'IoT tels que la gestion de la mobilité des utilisateurs et la garantie de meilleures performances (faible latence et meilleur débit), la flexibilité et le passage à l'échelle [SCKS20]. Par conséquent, l'intégration de la blockchain et du service de sécurité sensible au contexte est une solution intéressante pour la gestion décentralisée des autorisations sensibles au contexte dans l'IoT. En effet, le réseau blockchain peut être hébergé dans l'infrastructure de bordure et le service de sécurité sensible au contexte de l'utilisateur hébergé en tant que service sur un nœud de ce réseau. Cela permet d'éliminer les problèmes de latence dus aux communications entre ce service et le nœud constituant le point d'entrée du réseau blockchain.

### 5.4.3 Flux d'autorisation

La figure 5.5 montre les différents échanges caractérisant l'architecture proposée. Dans ce qui suit, nous supposons que l'utilisateur est préalablement inscrit dans la blockchain d'autorisation et que les dispositifs de l'utilisateur sont également enregistrés au préalable auprès du service de sécurité sensible au contexte. Nous avons effectué un travail similaire dans le chapitre 4 concernant la gestion sécurisée des dispositifs de l'utilisateur par un service de sécurité sensible au contexte. Ces étapes d'enregistrements ne sont pas traitées dans ce chapitre. Le flux d'autorisation de l'architecture décentralisée proposé débute par la spécification des préférences par l'utilisateur (étape 1a). Cela s'effectue via le service de sécurité sensible au contexte. Ainsi, l'utilisateur peut spécifier à tout moment les applications qui peuvent obtenir l'autorisation d'accès contextuels, les ressources auxquelles elles peuvent accéder et dans quelles conditions elles peuvent obtenir ces autorisations. Le service de sécurité sensible au contexte contacte le *smart contract* en envoyant les données nécessaires à la génération des jetons d'accès contextuels (étape 1b). L'utilisateur est le seul à pouvoir autoriser la génération des jetons d'accès contextuels.

Le *smart contract* exécute la fonction "Générer un jeton d'accès contextuel" et génère dynamiquement le jeton d'accès contextuel sécurisé (étape 2). La fonction "Générer un jeton d'accès contextuel" du *smart contract* est similaire au point de terminaison de jeton (Token Endpoint) du serveur d'autorisation du *framework* ACE. Les jetons d'autorisation contextuels de notre proposition ont le format des jetons d'autorisation du *framework* ACE, avec en plus la prise en charge de la sensibilité au contexte. Nous proposons de lier à un jeton, l'identifiant de son contexte de validité (géré par le service de sécurité

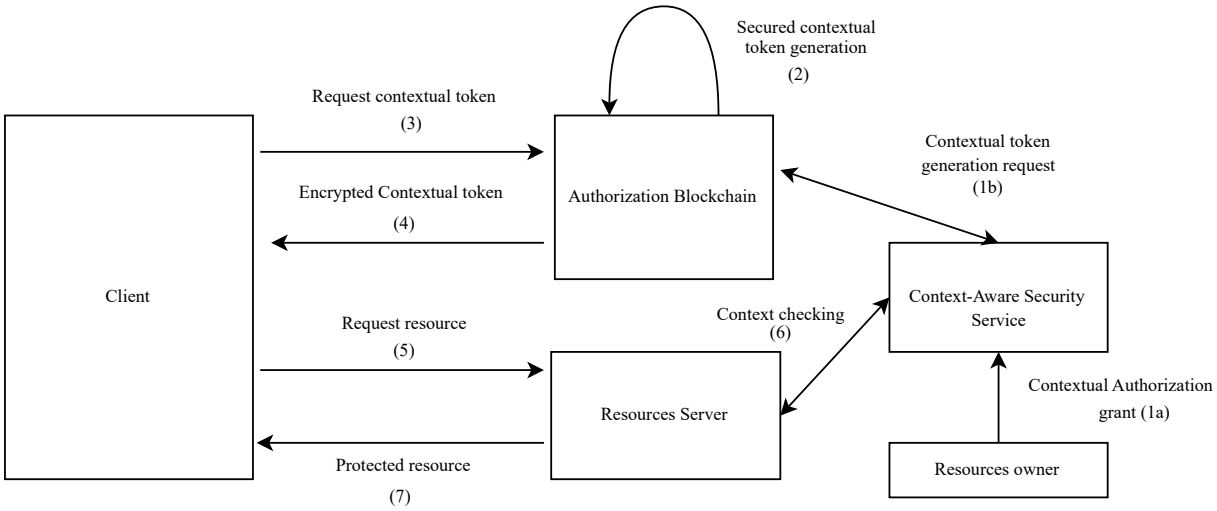



FIGURE 5.5 – Flux d'autorisation du *framework* proposé

sensible au contexte). Cela permet au *smart contract* de vérifier la validité du jeton d'accès contextuel auprès du service de sécurité sensible au contexte.

La durée de validité du jeton est définie lors de l'émission du jeton et dépend des cas d'utilisation. D'une manière générale, la durée de validité des jetons d'accès OAuth 2.0 peut varier entre 1 et 60 jours [Aut]. Ainsi, chaque jeton d'accès contextuel comprend l'empreinte numérique de l'identifiant de son contexte de validité. En outre, similairement à ACE, le jeton d'accès contextuel proposé comprend également les empreintes numériques de la clé publique du client et du serveur de ressources et deux signatures numériques générées par le *smart contract* à destination du client et du serveur de ressources (ici le dispositif IoT). Cela permet de protéger le jeton d'accès contextuel contre les modifications et aux destinataires (client et serveur de ressources) de s'assurer que le jeton d'accès contextuel a bien été généré dans la blockchain. La figure 5.6 représente l'extension du jeton PoP en jeton d'accès contextuel proposé.

La sécurité du jeton d'accès contextuel est discutée dans la section 5.4.4. En plus de la fonction "Générer un jeton d'accès contextuel", le *smart contract* présente les fonctions suivantes : "Demander un jeton d'accès contextuel", "Vérifier le contexte d'un jeton", "Révoquer un client ou un jeton d'accès contextuel". Le client a seulement le droit d'appeler la fonction "Demander un jeton d'accès contextuel". Les autres fonctions sont réservées au service de sécurité sensible au contexte de l'utilisateur. Le tableau 5.2 résume les fonctions du *smart contract* proposé.


FIGURE 5.6 – Jeton d'accès contextuel proposé *avec conid : empreinte numérique de l'identifiant de contexte, clid : empreinte numérique de l'identifiant du client*

Tableau 5.2 – Fonctions du *smart contract* de l'utilisateur

| **Fonctions** | **Peut être appelée par** |
|---|---|
| Générer un jeton d'accès contextuel | Service de sécurité sensible au contexte |
| Autoriser un client | Service de sécurité sensible au contexte |
| Demander un jeton d'accès contextuel | Client |
| Vérifier le contexte d'un jeton | Service de sécurité sensible au contexte |
| Révoquer un jeton d'accès contextuel | Service de sécurité sensible au contexte |
| Révoquer un client | Service de sécurité sensible au contexte |

Le client, qui souhaite accéder aux ressources protégées, doit contacter le *smart contract* de l'utilisateur (étape 3). Pour ce faire, il appelle la fonction "Demander un jeton d'accès contextuel" du *smart contract*. Le *smart contract* signe le jeton d'accès contextuel sécurisé puis le délivre au client (étape 4). Lorsque le client reçoit le jeton, il vérifie la signature du *smart contract* avant de procéder au déchiffrement du jeton d'accès contextuel. Le client présente le jeton d'autorisation contextuel au serveur de ressources (étape 5). La sécurité de la communication entre le client et le serveur de ressources et la sécurité du jeton sont discutées dans la section 5.4.4.2. Deux cas de figure peuvent se présenter : le dispositif est capable de directement communiquer avec le client (cas 1) ou le serveur de ressources est très contraint et doit passer par un proxy pour communiquer avec le client (cas 2).

Le second cas de figure ne sera pas étudié dans ce chapitre. Lorsque le dispositif est capable de communiquer directement avec le client (premier cas), le serveur de ressources doit authentifier le client puis valider le jeton en deux étapes. L'authentification du client par le serveur de ressources est un préalable à toute opération avec le client. Premièrement, il vérifie les signatures du *smart contract* et du client. Il vérifie ensuite la durée de vie du jeton. Si le jeton est encore valide, il vérifie la preuve de possession (PoP) du jeton par le client. Lorsque la preuve de possession est établie, le serveur de ressources utilise l'identifiant de contexte du jeton pour vérifier la concordance du

contexte de l'utilisateur avec celui du jeton d'accès auprès du service de sécurité sensible au contexte (étape 6). La vérification du contexte au moment de la demande permet de dynamiquement autoriser ou rejeter l'accès suivant le contexte de l'utilisateur. Par exemple, dans le cas de l'infirmier, si la vérification du contexte '*situation critique à l'hôpital*' (*la personne est localisée à hôpital et le monitoring de ses signes vitaux fait apparaître une situation très dégradée par exemple*) donne comme réponse *valide*, alors le glucomètre du patient (utilisateur) donne l'accès à l'application cliente (celle de l'infirmier). Si le contexte du jeton est valide, alors le client peut accéder aux ressources protégées (étape 7).

### 5.4.4 Sécurité des échanges de jetons et jetons sécurisés

Dans cette section, nous introduisons la sécurité des jetons d'accès contextuels et leurs échanges sécurisés entre le serveur d'autorisation décentralisé, les clients et les serveurs de ressources.

#### 5.4.4.1 Jeton d'accès contextuel sécurisé

Les jetons d'accès contextuels générés par le *smart contract* se présentent sous le format *JSON Web Token* (JWT). Lorsque le client déchiffre le jeton, il doit le mettre sous le format COSE avant de le présenter au dispositif. Ce changement de format s'explique par le besoin d'adapter le jeton d'autorisation au dispositif contraint. A la génération d'un jeton d'accès contextuel, le *smart contract* chiffre l'objet JWT en utilisant la clé publique du client auquel le jeton est destiné et le signe. La signature est effectuée avec l'algorithme *Elliptic Curve Digital Signature Algorithm* (ECDSA) [JMV01]. En outre, seul un client autorisé au préalable auprès du *smart contract* peut appeler la fonction de demande de jeton d'accès contextuel. A la réception, le client vérifie la signature du *smart contract* et procède au déchiffrement du jeton. Lorsque le client déchiffre avec succès, il encode le jeton sous le format COSE, ajoute sa propre signature et transfère l'objet COSE contenant le jeton d'accès contextuel chiffré au serveur de ressources. Pour ce faire, il établit une communication sécurisée avec le serveur de ressources.

A la réception du jeton, le dispositif vérifie la signature du client et procède au déchiffrement si la signature correspond. Le dispositif vérifie ensuite la signature du *smart contract* associé au jeton et procède alors à la vérification de la PoP du jeton du client. Si la signature du *smart contract* ne correspond pas, il rejette le jeton.

#### 5.4.4.2 Communication sécurisée

Les communications entre le propriétaire de ressources et le service de sécurité sensible au contexte, aussi bien qu'entre le client et la blockchain, seront effectuées sur des canaux sécurisés par *Transport Layer Security* (TLS). Cela s'explique par les capacités suffisantes des dispositifs qui exécutent les applications du propriétaire de ressources et du client. Par exemple, cela peut être un smartphone, une tablette ou un PC portable.

La communication entre le client et le serveur de ressources peut s'effectuer à travers une multitude de réseaux sous-jacents non sécurisés. Une alternative intéressante à la sécurité de la couche transport est la sécurité des messages de la couche application. En effet, la sécurisation des données applicatives est particulièrement adaptée à l'IoT car sa mise en œuvre nécessite peu de ressources. Ainsi, comme prévu dans le *framework* ACE, nous utilisons *Ephemeral Diffie-Hellman Over COSE* (EDHOC) [SMP19] pour les échanges sécurisés de jetons entre le client et le serveur de ressources. Cela est justifié par les propriétés intéressantes d'EDHOC pour la sécurité des échanges de jeton COSE. En effet, EDHOC permet d'établir un canal de communication sécurisé et adapté pour les dispositifs contraints [SMP19].

### 5.4.5 Révocation dynamique de clients et de jetons

Un autre avantage majeur de notre proposition est la révocation dynamique des jetons d'autorisation ou des entités autorisées par l'utilisateur. Ainsi, notre proposition met en œuvre l'approche centrée sur l'utilisateur.

#### 5.4.5.1 Client

Comme pour l'ajout, la révocation d'un client se fait aisément et dynamiquement selon les préférences de l'utilisateur, définies à travers le service de sécurité sensible au contexte. Lorsque l'utilisateur souhaite retirer les autorisations d'un client, le service de sécurité sensible au contexte appelle la fonction "Révoquer un client" du *smart contract*. Par la suite, la clé publique du client est ajoutée à la liste des autorisations révoquées. Les jetons d'autorisation d'accès contextuels générés pour ce client sont tous invalidés et ne sont plus acceptés par le serveur de ressources.

#### 5.4.5.2 Jetons

Le propriétaire de ressources peut décider de révoquer un jeton d'accès contextuel encore valide attribué à un client. Par exemple, lorsque le contexte de l'utilisateur change (le patient n'est plus en situation critique si on reprend notre exemple du début de ce chapitre, i.e. amélioration de ses signes vitaux), les droits contextuels exceptionnels qui ont été accordés à l'infirmier doivent être retirés. Dans ce cas, l'utilisateur envoie une demande de révocation de jeton au service de sécurité sensible au contexte. Ce service appelle la fonction "Révoquer un jeton d'accès contextuel" du *smart contract*, qui invalidera aussitôt le jeton.

## 5.5 Evaluation des performances

Dans cette section, nous évaluons les performances de la solution proposée. Tout d'abord, nous décrivons l'environnement d'expérimentations et le scénario mis en œuvre pour prouver la faisabilité de notre approche. Par la suite, nous discutons des résultats obtenus. Enfin, nous analysons la sécurité du système proposé.

### 5.5.1 Environnement d'expérimentations

Afin de prouver la faisabilité de la solution proposée, nous avons considéré le scénario e-santé évoqué comme exemple depuis le début de ce chapitre. La solution reste valable pour d'autres applications IoT telles que la maison intelligente, le fitness intelligent, etc. Dans le scénario considéré (e-santé), l'utilisateur possède un glucomètre intelligent (serveur de ressources). Ce glucomètre est basé sur Raspberry Pi Zero w. Le service de sécurité sensible au contexte est hébergé comme une machine virtuelle (également nœud de la blockchain de test) sur une machine hôte avec le système d'exploitation Ubuntu 20.04 LTS. Cette machine hôte est équipée d'un processeur Intel Core i7-6820HQ à deux cœurs de 2,7 GHz et contient 16 Go de mémoire vive.

Pour le serveur d'autorisation décentralisé que nous proposons, nous nous sommes basés sur deux implémentations différentes de la technologie blockchain : Hyperledger Fabric et Ethereum. Le choix de ces deux plateformes est lié à deux éléments importants. Tout d'abord, ces deux plateformes sont régulièrement utilisées dans la littérature pour l'évaluation de solutions basées sur la blockchain. Ensuite, chacune d'entre elles permet l'implémentation et l'évaluation de *smart contracts*, nécessaires à la mise en place de

notre système d'autorisation. Pour les implémentations, nous avons considéré les architectures Blockchain Hyperldeger Fabric (v1.4.10) [Hyp21] et Ethereum (Go Ethereum v1.10.1) [GoE21], des réseaux *permisioned* (l'identité des nœuds est vérifiée) composés de 8 nœuds blockchain impliqués dans le processus de vérification des données.

De même, les *smart contracts* ont été implémentés avec Hyperledger Fabric et Ethereum (respectivement en *Go* et en *Solidity*). Par conséquent, les différences de performances qui pourront être observées dans la section 5.5 sont uniquement dues aux performances mêmes de ces deux implémentations de la blockchain.

Pour mettre en perspective les performances des implémentations réalisées dans l'environnement blockchain, nous avons également utilisé comme base de comparaison une implémentation Python [Hap18] d'un serveur d'autorisation centralisé ACE-OAuth 2 déployé localement. Les *smart contracts* définis et les scripts utilisés dans l'environnement mis en place pour l'évaluation des résultats sont accessibles en ligne : [MS21].

### 5.5.2 Résultats

Dans cette section, nous présentons et discutons les résultats des évaluations effectuées, c'est-à-dire, en matière de sécurité des communications et de gestion décentralisée des autorisations dans la blockchain.

#### 5.5.2.1 Sécurité des communications

Pour démontrer la faisabilité et les avantages de notre proposition en termes de sécurité, de temps d'exécution et de consommation énergétique des dispositifs contraints, nous l'avons comparée avec deux approches de gestion des autorisations basées, respectivement, sur CoAP sans sécurité et sur CoAP DTLS. Pour ce faire, nous avons utilisé la bibliothèque LibCoAP [Ber21] pour l'IoT avec un certificat RSA (Rivest Shamir et Adleman) de 2048 bits.

DTLS est très utilisé pour la mise en œuvre des communications sécurisées basées sur CoAP dans l'IoT. Cependant, il requiert une surcharge de 341 octets dans le meilleur des cas, c'est-à-dire, avec une distribution préalable des clés cryptographiques (clés prépartagées) [Mat19]. En revanche, EDHOC requiert une surcharge maximale de 97 octets avec une distribution préalable des clés. Par conséquent, notre solution basée sur EDHOC requiert beaucoup moins de bande passante que DTLS pour effectuer une communication sécurisée (accès aux ressources sécurisées des dispositifs).

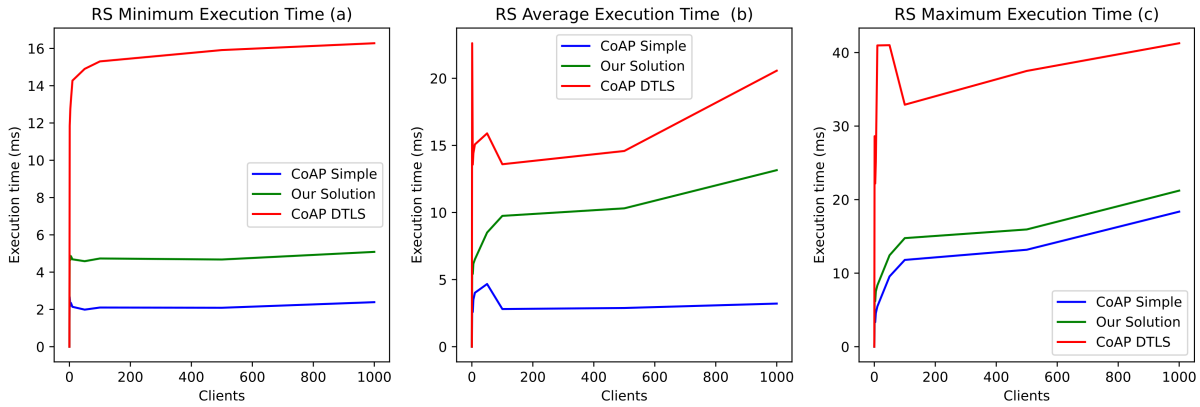



FIGURE 5.7 – Temps de réponse du serveur CoAP du serveur de ressources

Par ailleurs, le traitement temps réel ou quasi temps réel est nécessaire dans les applications d'e-santé, notamment les applications de télésurveillance des signes vitaux. La mise en œuvre des mécanismes d'authentification et de contrôle d'accès ne doivent pas excéder quelques millisecondes. Les échanges initiaux pour l'établissement d'un canal DTLS (*Handshake* DTLS) impactent significativement les performances en augmentant fortement le temps de traitement par le serveur de ressources (dispositif médical), que ce soit dans le meilleur (Fig. 5.7a) ou le pire des cas (14 millisecondes avec 5 requêtes simultanées, 30 millisecondes avec 1000 requêtes simultanées) (Fig. 5.7c). Par contre, en moyenne, l'utilisation d'EDHOC permet de réduire cet impact sur les performances des serveurs de ressources et n'augmente que légèrement la latence, de l'ordre de 5 millisecondes avec le même nombre de requêtes simultanées (Fig 5.7b). La figure 5.7 illustre la comparaison de CoAP sans sécurité (appelé ici CoAP simple), CoAP DTLS et notre solution. Les évaluations montrent également que notre proposition supporte mieux le traitement simultané d'un grand nombre de requêtes d'accès des clients aux ressources protégées en utilisant les jetons d'accès contextuels.

Dans le scénario de l'application considérée, la grande majorité des dispositifs possède une batterie de faible capacité (par exemple des piles AA ou AAA de 1,5 Volts). Par conséquent, la consommation énergétique est un facteur très important qu'il faut prendre en considération lors de la mise en œuvre de mécanismes de sécurité adaptés. Nous avons, ainsi, évalué l'impact énergétique de notre proposition par rapport à l'utilisation de CoAP simple et de CoAP DTLS.

La figure 5.8 illustre l'impact de l'utilisation de ces mécanismes sur la consommation

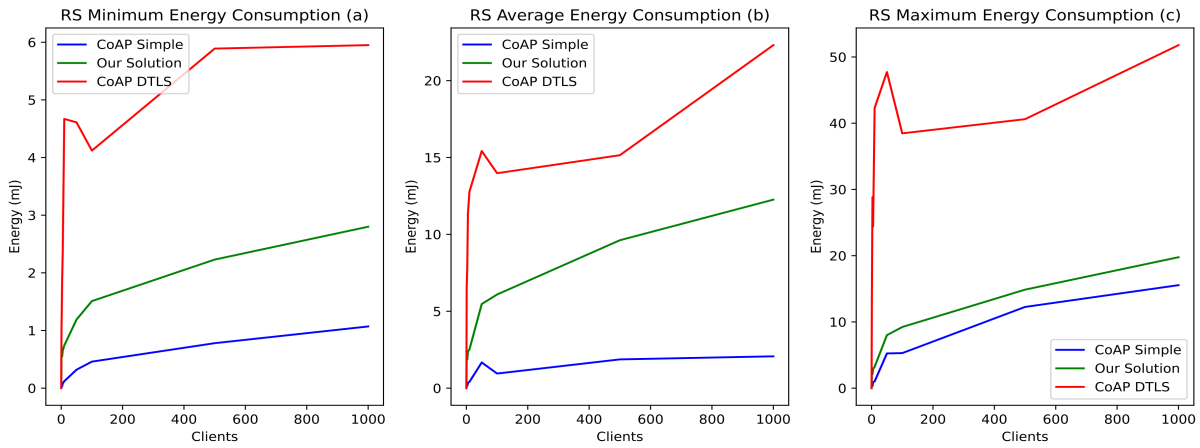



FIGURE 5.8 – Impacts de CoAP simple, CoAP DTLS et d'EDHOC sur la consommation d'énergie du serveur de ressources

énergétique. Les résultats montrent, qu'en moyenne, l'utilisation de notre solution augmente la consommation d'énergie de 4 à 11 mJ par rapport à CoAP simple, tandis que l'utilisation de CoAP DTLS augmente la consommation d'énergie de 15 à 25 mJ suivant le nombre de requêtes simultanées (Fig 5.8). La consommation d'énergie maximale induite par notre solution est légèrement supérieure à celle de CoAP simple d'environ 2 mJ. Cependant, l'impact de l'usage de DTLS sur la consommation d'énergie est fortement supérieur à celui de notre solution ainsi que celui de CoAP simple de plus 20 mJ (Fig 5.8b). Par conséquent, notre proposition a un faible impact sur la consommation énergétique des dispositifs, et est donc adaptée au contexte IoT en garantissant un très bon équilibre entre la sécurité et la consommation énergétique des nœuds.

#### 5.5.2.2 Gestion décentralisée des autorisations dans l'IoT

Dans un second temps, nous avons cherché à démontrer l'intérêt que pourrait présenter l'utilisation d'une approche décentralisée pour le serveur d'autorisation du *framework* ACE. Pour ce faire, nous avons comparé les durées nécessaires à la génération des jetons d'accès contextuels dans une approche centralisée (serveur d'autorisation classique ACE-OAuth2) et décentralisée (Hyperledger Fabric, Ethereum) pour un nombre variable de clients simultanés (200-1000).

Ce que l'on peut constater avec la figure 5.9, c'est que l'approche centralisée clas-

sique offre un niveau de passage à l'échelle important. En effet, quel que soit le nombre de clients simultanés considéré, les temps de génération d'un jeton d'accès contextuel (moyen, minimal, maximal) restent constants (respectivement environ 10, 6 et 14 ms). Ceci est un avantage certain par rapport à un serveur d'autorisation décentralisé basé sur Ethereum qui s'avère plus performant pour un nombre de clients faible (inférieur à 500) (Fig 5.9a.) mais moins performant pour un nombre plus élevé de clients. De même, l'approche centralisée garantit, par rapport à Ethereum, un temps maximal de génération plus faible (gain de plus de 2ms) (Fig 5.9c). Ainsi, l'implémentation actuelle de la Blockchain Ethereum montre ses limites dans ce contexte. Le processus de consensus (Preuve de travail) et la procédure de traitement des requêtes (faible parallélisation) actuellement utilisés par cette technologie entrainent des délais plus importants que l'approche centralisée pour un nombre de clients (i.e. requêtes simultanées) élevé.

En revanche, on peut également constater sur la figure 5.9 que l'implémentation que nous avons réalisée avec Hyperledger garantit des performances bien supérieures, tant par rapport à la solution basée sur Ethereum que par rapport à la solution classique centralisée. En effet, tout comme l'approche classique, la solution basée sur Hyperledger garantit un temps de génération moyen constant, quel que soit le nombre de clients (Fig 5.9b). De plus, ce temps moyen de génération est environ 4 à 5 fois inférieur à celui offert par la solution centralisée. De même, les temps de génération minimaux et maximaux offerts par cette approche sont bien moins importants que ceux garantis par l'approche classique (environ 4 ms de moins en moyenne pour un nombre de clients compris entre 200 et 1000). Par conséquent, cette implémentation basée sur Hyperledger semble démontrer l'intérêt que pourrait présenter l'utilisation d'un serveur d'autorisation décentralisé pour le *framework* ACE.

De plus, on peut également rajouter que cette approche décentralisée pourrait permettre le déploiement de serveurs d'autorisation en bordure du réseau (Edge). Or, comme cela a été démontré par les auteurs de [MRSB18], l'Edge pourrait permettre de réduire la latence de plusieurs millisecondes par rapport à des approches Cloud. Par conséquent, le déploiement d'une solution basée sur Hyperledger, s'appuyant sur l'algorithme de consensus RAFT et un niveau de parallélisation élevé, pourrait permettre d'améliorer les performances à différents niveaux : au niveau du temps de génération des jetons d'accès contextuels lui-même mais également au niveau du temps de transmission de ces jetons.

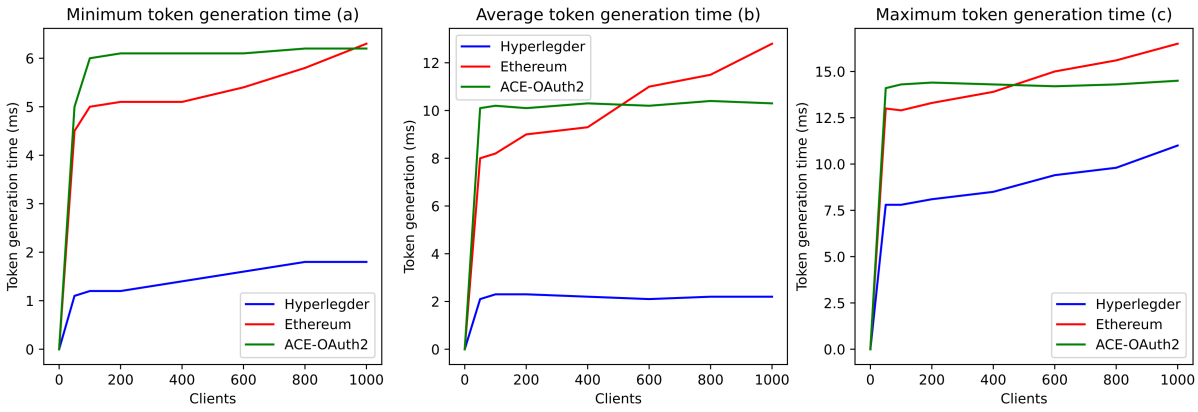



FIGURE 5.9 – Temps de génération des jetons par la blockchain et un serveur ACE-OAuth classique

Enfin, chaque transaction est associée à un coût dans la plupart des implémentations de la blockchain. Ce point important pourrait rendre une mise en œuvre basée sur Hyperledger préférable à une mise en œuvre basée sur Ethereum. En effet, avec Ethereum, le "gas" est utilisé pour mesurer la quantité de travail nécessaire pour accomplir des tâches telles que la vérification des transactions. Toute opération liée à la vérification de l'identité et à la sécurité du système pourrait impliquer des frais de transaction de plusieurs centimes (conversion entre le montant du 'gas' d'Ethereum et le dollar Américain) [LL19]. En revanche, avec Hyperledger, une blockchain privée, le coût d'une transaction est nul.

### 5.5.3 Analyse de la sécurité

Un système de gestion des autorisations sensible au contexte et '*as a service*' pour l'IoT peut être confrontée à plusieurs menaces de sécurité (cf. Section 5.4.1). Dans cette section, nous analysons la sécurité de notre proposition en vérifiant certaines propriétés.

#### 5.5.3.1 Sécurité des jetons d'accès contextuels

Les jetons d'accès contextuels proposés sont sécurisés afin de résister à plusieurs attaques, notamment les attaques d'écoute passive, de modification (forger le contenu d'un jeton valide ou changer le contenu d'un jeton lors de son transfert) et de rejeu de jeton. En effet, les jetons d'accès contextuels sont chiffrés et signés par la blockchain après la génération avant d'être transmis au client via un canal sécurisé par TLS. Le

chiffrement permet d'assurer la confidentialité du jeton lors du transfert. La signature du jeton permet au client d'authentifier l'origine du jeton, c'est-à-dire de s'assurer que l'émetteur du jeton est la blockchain d'autorisation. Elle lui permet également de s'assurer que le jeton n'a pas été modifié lors du transfert. La signature du jeton permet également au serveur de ressources de s'assurer que le jeton a effectivement été émis par le serveur d'autorisation décentralisé. Ainsi, ces mécanismes permettent de pallier les attaques d'écoute passive, de modification de jeton et d'usurpation d'identité lors du transfert de ce dernier de la blockchain au client, d'une part, et du client au serveur de ressources, d'autre part.

Les jetons d'accès contextuels sont également protégés contre le rejeu et l'utilisation dans des contextes invalides. L'identité de contexte d'un jeton d'accès contextuel permet au service de sécurité sensible au contexte de valider le contexte. La preuve de possession (PoP) du jeton permet au service de sécurité sensible au contexte de vérifier l'utilisation du jeton par un client légitime. Lorsqu'un jeton déjà utilisé est présenté au service de sécurité sensible au contexte, il invalide ce jeton. Si le contexte du jeton ne correspond pas au contexte de l'utilisateur, alors la demande d'autorisation sera rejetée.

#### 5.5.3.2 Sécurité du serveur d'autorisation

Les jetons d'accès contextuels proposés sont sécurisés afin de résister à plusieurs attaques, notamment les attaques d'écoute passive, de modification (forger le contenu d'un jeton valide ou changer le contenu d'un jeton lors de son transfert) et de rejeu de jeton. En effet, les jetons d'accès contextuels sont chiffrés et signés par la blockchain après la génération avant d'être transmis au client via un canal sécurisé par TLS. Le chiffrement permet d'assurer la confidentialité du jeton lors du transfert. La signature du jeton permet au client d'authentifier l'origine du jeton, c'est-à-dire de s'assurer que l'émetteur du jeton est la blockchain d'autorisation. Elle lui permet également de s'assurer que le jeton n'a pas été modifié lors du transfert. La signature du jeton permet également au serveur de ressources de s'assurer que le jeton a effectivement été émis par le serveur d'autorisation décentralisé. Ainsi, ces mécanismes permettent de pallier les attaques d'écoute passive, de modification de jeton et d'usurpation d'identité lors du transfert de ce dernier de la blockchain au client, d'une part, et du client au serveur de ressources, d'autre part.

Les jetons d'accès contextuels sont également protégés contre le rejeu et l'utilisation dans des contextes invalides. L'identité de contexte d'un jeton d'accès contextuel permet au service de sécurité sensible au contexte de valider le contexte. La preuve de possession (PoP) du jeton permet au service de sécurité sensible au contexte de vérifier l'utilisation du jeton par un client légitime. Lorsqu'un jeton déjà utilisé est présenté au service de sécurité sensible au contexte, il invalide ce jeton. Si le contexte du jeton ne correspond pas au contexte de l'utilisateur, alors la demande d'autorisation sera rejetée.

#### 5.5.3.3 Protection de la vie privée de l'utilisateur et du client

Dans notre proposition, la décentralisation du serveur d'autorisation le rend plus robuste. En effet, la génération de jetons d'accès (non autorisés) ne sera possible pour l'adversaire qu'à condition qu'il parvienne à compromettre 51% des mineurs du réseau. En prenant l'exemple du Bitcoin, la crypto-monnaie la plus répandue, le nombre de mineurs est aujourd'hui d'un million : compromettre plus de 51% d'un million de machines différentes, sans être détecté, semble être une entreprise complexe. Par conséquent, l'architecture décentralisée offre un niveau de robustesse plus élevé. On peut également rajouter que le niveau de disponibilité permis par cette architecture décentralisée est lui aussi plus élevé. En effet, même si un adversaire prend le contrôle d'un nœud et le rend inaccessible, il sera toujours possible pour les clients de communiquer avec un autre nœud pour accéder au service d'autorisation. Par conséquent, notre proposition est plus sécurisée, plus robuste et apporte plus de confiance dans la gestion des autorisations dans l'IoT.

## 5.6 Conclusion

Dans ce chapitre, nous avons introduit une architecture de gestion décentralisée des autorisations sensibles au contexte '*as a service*' dans l'IoT. Cette architecture permet de résoudre le problème de confiance du serveur d'autorisation du *framework* ACE, par l'introduction de la blockchain. Cette extension permet une gestion décentralisée des autorisations sensibles au contexte dans l'IoT. La conception '*as a service*' de l'architecture proposée permet son intégration à tout service de sécurité sensible au contexte pour l'IoT. Cela permet également sa mise en œuvre dans de nouvelles architectures réseaux telles que l'IoT sur l'Edge avec une très faible latence et de prendre en charge la

mobilité des utilisateurs. Grâce aux jetons d'accès contextuels, les serveurs de ressources ne donneront l'autorisation d'accès à une ressource que si et seulement si le contexte de l'utilisateur l'autorise. Les résultats des évaluations ont prouvé la faisabilité et l'efficacité de l'architecture proposée par rapport au *framework* ACE classique. Les résultats ont également prouvé que la décentralisation du serveur d'autorisation est tout à fait acceptable et a un faible impact sur la latence globale du flux d'autorisation.

Par ailleurs, ce travail présente quelques limites. Premièrement, le serveur de ressources a été testé sur un seul type de dispositif. Il serait intéressant de voir comment la solution se comporterait sur d'autres dispositifs, par exemple sur Arduino MKR Wifi 1010 ou sur ESP8266 Node MCU. Deuxièmement, les expérimentations ont été effectuées en local. Donc, tester l'orchestration et le placement dynamique du service de sécurité sensible au contexte et des nœuds blockchain dans une infrastructure IoT Edge réelle permettrait de confirmer les résultats obtenus.

Dans le chapitre suivant, un cas d'utilisation de l'architecture CASPaaS et de son déploiement (orchestration et placement dynamique) dans une infrastructure Edge sera mis en œuvre afin d'évaluer ses performances.

# Chapitre 6

# Placement de l'architecture CASPaaS dans une infrastructure *Edge computing*

## 6.1 Introduction

Les problèmes de sécurité et de protection de la vie privée dans l'IoT ont fait l'objet de plusieurs travaux. La grande majorité des solutions proposées dans la littérature présente des limites. En effet, ces solutions sont conçues pour une sécurisation basée sur un déploiement dans le *Cloud* [CLR+18, LWL+19]. Ceci s'explique par le fait que la majeure partie des traitements des applications IoT soit effectuée dans le *Cloud*. Cependant, le déploiement des services de sécurité dans un point fixe et distant, notamment le *Cloud*, peut impacter négativement leurs performances ainsi que celles des applications IoT. Ceci s'explique par plusieurs facteurs : la latence importante, le manque de flexibilité, l'overhead important, les pertes de données, etc. Par conséquent, une architecture basée sur le déploiement des services de sécurité dans le *Cloud* n'est pas adaptée pour certaines applications IoT.

La nécessité de rapprocher les services des dispositifs et des utilisateurs pour résoudre ces problèmes a conduit à la proposition de nouvelles architectures réseau : Informatique de bordure/brouillard (Edge/Fog Computing) [KAH+19, LTL+19, RZH+20]. Une infrastructure d'*Edge Computing* permet de pallier aux limites précédemment identifiées en déployant des capacités de calcul (ou de stockage) près des sources/consommateurs de données. Ainsi, il est possible de prendre en charge la mobilité, de décharger ou d'initier certains traitements du *Cloud* vers les nœuds de bordure pour réduire la latence et garantir la QoS. Ces nœuds de bordure peuvent être de différents types : station de base, point d'accès sans fil, unité de bordure de route, etc.

Par ailleurs, la mise en œuvre dans l'*Edge Computing* de services de sécurité et de

protection de la vie privée conçus suivant l'approche monolithique est une tâche très difficile [SKWT17]. En effet, la rigidité de l'architecture d'un tel service empêche son partitionnement et sa distribution pour qu'il puisse être placé dans une infrastructure Edge. En revanche, la conception de services de sécurité et de protection de la vie privée suivant l'approche *as a service* est une solution prometteuse [AALS16, TBME19]. Cette approche permet de concevoir des services distribués sous forme modulaires, flexibles et indépendants qui peuvent être placés de différentes manières dans l'Edge.

Les composants de ce service peuvent être placés comme un seul service sur un seul nœud *Edge* ou de manière distribuée sur plusieurs nœuds *Edge* selon les contraintes de placement, de qualité de service et/ou de qualité d'expérience de l'utilisateur [PRZR19]. Le placement et la migration dynamique des services ou des composants de service permettent le passage à l'échelle nécessaire pour certaines applications IoT (par exemple la e-santé, la réalité augmentée, etc.). En ce sens, nous avons proposé une architecture de sécurité et de protection de la vie privée sensibles au contexte *as a service* pour l'IoT (cf. chapitre 3). En effet, cette architecture a été conçue pour être placée, soit en un seul bloc sur un seul nœud, soit en plusieurs blocs sur plusieurs nœuds selon la disponibilité des ressources. Ainsi, cette architecture bénéficie des avantages de l'approche *as a service* et de l'*Edge Computing* pour assurer la sécurité et la protection de la vie privée des applications IoT tout en tenant compte des exigences ci-dessus énoncées.

Les scénarios de placement de services de sécurité et de protection de la vie privée dans l' *Edge* comprennent des opérations dynamiques exécutées au niveau des nœuds, nombreux et hétérogènes [PRZR19, SDL20, SOE19]. Autrement dit, ces nœuds sont hétérogènes et ont des capacités variées et leurs disponibilités peuvent dépendre de certains facteurs comme la zone géographique, l'heure (moment de la journée), etc. Ces services se traduisent concrètement par des tâches plus ou moins complexes. La mobilité, le maintien d'une bonne QoS (faible latence et bande passante suffisante) et ainsi une bonne qualité d'expérience (QoE) requièrent un placement efficace du service de sécurité et de protection de la vie privée sensibles au contexte dans l'infrastructure Edge. En ce sens, l'accent devra être mis sur la définition de stratégies de placement dynamique des composants de l'architecture *as a service* dédiée à la sécurité et la protection de la vie privée sensibles au contexte pour l'IoT, dans les infrastructures Edge.

Dans ce chapitre, nous proposons une nouvelle stratégie de placement des composants de l'architecture CASPaaS, basée sur les fonctions d'utilité et la logique floue pour

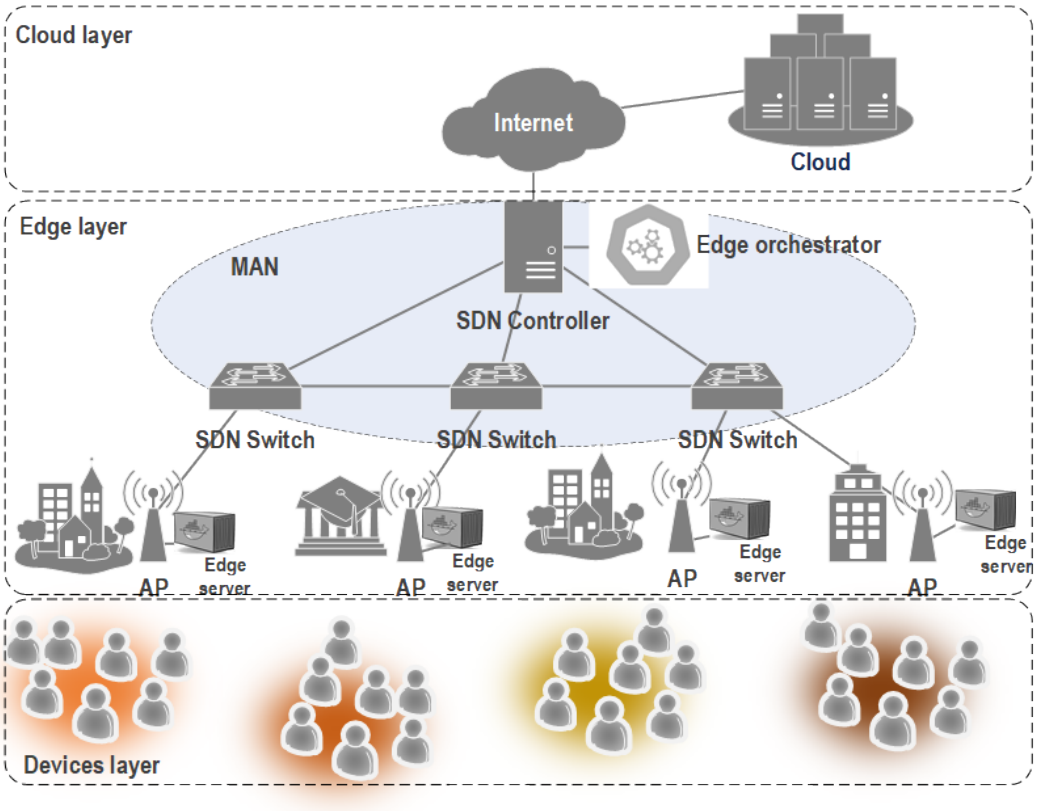



FIGURE 6.1 – Etude de cas proposée avec de multiple nœuds *Edge* dans une ville intelligente

gérer efficacement le placement de ces composants. L'orchestrateur *Edge* décidera du placement d'un service à l'issue d'un processus rapide d'évaluation de : la mobilité de l'utilisateur, la disponibilité des ressources des nœuds, la latence globale des nœuds et le coût de la migration des données. A notre connaissance, une telle stratégie d'orchestration dynamique n'a pas encore été proposée. La fonction d'utilité est un outil simple et efficace pour l'évaluation des problèmes de sélection multicritères dans le domaine des réseaux. La logique floue est un processus décisionnel très performant et caractérisé par une faible complexité. Ainsi, les traitements réalisés par l'orchestrateur *Edge* auront un faible impact sur le temps global de placement des services.

Par ailleurs, pour évaluer, prouver la faisabilité et mettre en évidence les avantages de notre proposition, nous avons considéré, comme étude de cas, la mise en œuvre de l'architecture CASPaaS. Il s'agit d'utilisateurs d'une application d'e-santé vivant dans une ville intelligente dans laquelle une infrastructure *Edge* est déployée à travers les points

d'accès WiFi couvrant la ville. Ces points d'accès WiFi sont interconnectés par les commutateurs SDN (*Software Defined Networking*). Les commutateurs SDN sont contrôlés par un contrôleur SDN. L'orchestrateur Edge, déployé près du contrôleur SDN, orchestrera et gèrera le placement et la migration dynamique des composants de services de sécurité et de protection de la vie privée dans l'Edge. La figure 6.1 illustre l'infrastructure *Edge Computing* considérée.

La suite de ce chapitre est organisée comme suit. La section 6.2 introduit les concepts fondamentaux abordés dans ce chapitre. La section 6.3 compare les travaux connexes sur les stratégies de placement de services dans les infrastructures *Edge Computing*. La section 6.4 décrit la stratégie proposée, son fonctionnement ainsi que les algorithmes qu'elle utilise. La section 6.5 présente le cas d'étude que nous avons considéré. La section 6.6 évalue la stratégie proposée et discute les résultats obtenus. Enfin, la section 6.7 conclut ce chapitre.

## 6.2 Principes fondamentaux

Dans cette section, nous introduisons les principaux concepts abordés dans ce chapitre, à savoir : l'*Edge Computing* ainsi que l'orchestration et le placement de services dans l'Edge.

### 6.2.1 *Edge/Fog Computing*

L'informatique de bordure, ou *Edge Computing* en anglais, est un paradigme qui englobe les concepts informatiques tels que : l'informatique dans le brouillard (*Fog Computing*), l'informatique mobile de bordure multi accès (*Multi Access Mobile Edge Computing*-MEC) et l'informatique basée sur les petits nuages (*Cloudlet-based Computing*). Le but recherché par ces nouveaux concepts informatiques est le rapprochement des services au plus près des utilisateurs, en déplaçant ces services du *Cloud* vers la bordure du réseau. Les services classiquement déployés dans le *Cloud* ont un temps de réaction plus ou moins long (délais liés aux traitements et au transit des données sur le réseau) selon la position géographique des utilisateurs et la disponibilité de la bande passante. L'informatique de bordure permet de résoudre ces problèmes rapidement (la latence liée à ce traitement est très faible). Ces propriétés de l'informatique de bordure sont demandées par les nouvelles applications ubiquitaires telles que les véhicules autonomes, la réalité augmentée ou la e-santé.

L'informatique de bordure, l'informatique dans le brouillard, l'informatique mobile de bordure multi accès et l'informatique basée sur les petits nuages représentent le même concept. L'informatique basée sur les petits nuages (Cloudlet-based Computing) consiste à utiliser des petits centres de données qui fournissent des services aux dispositifs mobiles (smartphone, dispositif portatif, etc.) d'une zone géographique [RZH+19]. L'informatique dans le brouillard (Fog Computing) a été introduite pour la première fois par Cisco en 2015 et est soutenue par l'Open Fog Consortium [BBC+19, Sol15]. Elle est basée sur les équipements de réseaux classiques (routeurs, commutateurs, etc.) dotés de capacités de calcul et de stockage supplémentaires et généralement déployés en zones métropolitaines pour rapprocher le traitement des données des sources (exemple : réseaux de capteurs sans fil).

L'informatique mobile de bordure multi accès (*Multi Access Mobile Edge Computing*-MEC), appelée à l'origine Informatique mobile de bordure, est un terme défini par l'ETSI qui désigne la mise en œuvre d'une infrastructure d'accès mobile dans laquelle les nœuds prennent en charge l'exécution de machines virtuelles d'une architecture de type NFV [Puj19, SVK+16, TSM+23] dans le but d'offrir des services (applications) aux dispositifs mobiles.

Selon la littérature, l'informatique de bordure se différencie de l'informatique dans les nuages par ses caractéristiques suivantes : nœuds hétérogènes, prise en charge de la mobilité, proximité des utilisateurs, distribution géographique dense, faible latence, sensibilité au contexte, etc. [KAH+19, RZH+19, SKWT17].

### 6.2.2 Orchestration et placement de services dans l'*Edge*

La planification et la mise en œuvre des services informatiques virtualisés (routage, inspection de paquets de données, applications IoT, etc.), dans les réseaux logiciels, essentiellement basés sur SDN et NFV s'effectuent avec des outils spécifiques. Ces outils nécessitent beaucoup de coordinations et d'efforts des opérateurs. L'orchestration est un processus qui consiste à planifier, gérer et exécuter le déploiement et/ou la migration de services informatiques virtualisés et d'autres charges de travail informatique (par exemple les tâches relatives à l'apprentissage machine) dans une infrastructure informatique (exemple : informatique dans les nuages, de bordure, etc.). Elle permet aux opérateurs des infrastructures informatiques de coordonner et de gérer facilement la mise en œuvre de services et de charge de travail complexes. Autrement dit, l'orchestra-

tion permet aux opérateurs de savoir où et quand exécuter un service. Le logiciel chargé de la mise en œuvre de l'orchestration est appelé orchestrateur.

L'orchestration de services se décompose en plusieurs sous-tâches : le placement, la migration, la surveillance et la montée en charge de services. Le placement de services consiste à rechercher les emplacements d'exécution favorables pour instancier l'exécution d'un service. Pour ce faire, l'orchestrateur doit évaluer (en ligne ou hors ligne) plusieurs paramètres afin de placer le service. Parmi ces paramètres, nous trouvons l'emplacement d'exécution (*Cloud* ou *Edge*), les contraintes du service, généralement décrites dans le contrat de service (*Service Level Agreement*) et qui concernent notamment la QoS, la latence globale des communications, la mobilité des utilisateurs, etc., pour placer le service. La migration de services est le fait de déplacer un service d'un emplacement (exemple fournisseur) vers un autre emplacement. Dans l'informatique de bordure, celle-ci consiste à déplacer le service d'un nœud vers un autre nœud tout en assurant la continuité du service.

La migration et le placement sont deux termes interchangeables selon le contexte dans lequel ils sont employés. Lors d'une première exécution d'un service dans une infrastructure, on parle de placement. Lorsque le service est en cours d'exécution et s'il doit être déplacé (par exemple au plus proche de l'utilisateur), alors il s'agit de migration. Cependant, lors du placement de service, le terme migration est également employé pour désigner le déplacement des données du service vers un emplacement plus proche du service. La montée en charge de services est un mécanisme qui a pour rôle de dynamiquement augmenter ou diminuer les ressources d'un nœud pour l'exécution d'une tâche (processeur, mémoire vive, etc.) en fonction des besoins de la tâche. La surveillance consiste à vérifier continuellement l'état d'exécution d'un service. L'objectif d'un tel mécanisme est de s'assurer de la disponibilité du service.

Le placement et la migration de services sont les tâches les plus difficiles de l'orchestration dans une infrastructure d'informatique de bordure et de brouillard. En effet, mettre en œuvre une bonne stratégie d'orchestration soulève le problème du placement des services. Ce problème s'explique par l'hétérogénéité des nœuds, très dynamiques, géographiquement éparses et contraints en ressources [SDL20]. Ainsi, le problème du placement des services pose plusieurs questions auxquelles il peut être difficile d'apporter une réponse 'optimale', notamment : comment placer un service sur les nœuds qui ont des ressources disponibles ? comment suivre les déplacements de l'utilisateur ?

comment placer un service sans augmenter la consommation d'énergie des nœuds ? comment placer un service près de l'utilisateur afin d'obtenir la meilleure latence ? etc.

Au final, le problème de placement de services est un problème complexe. Sa résolution nécessite des outils d'analyse et d'optimisation spécifiques. La résolution de ces problèmes a été adressée dans plusieurs travaux [BCFS19, CLF+19, MM20, OZC18]. Les solutions proposées mettent en œuvre des approches différentes et ne considèrent pas le problème de placement sous le même angle. Pour l'analyse, il s'agit ici, d'une part, de décider le type de placement demandé par le service (c'est-à-dire, en ligne ou hors ligne) et, d'autre part, de prendre en charge la dynamicité du service. L'optimisation permet après un processus de calcul de décider sur quel nœud le service doit être placé. Dans ce sens, plusieurs outils d'optimisation existent : Programmation Linéaire des Entiers, Optimisation de Lyapunov, Problème de Décision de Markov, etc. [SDL20]. Finalement, une bonne stratégie d'orchestration doit être basée sur des algorithmes qui utilisent ces méthodes d'optimisation afin de minimiser les échecs de placement des services. Ces méthodes d'optimisation peuvent être associées à des techniques de l'Intelligence Artificielle pour des résultats de placement probants et prenant en charge les caractéristiques de l'*Edge Computing*.

## 6.3 Travaux connexes

Les avantages offerts par les infrastructures *Edge Computing* sont nombreux. Cependant, placer un service et le migrer dans une infrastructure *Edge* pour maintenir une bonne QoS / QoE sont des tâches difficiles. En effet, le service peut être placé sur un nœud surchargé et mener à l'échec du placement. Il peut également être placé dans une portion congestionnée de l'infrastructure. Ce qui peut augmenter la latence. Dans cette section, nous décrivons les travaux qui se sont intéressés aux problèmes du placement des services dans les infrastructures Edge.

Dans [SDL20], Salaht et al. ont réalisé une revue sur le problème du placement des services dans l'Edge. Liu et al. [LTL+19] ont exploré les systèmes et outils d'*Edge Computing* existants utilisés dans la recherche. Il ressort de ces revues que le problème du placement dynamique de services sur les nœuds *Edge* avec la considération des contraintes de ces nœuds est un problème d'optimisation complexe.

La mise en œuvre de projets de ville intelligente repose essentiellement sur la collecte des données des différents capteurs qui doivent être disséminés à travers la ville.

La grande quantité de données collectées doit être envoyée vers des centres de données pour le traitement. C'est dans ce contexte que Velasquez et al. ont proposé une architecture pour le placement de services IoT dans l' *Edge* et le *Cloud* [VACM17]. L'idée principale de cette architecture est la mise en œuvre d'un algorithme d'orchestration qui utilise les informations du réseau et la mobilité des utilisateurs pour effectuer le placement des services. La solution proposée repose également sur un modèle qui optimise le placement de services en utilisant la Programmation Linéaire en nombres Entiers (Integer Linear Problem). Cependant, les auteurs n'ont pas pris en compte la disponibilité des ressources lors du placement. Cela peut conduire à un échec si les nœuds *Edge* sont congestionnés au moment de la tentative de placement.

Le placement dynamique des services dans l' *Edge* en fonction de la mobilité des utilisateurs peut rendre le problème plus complexe. L'imprévisibilité de la mobilité des utilisateurs, l'évolution rapide des conditions du réseau *Edge* et le placement trop rapide ou lent du service en fonction de la mobilité des utilisateurs peuvent mener à la surcharge de certains nœuds de l'infrastructure (par exemple, les nœuds situés dans des endroits avec beaucoup d'utilisateurs), augmenter la latence et dégrader la QoS / QoE. Ouyang et al. [OZC18] ont abordé ces problèmes en proposant une solution de placement dynamique de service à faible coût en fonction de la mobilité des utilisateurs. La contribution principale de ce travail est la prise en compte des coûts de migration à long terme dans les optimisations en temps réel. La solution proposée est basée sur, d'une part, l'utilisation de l'optimisation de Lyanupov (pour la décomposition du problème d'optimisation en sous problèmes) et, d'autre part, l'approximation de Markov (pour la recherche de solution optimale). Ceci permet le placement dynamique (en ligne) sans besoin d'informations futures à priori sur la mobilité des utilisateurs et réduit la latence perçue par les utilisateurs.

Cependant, les coûts opérationnels dus à la migration de service considérés dans ce travail sont l'utilisation importante de la bande passante des liaisons étendues (WAN : *Wide Area Network*) et la consommation d'énergie supplémentaire induite par la migration sur les nœuds Edge. Ainsi, les auteurs n'ont pas considéré le coût de la migration des données et la disponibilité des ressources des nœuds.

Les avancées réalisées dans l'IoT et l'*Edge Computing* ces dernières années ont permis l'émergence de nouveaux services tels que des services cognitifs riches basés sur l'intelligence artificielle. Ces services reposent sur le concept de l'*Edge Cognitive Com-*

*puting* (ECC). Dans ce sens, Chen et al. [CLF+19] ont proposé une architecture fondée sur l'ECC qui permet de fournir des services dynamiques et élastiques de stockage et de calcul dans l'Edge. Ils ont également proposé un mécanisme cognitif dynamique de migration de service basé sur l'ECC. Ce mécanisme prend en considération l'allocation élastique des services de calcul cognitif, la mobilité de l'utilisateur pour le placement et l'ajustement dynamique des services. Le mécanisme de migration proposé est basé sur l'apprentissage par renforcement Q-Learning. L'avantage principal de cette technique est qu'elle optimise les performances et permet une évolution à long terme. Dans Petri et al. [PRZR19], les auteurs ont proposé des stratégies d'orchestration pour le déchargement de traitement du *Cloud* vers l'*Edge*, et vice versa, dans le but de réduire la consommation d'énergie.

Bien que l'*Edge Computing* ait fait l'objet de beaucoup de travaux de recherche, son environnement dynamique (nombreux dispositifs hétérogènes, connectivité intermittente, mobilité des utilisateurs, état du réseau, applications hétérogènes, etc.) rend la mise en œuvre de l'orchestration, la gestion efficace et le passage à l'échelle, une tâche très difficile. Dans ce sens, Sonmez et al. [SOE19] ont traité le problème d'orchestration de la charge de travail (service et tâches à exécuter) dans une infrastructure Edge. L'idée principale de ce travail est de choisir où exécuter les charges de travail en provenance des dispositifs mobiles des utilisateurs en utilisant la logique floue. La solution présente l'avantage de placer les services en fonction des ressources disponibles dans le réseau, de la complexité du service, de la sensibilité au délai du service et de la bande passante disponible sur les liens WAN. Cependant, ce travail présente quelques limites. La mobilité des utilisateurs n'est pas spécifiquement considérée. De même, le coût de la migration des données liées au service lors de l'orchestration n'est pas pris en compte.

Maleki et al. [MM20] ont traité le problème de déchargement des calculs depuis le *Cloud* vers les nœuds *Edge*. Ils ont ainsi proposé une solution de déchargement des calculs en fonction de la mobilité des utilisateurs qui minimise le temps d'exécution des applications mobiles. La solution proposée comprend un algorithme de prédiction qui détermine les caractéristiques des applications mobiles (exemple : temps processeur, mémoire nécessaire, localisation) et décharge (migre) les calculs des utilisateurs. Pour ce faire, les auteurs ont formulé le problème (placement dynamique) en utilisant la programmation des entiers. Néanmoins, les auteurs n'ont pas considéré la disponibilité des ressources dans les nœuds de destination et le coût de la migration des données.

De nombreux travaux se sont intéressés au placement dynamique de services dans l'Edge, mais, à notre connaissance, aucun de ces travaux n'a proposé de solution de placement qui assure une latence faible, prend en charge la mobilité des utilisateurs, la disponibilité des ressources dans les nœuds et les coûts de la migration des données pour le placement dynamique de services dans l'Edge. Ce travail a pour objectif de proposer une stratégie d'orchestration pour le placement dynamique de services dans une infrastructure d'informatique de bordure. Cette stratégie tiendra compte des exigences identifiées : prise en compte de la mobilité des utilisateurs, disponibilité des ressources des nœuds de la zone de destination de l'utilisateur, coût de la migration des données et latence globale perçue par les utilisateurs. Notre approche pour la résolution du problème de placement de service sur les nœuds d'une infrastructure *Edge* s'appuie sur le travail effectué dans [SOE19].

Cependant, notre travail diffère substantiellement de [SOE19] au niveau de la gestion du placement qui fait appel à la logique floue et aux fonctions d'utilité et le complète sur les points suivants :

— le suivi de la mobilité des utilisateurs pour le placement et la migration efficace des services vers l'Edge ;

— le placement des services en plusieurs blocs repartis entre plusieurs nœuds lorsque les ressources disponibles ne permettent pas un placement du service en un bloc.

Les détails de notre proposition sont décrits dans la section suivante (6.4).

## 6.4 Nouvelle stratégie d'orchestration de services *Edge*

Dans cette section, nous détaillons la stratégie d'orchestration proposée et ses différents sous-composants, à savoir les mécanismes de prédiction de la mobilité, de découverte de la disponibilité des ressources, d'évaluation du coût de la migration et de la latence globale. La résolution optimale du problème de placement du service en un bloc, ou en plusieurs blocs distribués, se fait en plusieurs étapes. La section 6.4.5 détaille la résolution du problème en faisant appel aux fonctions d'utilité et à la logique floue.

### 6.4.1 Modèle de mobilité

Comme expliqué auparavant, le cas d'étude considéré concerne une ville intelligente couverte par des points d'accès et formant une infrastructure d'informatique de bordure.

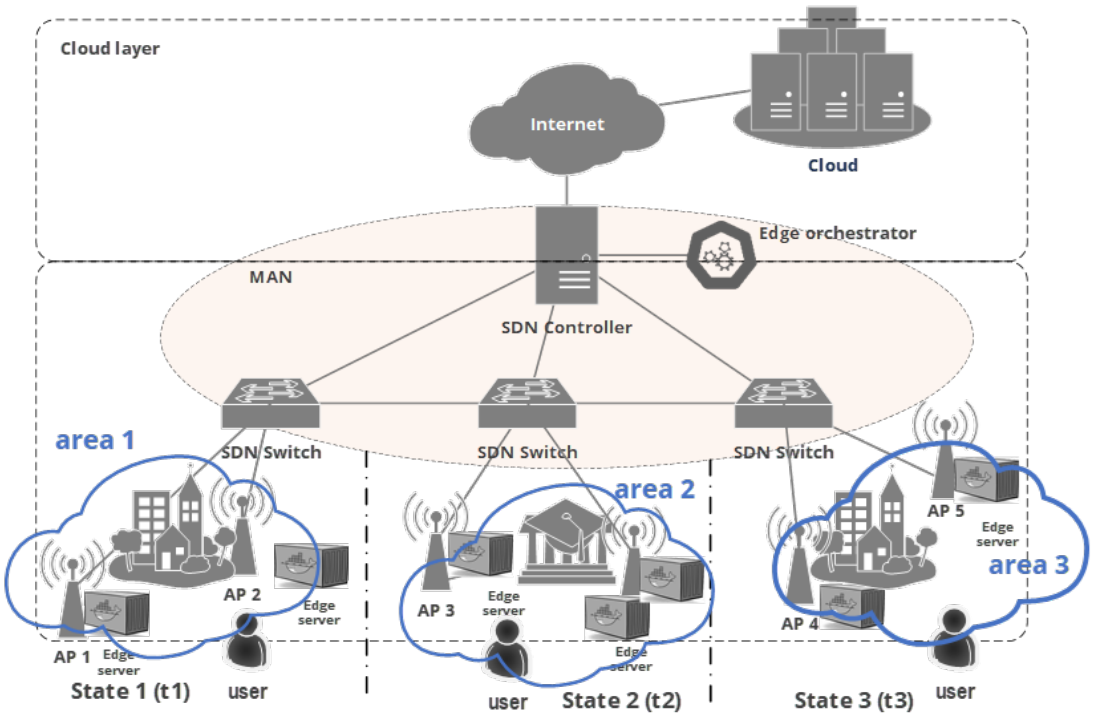



FIGURE 6.2 – Exemple de mobilité d'un utilisateur dans l'infrastructure d'informatique de bordure de la ville intelligente

Suivre la mobilité de l'utilisateur et placer les services en fonction de cette mobilité n'est pas une tâche aisée. En effet, nous supposons que les utilisateurs sont des piétons et suivent le modèle de mobilité nomade [SOE18]. Le placement de service suivant cette mobilité peut être sujet à plusieurs échecs [GBKB20]. Pour réduire le besoin fréquent de placement (et donc de migration de services), et maîtriser la complexité de l'évaluation, l'approche proposée consiste à former des groupes de nœuds de bordure situés dans un périmètre géographique donné et ayant des latences moyennes proches par rapport aux utilisateurs localisés dans ce périmètre. Par exemple, cela peut correspondre aux points d'accès couvrant deux ou trois pâtés de maisons, ayant une latence moyenne de 50 ms. . La répartition en groupes similaires aux cellules de réseaux cellulaires réduit fortement le besoin de placer/migrer les services lors des déplacements des utilisateurs dans les zones concernées. Cela permet de réduire également le taux d'échec des placements. Il est également important de noter que les groupes sont dynamiquement formés par l'orchestrateur selon les critères définis plus haut. A l'instar de [MZC19], nous considérons le problème du placement dynamique de service en fonction de la mobilité de

l'utilisateur comme un processus stochastique. En effet, c'est un problème composé de variables aléatoires possédant une évolution discrète. Le problème est alors reparti en intervalle de temps ($t_0, 1 \ldots i$) et en états ($s_i$). Dans un état si, le service est placé en un bloc sur un seul nœud, ou en plusieurs blocs sur plusieurs nœuds. La figure 6.2 illustre le mouvement d'un utilisateur dans la ville intelligente.

Selon Si et al., il n'y a pas de mouvement totalement aléatoire d'un utilisateur [SWYS10]. Un utilisateur suit généralement une route habituelle (aller au travail, aller déjeuner) et peut exceptionnellement emprunter d'autres trajets. Le modèle de mobilité permet de déterminer à partir d'un mobile connecté à un contrôleur de station de base, le prochain contrôleur de station du secteur auquel le mobile sera connecté [LMQ+14, MCK20a, SWYS10]. Grâce à cette prédiction, l'orchestrateur peut évaluer en amont les autres paramètres nécessaires (disponibilité, latence, etc.) du secteur de destination dans lequel le service sera placé.

Il existe plusieurs approches de prédiction de la mobilité : Markov Decision Process, Hidden Markov Model, Heuristique, Lyapunov optimization method, game theory, etc [SWYS10, SFL17, MZC19]. Le modèle Hidden Markov Model (HMM) permet d'utiliser les informations de trajets passés (modèle de mobilité) des utilisateurs pour prédire avec une grande précision le prochain trajet de l'utilisateur. De plus, HMM est particulièrement adapté à la prédiction de la mobilité car la mobilité est un processus stochastique. Par exemple, dans la figure 6.2, en supposant que le service est placé sur l'AP2 de la zone 1 correspondant à l'état $S_1$. L'orchestrateur *Edge* peut utiliser notre stratégie d'orchestration pour évaluer le placement futur du service sur un nœud de la zone 2 si la prédiction de la mobilité de l'utilisateur indique une forte probabilité que le prochain état est $S_2$, c'est-à-dire, la zone 2. Ainsi, le placement pourra s'effectuer sans aucune perte et n'engendrera pas de latence supplémentaire.

### 6.4.2 Disponibilité des ressources

Après la prédiction de la zone de destination de l'utilisateur, l'étape suivante de notre solution consiste à choisir les nœuds *Edge* candidats sur lesquels le service sera placé. Pour ce faire, les nœuds couverts par la zone de destination seront comparés en matière de disponibilité des ressources. Cette comparaison a pour but de trouver les nœuds candidats ayant les ressources disponibles dépassant un seuil donné pour l'exécution sans échec du service ou de certains blocs du service. Par ressources, nous entendons

l'utilisation du processeur, de la mémoire vive et de la bande passante des nœuds vers le réseau MAN et le réseau WAN. A l'issue de la comparaison (cf. section 6.4.5), la latence perçue par l'utilisateur (on parle en général de temps de réponse côté utilisateur car la latence est un paramètre de QoS au niveau réseau et donc non perçu directement par l'utilisateur) des nœuds candidats sera évaluée.

### 6.4.3 Latence

La surveillance continue de la latence perçue par l'utilisateur permet à une infrastructure *Edge* de détecter les problèmes impactant la latence, de résoudre ces problèmes et de garantir la QoS nécessaire (par exemple une très faible latence pour les services temps réels). Lorsque plusieurs utilisateurs sont présents dans la même zone, il est possible que les bandes passantes des nœuds de la zone vers le réseau MAN soient fortement impactées. Une faible bande passante peut augmenter la latence perçue par les utilisateurs. Un autre phénomène impactant la latence de l'utilisateur est la distance avec le dispositif de bordure, par le point d'accès. Lorsqu'un utilisateur s'éloigne du point d'accès hébergeant son service, la qualité du signal diminue et la latence est impactée. Il est important d'évaluer périodiquement la latence perçue par l'utilisateur vis-à-vis du réseau.

A l'issue de l'évaluation de la latence des nœuds de la zone de destination, les nœuds candidats les plus intéressants (faible latence) seront choisis. Cependant, si aucun des nœuds candidats ne présente une latence conforme aux exigences de QoS du service, alors le placement dans les nœuds de cette zone n'a plus d'intérêt. L'alternative sera alors de trouver un compromis entre garder le service sur les nœuds de la zone actuelle de l'utilisateur et trouver d'autres nœuds dans le voisinage direct de la zone de destination de l'utilisateur. Cette dernière recherche peut rendre plus complexe la recherche de solutions optimales de placement.

### 6.4.4 Coût de la migration des données

La bande passante disponible d'un nœud de bordure est un élément important de la performance du placement/migration des services. En effet, une bande passante disponible élevée permet une migration des données avec un temps de migration négligeable. Par contre, une bande passante faible a pour résultat une migration des données avec un temps de migration long. L'évaluation de la bande passante lors du placement permet

de déterminer les nœuds de destination permettant de réaliser un placement optimal (temps de migration des données). C'est pourquoi, la bande passante disponible est un critère important de la stratégie proposée lors du choix dynamique des nœuds de bordure.

### 6.4.5 Stratégie d'orchestration basée sur les fonctions d'utilité et la logique floue

Dans cette section, nous présentons les éléments essentiels de notre stratégie d'orchestration basée sur les fonctions d'utilité et la logique floue. Comme expliqué précédemment, l'idée de cette stratégie est de prédire la zone de destination de l'utilisateur en fonction de sa mobilité et d'orchestrer en conséquence le placement du service de l'utilisateur de manière transparente et sans interruption. Pour mettre en œuvre une stratégie réaliste, en plus des informations provenant de l'infrastructure Edge, les informations provenant des dispositifs des utilisateurs (par exemple, la latence perçue) et des SLA (par exemple, la latence maximale autorisée et le débit minimal) sont nécessaires.

La stratégie proposée est basée sur deux algorithmes : l'algorithme principal de la stratégie de placement (Algorithme 1) et l'algorithme de placement dynamique des blocs d'un service en fonction des conditions globales (réseau, nœuds, etc.) (Algorithme 2). Nous utilisons les fonctions d'utilité dans le premier algorithme pour évaluer et classer les zones de destinations probables selon le niveau de satisfaction des contraintes du service. L'algorithme de placement dynamique des blocs d'un service utilise également les fonctions d'utilité pour classer les nœuds *Edge* d'une zone (sélectionnée) en fonction de leurs capacités à recevoir les blocs du service à placer. Par la suite, nous utilisons la logique floue dans ce second algorithme afin de déterminer si les blocs du service peuvent être placés sur un ou plusieurs nœuds selon les capacités de ces nœuds et des contraintes du service (cf. section 6.4.5.2). Les fonctions d'utilité proposées pour la sélection de zones et de nœuds *Edge* sont détaillées dans la section 6.4.5.2.

#### 6.4.5.1 Algorithmes

Pour éviter le besoin erratique de placement des services suivant les mouvements des utilisateurs, notre stratégie découpe l'infrastructure *Edge* en plusieurs zones. La formation des zones (c.f. section 6.4.1) est l'étape initiale. L'algorithme principal (Algorithme 1) utilise les paramètres des différents services spécifiés dans le SLA correspon-

dant au service. Les paramètres lus sont : la latence maximale autorisée (*SeuilL*) et les blocs du service (*SBlocks*). L'orchestrateur *Edge* surveille l'exécution des services et les mouvements des dispositifs des utilisateurs dans les zones. L'algorithme 1 est exécuté lorsque la latence du service (*SL*) se rapproche de la latence maximale autorisée du service (*SeuilL*). L'augmentation de la latence perçue par l'utilisateur peut avoir plusieurs causes. Premièrement, l'utilisateur est en mouvement et s'éloigne du nœud exécutant le service. Deuxièmement, l'utilisateur se trouve dans une zone attractive congestionnée par de nombreux autres utilisateurs.

Lorsque l'utilisateur est en mouvement avec une vitesse supérieure à un seuil fixé (*SeuilMobilite*), la prédiction de la zone de destination de l'utilisateur est effectuée pour l'intervalle de temps suivant $t_1$. Plusieurs zones de destination probables peuvent être détectées en plus de la zone actuelle de l'utilisateur, si ce dernier s'y trouvera encore. Les fonctions d'utilité proposées pour la sélection de zones évaluent les zones de destination probables selon leurs conditions et effectuent un classement de ces zones par ordre décroissant, c'est-à-dire, de la zone ayant le meilleur score à la zone ayant le plus petit score. Les zones sont ensuite successivement sélectionnées en fonction de leur score et l'algorithme de placement dynamique des blocs d'un service est exécuté pour placer le service. Lorsqu'aucun nœud de la zone sélectionnée n'est capable de recevoir l'ensemble des blocs du service, ces blocs seront placés sur plusieurs nœuds de la zone en fonction de leurs caractéristiques et des besoins du service à placer. Ces actions seront effectuées jusqu'au placement du service. Dans le cas où l'utilisateur ne serait pas en mobilité, un ou plusieurs autres nœuds de la zone actuelle de l'utilisateur moins chargé et répondant aux exigences identifiées est sélectionné. Le tableau 6.1 liste les abréviations utilisées dans les deux algorithmes.

L'algorithme de placement dynamique de blocs d'un service (Algorithme 2) reçoit en entrées les informations sur les nœuds de la zone sélectionnée NA et l'ensemble des blocs du service. Une deuxième fonction d'utilité est alors utilisée pour calculer et classer les scores des nœuds de la zone NA. Pour chaque nœud classé, en commençant par le nœud ayant le meilleur score, les ressources disponibles du nœud sont évaluées par le SLF proposé. Suivant le résultat du SLF (par exemple très bon, bon ou moyen) les blocs du service sont, soit tous placés sur le même nœud, soit placés successivement sur les meilleurs nœuds de la zone NA en fonction des ressources disponibles.

Spécifiquement, deux fonctions *PlacerSBlocksMax* et *PlacerSBlocksMin* utilisent la

Tableau 6.1 – Liste des abréviations utilisées dans les algorithmes

| **Abréviations** | **Descriptions** |
|---|---|
| $t_0$ | le temps à l'instant 0 |
| $t_1$ | l'instant$t_0 + 1$ |
| $t_2$ | l'instant $t_0 + 2$ |
| $N$ | Nœud |
| $CN$ | Nœud choisi pour le placement |
| $A$ | Zone |
| $P$ | Périmètre (Largeur x Longueur) |
| $AA$ | Zone actuelle de l'utilisateur |
| $NA$ | Zone de destination de l'utilisateur sélectionnée |
| $PA$ | Zones potentielles de destination de l'utilisateur |
| $RankedNA$ | Zones voisines classées pouvant recevoir le service |
| $AvgL$ | Latence moyenne des nœuds |
| $AvgLNA$ | Latence moyenne de la prochaine zone NA |
| $AvgBwNA$ | Bande passante moyenne de la prochaine zone NA |
| $NAUsers$ | Nombre d'utilisateurs présents dans la prochaine zone NA |
| $SeuilL$ | Seuil de latence maximale au-delà duquel le service ne peut plus fonctionner correctement. Provient du SLA. |
| $SL$ | Latence du service |
| $SeuilMobilite$ | Vitesse minimale à partir de laquelle un utilisateur est considéré en mouvement. |
| $SBlocks$ | L'ensemble des blocs du service S |
| $SBlocksLoad$ | Charge requise par les blocs du service S |
| $SBw$ | Bande passante nécessaire du service S |
| $RankedN$ | Les nœuds classés par la fonction d'utilité |
| $Uv$ | Vitesse de mouvement de l'utilisateur (Velocity) |
| $NCPULoad$ | Charge processeur du nœud N |
| $NMemLoad$ | Utilisation de la mémoire du nœud N |
| $NBw$ | Bande passante disponible vers le MAN du nœud N |
| $SLF$ | Système de logique floue |

sortie du SLF et les informations spécifiques des blocs du service (charge mémoire et processeur, bande passante, etc.) pour la réalisation du placement. La fonction *PlacerSBlocksMax* est appelée lorsque le placement de l'ensemble des blocs restants du service sur un seul nœud est possible selon le résultat du SLF (Très bon ou Bon). Dans le cas où le résultat du SLF est Très bon ou Bon pour le nœud de la zone considérée avec les meilleures capacités (première itération), tous les blocs du service seront placés sur ce nœud. La fonction *PlacerSBlocksMin* est appelée, le cas échéant, pour placer quelques blocs du service selon les ressources disponibles sur le nœud. En effet, si le résultat du SLF est Moyen sur un nœud, alors ce nœud n'est pas en mesure de recevoir tous les blocs restants du service. Le résultat du placement est renvoyé vers l'algorithme 1. Si le placement réussit, alors l'algorithme 1 met fin au parcours des zones potentielles et renvoie comme valeur le succès du placement.

Il est possible qu'aucun nœud à l'instant t1 ne satisfasse les conditions pour un placement optimal dans les deux cas (en mobilité ou non). Ainsi, lorsque le placement est infructueux dans l'Edge, le service est alors placé dans le *Cloud*.

#### 6.4.5.2 Fonction d'utilité proposées

Une fonction d'utilité a pour rôle de déterminer le niveau de satisfaction obtenu par la prise d'une décision, par exemple la consommation d'un service [BTTK14]. Dans les réseaux, les critères de satisfaction peuvent être de différents types : optimiser la QoS, prendre en charge les préférences de l'utilisateur, etc. Plusieurs travaux ont utilisé les fonctions d'utilité dans la gestion de la mobilité, la sélection du meilleur réseau sans fil parmi plusieurs réseaux disponibles et le contrôle de la congestion dans les réseaux [BTTK14, CLST20, LY18].

Nous proposons, ici, d'utiliser les fonctions d'utilité pour classer et sélectionner les zones de destination probables de l'utilisateur. Nous mettons également en œuvre les fonctions d'utilité pour classer et sélectionner les nœuds *Edge* de la zone sélectionnée sur lesquels les blocs du service peuvent être placés.

L'utilité, c'est-à-dire, le degré de satisfaction du placement d'un service dans une infrastructure *Edge* dépend des contraintes du service. Un service de réalité augmentée a besoin d'un débit stable tandis qu'un service de e-santé ou de communication pour réseau véhiculaire a besoin d'une latence très faible. Dans ce travail, nous proposons

**Algorithme 1 :** Algorithme de la stratégie de placement proposée

**Initialisation**
*Formation des nœuds $Ni$ en zone $Ak$ selon le périmètre $P$ et $AvgL$ proches.*
*Lire le $SeuilL$ à partir du $SLA$ de $S$*
*Lire le nombre de blocs du service $SBlocks$*
*'a partir du $SLA$ de $S$*
*Déterminer la latence moyenne $AvgL$ de la zone actuelle $AA$ de l'utilisateur*
*Déterminer le mouvement de l'utilisateur $Uv$*
**Fin initialisation**
**Input :** $SL, SBw, AA$
**Output :** Le service est bien placé ou non
**if** $SL \geq SeuilL$ **then**
  **if** $Uv > SeuilMobilit$ **then**
    $PA \leftarrow PredireProchainesZones(t_1)$
    **for** $NAinPA$ **do**
      $AvgLNA \leftarrow$*Déterminer la latence de la moyenne zone $NA$*
      $AvgBwNA \leftarrow$*Déterminer la bande passante moyenne de la zone $NA$*
      $NAUsers \leftarrow$*Lire le nombre d'utilisateurs présents dans la zone $NA$*
      *Ajouter résultat* $FonctionUtilite(AvgLNA, AvgBwNA, NAUsers)$ *à* $RankedNA$
    $Reussite \leftarrow Faux$
    $n \leftarrow 1$
    **while** *Reussite = Faux* **do**
      $NA \leftarrow RankedPA(n)$
      $Reussite \leftarrow ResultatAlgorihtme2(NA, SBw, SBlocks)$
    **if** $Reussite = Vrai$ **then**
      $ReussitePlacementEdge \leftarrow Vrai$
    **else**
      $PlacerleservicedansleCloud$
      $ReussitePlacementEdge \leftarrow Faux$
    **return** *ReussitePlacementEdge*
  **else**
    *Exécuter l'algorithme de placement de blocs avec $AA$*
**return** *ReussitePlacementEdge*

**Algorithme 2 :** Agorithme de placement dynamique de blocs d'un service

**Input :** NA,SBw,SBlocks
**Output :** Le service est bien placé ou non
$RankedN \leftarrow FonctionUtilit(NA)$ *//Calculer le score des nœuds N et les classer*;
$i \leftarrow 1$;
**while** $Card(SBlocks) > 0$ *et* $i \leq Card(RankedN)$ **do**
  $N \leftarrow RankedN(i)$ ;
  $NCPULoad(N) \leftarrow$ *Charge CPU de* $N$;
  $NMemLoad(N) \leftarrow$ *Utilisation mémoire de* $N$ ;
  $ResultatSLF \leftarrow SLF(NCPULoad, NMemLoad, NBw, SBlocksLoad, SBw)$;
  **if** $ResultatSLF = Trsbon$ *ou* $ResultatSLF = Bon$ **then**
    $blocksplacs \leftarrow PlacerSBlocksMax(ResultatSLF, SBlocks, SBw, N)$;
  **else**
    **if** $ResultatSLF = Moyen$ **then**
      $blocksplacs \leftarrow PlacerSBlocksMin(ResultatSLF, SBlocks, SBw, N)$;
  *Retirer blocsplacés de* $SBlocks$;
  $i \leftarrow i + 1$;
**if** $Card(SBlocks = 0$ **then**
  $Reussite \leftarrow Vrai$
**else**
  $Reussite \leftarrow Faux$
**return** *Reussite*

d'utiliser les fonctions d'utilité pour évaluer l'utilité des zones de destination potentielles de l'utilisateur et l'utilité des nœuds de la zone de destination sélectionnée pour le placement du service en fonction des contraintes spécifiées dans le SLA correspondant au service. Pour ce faire, les attributs des zones et des nœuds doivent être convertis par les fonctions d'utilité en valeurs d'utilité.

Les valeurs d'utilité globales sont, par la suite, calculées pour chaque zone de destination potentielle et chaque nœud de la zone de destination sélectionnée. Les zones et les nœuds sont alors classés du meilleur, c'est-à-dire, celui ayant la plus grande valeur d'utilité (le meilleur score), au pire (le plus petit score) (cf. Algorithme 1 et 2). Les avantages de la mise en œuvre des fonctions d'utilité dans l'aide à la prise de décision sont multiples : simplicité de la mise en œuvre, efficacité et complexité réduite. Ainsi, notre stratégie de placement dynamique basée sur l'utilisation combinée des fonctions d'utilité et de la logique floue a un très faible coût. Ce qui implique un impact négligeable

sur la durée globale nécessaire au placement des services (cf. section 6.6).

Selon plusieurs travaux existant dans la littérature sur les décisions de sélection dans le domaine des réseaux, on distingue plusieurs fonctions d'utilité [KRA14, LY18, RLA17]. Cette distinction se fait en fonction des critères de sélection : *sigmoïdale, linéaire, logarithmique, exponentielle,* etc. Le choix d'une fonction d'utilité s'effectue en fonction des caractéristiques de la fonction et des critères à prendre en compte pour respecter les exigences du service (délai, bande passante, gigue, etc.). Par exemple, un service exigeant une bande passante fixe peut-être représenté par une fonction *sigmoïdale*. En effet, celle-ci est utilisée pour représenter des contraintes dont les valeurs peuvent fluctuer entre 0 et 1 de façon importante au cours du temps, contrairement à une fonction par pas.

*Classement des zones de destination potentielle de l'utilisateur*

Dans notre proposition, les critères retenus pour le classement des zones de destination potentielles de l'utilisateur (zones prédites) sont : la latence moyenne, la bande passante moyenne disponible et le nombre d'utilisateurs. Le choix d'une fonction d'utilité adéquate en fonction des attributs est primordial. Chaque attribut à évaluer peut avoir sa propre fonction d'utilité qui peut être différente des fonctions des autres attributs. En effet, cela dépend des caractéristiques des attributs à évaluer [WK21]. Sur la base des références précédemment mentionnées, nous proposons d'utiliser la fonction d'utilité *sigmoïdale* (équation 6.1) pour évaluer l'utilité de la latence et de la bande passante et la fonction d'utilité *linéaire* (équation 6.2) pour évaluer le nombre d'utilisateurs. Pour une fonction d'utilité sigmoïdale, le paramètre a représente le seuil et le paramètre b permet de régler la pente de la fonction. Pour la fonction d'utilité *linéaire*, $g(x)$, elle représente les préférences subjectives.

$$u(x) = \frac{(x/a)^b}{1 + (x/a)^b} \tag{6.1}$$

$$u(x) = g(x) + h \tag{6.2}$$

L'évaluation du score de classement des zones calcule la fonction d'utilité de chaque attribut d'une zone puis en déduit la valeur de l'utilité globale de la zone. Il est important de noter que la valeur de l'utilité globale est une valeur qui varie dans l'intervalle [0,1]. Une grande valeur, par exemple 0,8, signifie une utilité plus grande. Une petite valeur,

par exemple 0,2, signifie une petite utilité. Les fonctions d'utilité de la latence et de la bande passante sont définies par les équations 6.3 et 6.4. Nous supposons dans ce chapitre que la latence minimale $l_{min}$, par exemple 10 ms, est fournie par le SLA du service et que la satisfaction diminue lorsque la latence globale de la zone augmente. Nous supposons que la bande passante requise $bw_{min}$ est aussi fournie par le SLA.

$$u(l) = \frac{(l/l_{min})^4}{1 + (l/l_{min})^4} \tag{6.3}$$

$$u(bw) = \frac{(bw/bw_{min})^{10}}{1 + (bw/bw_{min})^{10}} \tag{6.4}$$

Une nouveauté de notre proposition est la considération des statistiques instantanées sur le nombre d'utilisateurs de service *Edge* (nouveaux entrants/sortants, temps moyen passé dans la zone) des zones de destination potentielles de l'utilisateur à l'instant $t_1$, en plus de la latence et de la bande passante. Cela permet de gérer l'environnement fortement dynamique du *Edge* et de détecter une zone potentielle qui est en voie de saturation et de l'éviter. Par exemple, une zone peut avoir une disponibilité acceptable à l'instant $t_0$, mais elle peut être rapidement saturée à cause de son attractivité à certaines heures de la journée. En effet, selon les expérimentations réalisées dans [GBKB20, SOE18] avec des données (dataset) de mobilité réelles, un nombre important d'utilisateurs piétons (modèle de mobilité piétonne) peut rapidement congestionner les nœuds *Edge* et causer un taux d'échec important de placement des services. Comme annoncé, la fonction d'utilité proposée pour le nombre d'utilisateurs est une fonction linéaire. Nous supposons que la satisfaction du service diminue fortement lorsque ce nombre atteint 70% de la capacité prévue $c$. Ce taux peut être ajusté en fonction des résultats obtenus à l'issue des expérimentations. La fonction d'utilité est définie par l'équation 6.5.

$$u(n) = 1 + \frac{1}{70} * n \tag{6.5}$$

La valeur de l'utilité globale d'une zone est calculée en associant un poids à la valeur d'utilité de chaque attribut, en fonction de leur importance. Suivant le scénario considéré dans notre proposition, la latence a le plus grand poids, suivi de la bande passante. Le nombre d'utilisateurs a un poids faible comparé à la latence et à la bande passante. Ceci peut s'expliquer par le fait que le nombre d'utilisateurs impacte peu le placement du service si la latence globale de la zone et la bande passante sont satisfaisants. L'équa-

tion 6.6 définit le score d'utilité d'une zone.

$$S_{NA} = (\frac{(l/l_{min})^4}{1+(l/l_{min})^4}) \times (0.5) \times (\frac{(bw/bw_{min})^{10}}{1+(bw/bw_{min})^{10}}) \times (0.3) \times 1 + \frac{1}{70} * n \times (0.2) \quad (6.6)$$

Classement des nœuds d'une zone potentielle sélectionnée Similairement au classement des zones de destination potentielles de l'utilisateur, nous utilisons les fonctions d'utilité pour classer les nœuds d'une zone de placement potentielle, du meilleur au plus mauvais, selon leurs capacités à recevoir des blocs du service considéré. Les fonctions d'utilité proposées sont *sigmoïdales* pour la charge processeur, la mémoire et la bande passante disponibles. Elles sont définies dans les équations 6.7, 6.8 et 6.9.

$$u(c) = \frac{(c/c_{min})^4}{1+(c/c_{min})^4} \quad (6.7)$$

$$u(m) = \frac{(m/m_{min})^4}{1+(m/m_{min})^4} \quad (6.8)$$

$$u(b) = \frac{(b/b_{min})^4}{1+(b/b_{min})^4} \quad (6.9)$$

La valeur d'utilité globale d'un nœud est calculée en priorisant la charge du processeur et l'utilisation de la mémoire. Ainsi, ces deux attributs ont un poids légèrement supérieur à l'attribut de la bande passante. Ceci peut s'expliquer par l'importance de la disponibilité des ressources de calcul (processeur et mémoire) dans la réussite du placement par rapport à la bande passante. L'équation 6.10 définit le score d'utilité d'un nœud de la zone sélectionnée.

$$S_N = (\frac{(c/c_{min})^4}{1+(c/c_{min})^4}) \times (0.4) \times (\frac{(m/m_{min})^4}{1+(m/m_{min})^4}) \times (0.4) \times (\frac{(b/b_{min})^4}{1+(b/b_{min})^4}) \times (0.3) \quad (6.10)$$

#### 6.4.5.3 Système de logique floue proposé

La logique floue est une technique de l'intelligence artificielle qui a pour but de modéliser des systèmes informatiques en utilisant le raisonnement logique humain impliquant des propositions imprécises et qualitatives comme "à peu près bon, très bon, etc." [CFN17, SCKS21]. Elle permet de prendre en compte des valeurs intermédiaires entre *Oui* et *Non* dans la prise de décision. En effet, dans les systèmes logiques classiques la décision prise a pour valeur *Oui/Non*. Ainsi, la logique floue nous permet dans ce travail de raisonner et de prendre des décisions selon les performances des nœuds Edge. Elle a

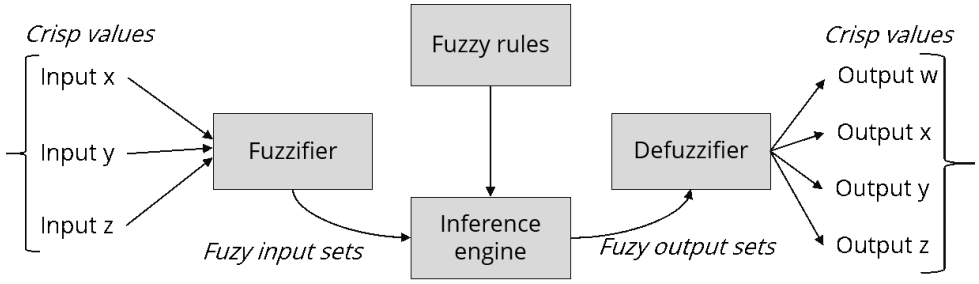



FIGURE 6.3 – Vue synoptique d'un système de logique floue

prouvé son utilité dans de nombreux travaux de nombreux domaines de l'informatique tels que l'aide à la décision, la sécurité, le réseau, etc. [APK19, SOE19, SCKS21].

La figure 6.3 illustre la vue synoptique d'un Système de Logique Floue (SLF). Les valeurs réelles fournies en entrées à un SLF sont fuzzifiées, c'est-à-dire qu'elles sont transformées en variables floues, encore appelées variables linguistiques ou ensembles flous. Chaque variable linguistique est composée de termes linguistiques et chaque terme à un degré d'appartenance, autrement dit, une quantification du terme. Les degrés d'appartenance des variables linguistiques sont définis par les fonctions d'appartenance [KKS20].

La stratégie de placement proposée dans ce travail met en œuvre la logique floue pour évaluer les conditions instantanées des nœuds afin de décider si le placement de l'ensemble des blocs du service sur un nœud est possible. Si ce n'est pas le cas, c'est-à-dire, si le nœud *Edge* sélectionné ne dispose pas de suffisamment de ressources pour héberger l'ensemble des blocs du service et respecter les exigences de QoS/QoE, alors l'algorithme de placement dynamique des blocs choisira les meilleurs nœuds *Edge* capables de recevoir les différents blocs du service.

Le SLF proposé reçoit comme entrées : la charge du processeur du nœud (CPULoad), l'utilisation de la mémoire (MemLoad) et la bande passante disponible du nœud vers le réseau MAN, la charge nécessaire pour l'exécution des blocs (SBlocksLoad) et la bande passante requise par le service (SBw). Ces paramètres permettent de connaitre l'état instantané d'un nœud et ainsi de savoir si le nœud est capable de recevoir les blocs du service à placer. Les variables linguistiques pour ces entrées sont *c* pour la charge du processeur, *m* pour l'utilisation de la mémoire et ab pour la bande passante disponible du nœud, *bl* pour la charge nécessaire des blocs et *sb* pour la bande passante requise du service. Les termes linguistiques des variables floues *c, m, bl* sont : *faible* (Lo) pour une

utilisation faible des ressources, *moyen* (M) pour une utilisation normale des ressources (par exemple entre 20 et 60%) et *chargé* (C) pour une forte utilisation des ressources (par exemple au-delà de 75%). En ce qui concerne la bande passante disponible du nœud, nous utilisons les termes linguistiques suivants : *saturé* (S), *faible* (Lo), *moyen* (M) et *élevé* (Hi).

La bande passante disponible *ab* d'un nœud est saturée (S) lorsqu'elle est inférieure à 50 Mbits/s, faible (Lo) si elle est comprise entre 50 et 100 Mbits/s. Elle est moyenne (M) lorsqu'elle est comprise entre 100 Mbits/s et 500 Mbits/s. Elle est élevée (Hi) lorsqu'elle est supérieure à 500 Mbits/s [CCT20]. Les fonctions d'appartenances utilisées sont de formes sigmoïdales pour *faible*, Z pour élevé et triangulaires pour *moyen*. Les sorties du SLF sont ***très bon, bon, moyen, faible, très faible***. Les expressions des fonctions d'appartenances des différents termes des variables linguistiques proposées sont décrites dans les équations 6.11 à 6.20 et illustrées par la figure 6.4.

Les sorties de la fuzzification sont fournies au moteur d'inférence pour la prise de décision. Le moteur d'inférence utilise une base d'inférence composée de règles de type SI-SINON (cf. section 4.3.3.2.2). Il existe plusieurs méthodes d'inférence en logique floue, dont notamment la méthode Mamdani et la méthode Sugneo. La base de règles d'inférence du SLF proposé pour la sélection du meilleur nœud d'une zone pour le placement des blocs d'un service comprend 243 règles. Le tableau 6.2 présente quelques règles d'inférence du SLF.

$$\mu_{Lo}(c) = \begin{cases} 1, & c \leq 30\% \\ \frac{30\%-c}{30\%-10\%}, & 10\% < c \leq 30\% \\ 0, & c \geq 30\% \end{cases} \tag{6.11}$$

$$\mu_{M}(c) = \begin{cases} 0, & c \leq 30\% \\ \frac{c-55\%}{55\%-30\%}, & 30\% < c \leq 55\% \\ \frac{75\%-c}{75\%-55\%}, & 55\% < c \leq 75\% \\ 0, & c \geq 75\% \end{cases} \tag{6.12}$$

$$\mu_{Hi}(c) = \begin{cases} 0, & c \leq 70\% \\ \frac{c-80\%}{95\%-85\%}, & 85\% < c \leq 95\% \\ 1, & c > 95\% \end{cases} \tag{6.13}$$

$$\mu_{Lo}(m) = \begin{cases} 1, & m \leq 30\% \\ \frac{30\%-m}{30\%-10\%}, & 10\% < m \leq 30\% \\ 0, & m \geq 30\% \end{cases} \tag{6.14}$$

$$\mu_{M}(m) = \begin{cases} 0, & m \leq 30\% \\ \frac{m-55\%}{55\%-30\%}, & 30\% < m \leq 55\% \\ \frac{75\%-m}{75\%-55\%}, & 55\% < m \leq 75\% \\ 0, & m \geq 75\% \end{cases} \tag{6.15}$$

$$\mu_{Hi}(m) = \begin{cases} 0, & m \leq 70\% \\ \frac{m-80\%}{95\%-85\%}, & 85\% < m \leq 95\% \\ 1, & m > 95\% \end{cases} \tag{6.16}$$

$$\mu_{S}(ab) = \begin{cases} 1, & ab \leq 30mb/s \\ \frac{50mb/s-ab}{50mb/s-30mb/s}, & 30mb/s < ab \leq 50mb/s \\ 0, & ab \geq 50mb/s \end{cases} \tag{6.17}$$

$$\mu_{Lo}(ab) = \begin{cases} 0, & ab \leq 40mb/s \\ \frac{ab-80mb/s}{80mb/s-40mb/s}, & 40mb/s < ab \leq 80mb/s \\ \frac{120mb/s-ab}{120mb/s-80mb/s}, & 80mb/s < ab \leq 120mb/s \\ 0, & ab \geq 120mb/s \end{cases} \tag{6.18}$$

$$\mu_{M}(ab) = \begin{cases} 0, & ab \leq 100mb/s \\ \frac{ab-240mb/s}{240mb/s-100mb/s}, & 100mb/s < ab \leq 240mb/s \\ \frac{440mb/s-ab}{440mb/s-240mb/s}, & 240mb/s < ab \leq 440mb/s \\ 0, & ab \geq 440mb/s \end{cases} \tag{6.19}$$

$$\mu_{Hi}(ab) = \begin{cases} 0, & ab \leq 420mb/s \\ \frac{ab-540mb/s}{540mb/s-420mb/s}, & 420mb/s < ab \leq 540mb/s \\ 1, & ab \geq 540mb/s \end{cases} \tag{6.20}$$

$$\mu_{Lo}(bl) = \begin{cases} 1, & bl \leq 30\% \\ \frac{30\%-bl}{30\%-10\%}, & 10\% < bl \leq 30\% \\ 0, & bl \geq 30\% \end{cases} \tag{6.21}$$

$$\mu_{M}(bl) = \begin{cases} 0, & bl \leq 30\% \\ \frac{bl-55\%}{55\%-30\%}, & 30\% < bl \leq 55\% \\ \frac{75\%-bl}{75\%-55\%}, & 55\% < bl \leq 75\% \\ 0, & bl \geq 75\% \end{cases} \tag{6.22}$$

$$\mu_{Hi}(bl) = \begin{cases} 0, & bl \leq 70\% \\ \frac{bl-80\%}{95\%-85\%}, & 85\% < bl \leq 95\% \\ 1, & bl > 95\% \end{cases} \tag{6.23}$$

$$\mu_{Lo}(sb) = \begin{cases} 0, & sb \leq 40mb/s \\ \frac{sb-80mb/s}{80mb/s-40mb/s}, & 40mb/s < sb \leq 80mb/s \\ \frac{120mb/s-sb}{120mb/s-80mb/s}, & 80mb/s < sb \leq 120mb/s \\ 0, & sb \geq 120mb/s \end{cases} \tag{6.24}$$

$$\mu_{M}(sb) = \begin{cases} 0, & sb \leq 100mb/s \\ \frac{sb-240mb/s}{240mb/s-100mb/s}, & 100mb/s < sb \leq 240mb/s \\ \frac{440mb/s-sb}{440mb/s-240mb/s}, & 240mb/s < sb \leq 440mb/s \\ 0, & sb \geq 440mb/s \end{cases} \tag{6.25}$$

$$\mu_{Hi}(sb) = \begin{cases} 0, & sb \leq 420mb/s \\ \frac{sb-540mb/s}{540mb/s-420mb/s}, & 420mb/s < sb \leq 540mb/s \\ 1, & sb \geq 540mb/s \end{cases} \tag{6.26}$$

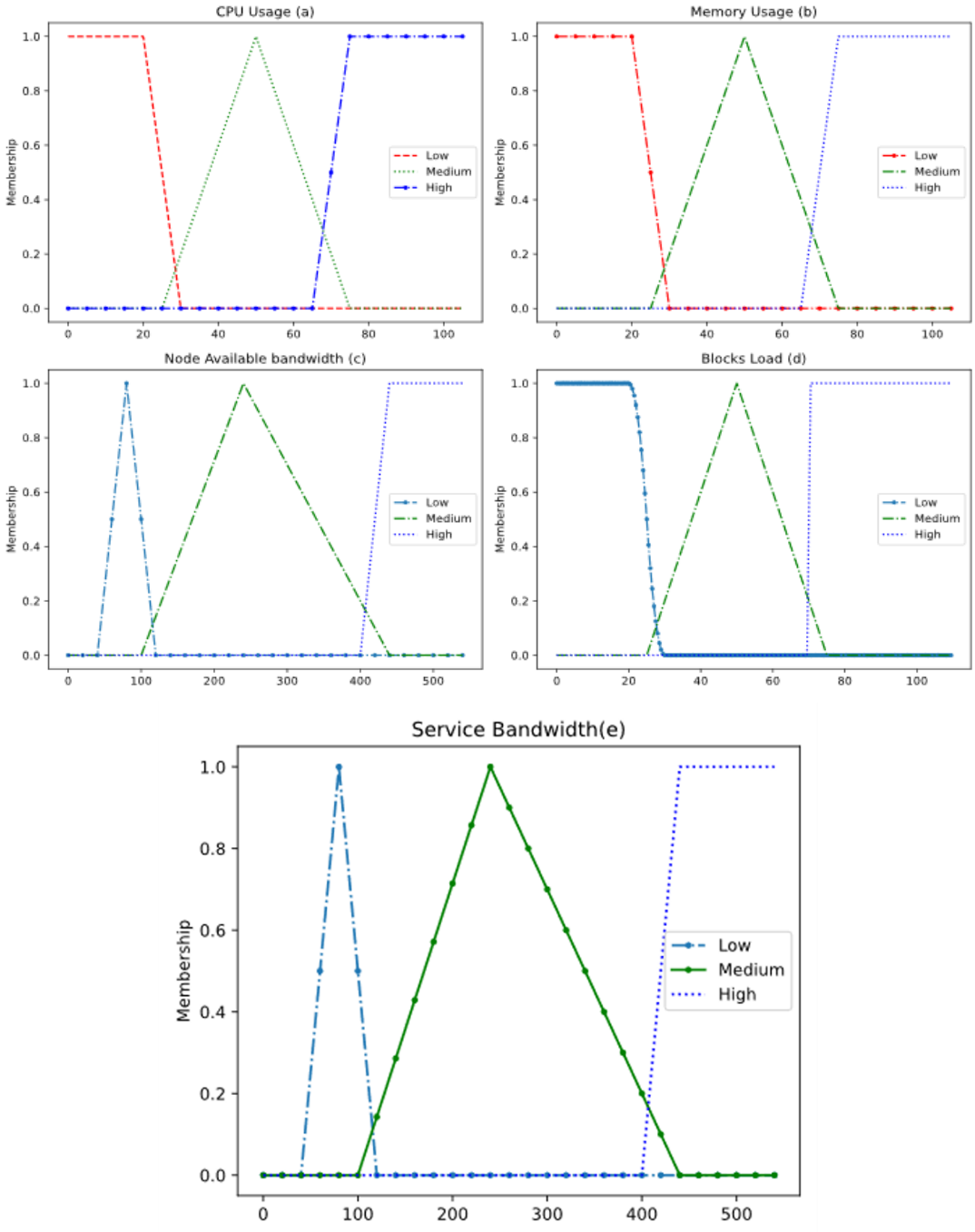



FIGURE 6.4 – Fonctions d'appartenance du CPU (a), de la mémoire (b) et de la bande passante disponible du nœud (c), de la charge nécessaire pour l'exécution des blocs (d) et de la bande passante requise par le service (e)

Tableau 6.2 – Quelques règles d'inférence pour la sélection du meilleur nœud d'une zone pour le placement des blocs d'un service

| Index | CPU | Mémoire | B. P. | Charge B.S | B. P. Service | Qualité N. |
|---|---|---|---|---|---|---|
| 1 | Low | Low | Low | Low | Low | Good |
| 2 | Low | Low | Medium | Low | Low | Good |
| 3 | Low | Low | High | Medium | High | Good |
| 4 | Medium | Medium | Saturated | Medium | Low | Medium |
| 5 | Medium | Medium | Low | High | Medium | Low |
| 6 | Medium | Medium | Medium | Medium | Medium | Medium |
| 7 | High | High | Low | High | High | Low |
| 8 | High | High | Saturated | High | High Very Low | |
| 9 | High | High | Medium | Medium | Medium | Medium |
| 10 | High | Medium | Low | Medium | Medium | Low |
| 11 | Medium | High | High | Low | Low | Medium |
| 12 | Low | Low High | High | Medium | Good | |
| 13 | Low | Medium | Medium | Medium | Low | Good |
| 14 | Medium | Low | High | Medium | High | Good |

Légende : B. P. : Bande passante, B. S. :Blocs du Service, N. :Nœud

La défuzzification est la dernière étape du SLF (Figure 7.3). Elle consiste à convertir les valeurs floues produites par le moteur d'inférence en valeurs nettes. Il existe plusieurs méthodes de défuzzification : appartenance maximale, centroïde, moyenne pondérée, etc. Notre SLF utilise la défuzzification par la méthode du centroïde, qui consiste à chercher le centre de gravité du polygone obtenu par le moteur d'inférence [JVB92] (équation 6.27).

$$COG = \frac{\sum_{x=a}^{b} \mu_A(x) \cdot x}{\sum_{x=a}^{b} \mu_A(x)} \tag{6.27}$$

## 6.5 Cas d'étude de CASPaaS

Dans cette section, nous décrirons les différents blocs fonctionnels de l'architecture de sécurité et de protection de la vie privée sensibles au contexte (CASPaaS). Ensuite, nous expliquerons comment ils peuvent être placés dans une infrastructure *Edge* en utilisant la stratégie de placement que nous proposons.

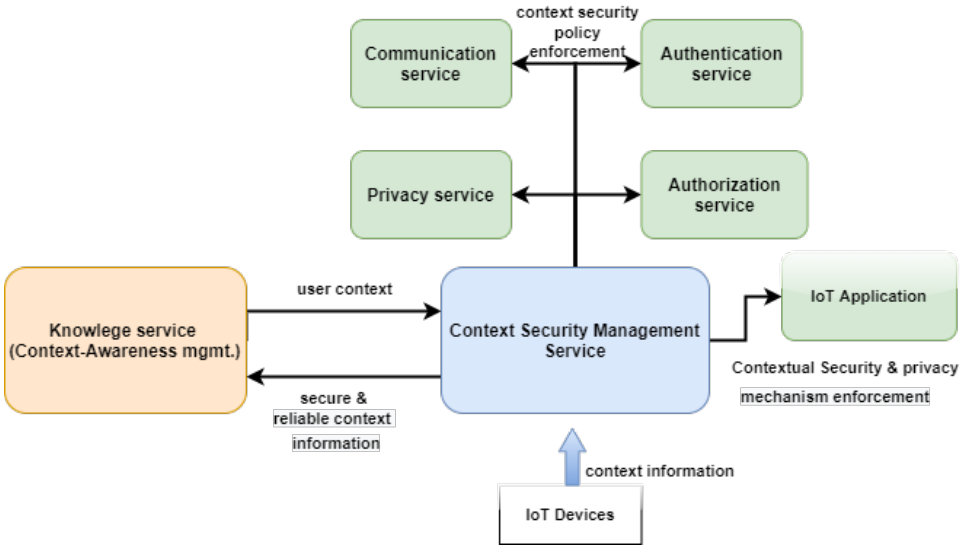



FIGURE 6.5 – Blocs de CASPaaS

### 6.5.1 Description du cas d'étude

Grâce à l'approche '*as a service*', l'architecture de sécurité et de protection de la vie privée sensibles au contexte dans l'IoT (CASPaaS) peut s'intégrer facilement dans les nouvelles architectures réseaux telles que l'*Edge Computing* et la 5G, et peut dynamiquement composer en fonction du contexte les mécanismes de sécurité permettant de répondre aux menaces et risques du moment et liés au contexte actuel. Ce qui permettra d'assurer la sécurité et la protection de la vie privée des utilisateurs partout et dans quasiment toutes les circonstances. La figure 6.5 illustre un découpage de l'architecture CASPaaS en 6 blocs de services. Chaque service peut être exécuté sur un nœud différent.

Nous notons qu'il est important de mettre en œuvre des mécanismes de communication entre les différents blocs du service pour qu'ils accomplissent convenablement leurs tâches et respectent les contraintes de QoS/QoE. On distingue plusieurs mécanismes de communication inter-services tels qu'Apache Kafka, ActiveMQ, Java Messaging Service, RabbitMQ, etc. Apache Kafka est le mécanisme d'échange distribué le plus utilisé dans les communications inter-services. C'est un mécanisme distribué évolutif et à très faible latence. Le choix d'Apache Kafka est basé sur ses propriétés d'échange à très faible latence et à débit élevé.

### 6.5.2 API Exposées par CASPaaS

Une interface de programmation d'application, ou *Application programming interface* (API) en anglais, est un moyen utilisé par deux ou plusieurs systèmes (ou applications) distincts pour communiquer entre eux. Une API offre un langage, une spécification de comment les échanges doivent s'effectuer entre les systèmes. Dans l'industrie, on distingue deux principaux types d'API : *Simple Object Access Protocol* (SOAP) et *Representational State Transfer* (REST) [HDMP+21].

Comme précédemment annoncé, CASPaaS a pour objectif de gérer les fonctions de sécurité et de protection de la vie privée des applications IoT des utilisateurs. Pour ce faire, cette architecture expose un certain nombre d'API pour les applications IoT cibles telles que celles de la ville intelligente comme la maison intelligente, la e-santé, etc. Dans ce chapitre, nous avons considéré la mise en œuvre de CASPaaS pour une application d'e-santé. L'application d'e-santé utilise plusieurs dispositifs tels que le glucomètre intelligent pour la prise régulière du taux de glucose dans le sang, la montre intelligente pour les relevés de la fréquence cardiaque, du nombre de pas effectués et un smartphone pour l'interface cliente de l'utilisateur et la géolocalisation, etc. Les informations contextuelles de l'utilisateur sont envoyées au CASPaaS pour la prise de décision d'adaptation du niveau de sécurité de l'application d'e-santé.

Dans le scénario proposé, les API exposées par CASPaaS sont les suivantes : authentification et autorisation, communication sécurisée et protection de la vie privée sensibles au contexte. Ces API exposées au niveau de la couche application permettent aux développeurs d'application IoT de décharger les fonctions de sécurité de leurs applications sur CASPaaS. Par exemple, l'API d'authentification et d'autorisation sensibles au contexte permet de prendre en charge toutes les opérations d'authentification à plusieurs facteurs et adaptative au contexte de l'utilisateur. Elle permettra également la gestion des autorisations sensibles au contexte.

## 6.6 Évaluation

Dans cette section, nous évaluons les performances de la stratégie d'orchestration proposée afin de démontrer son efficacité. Nous effectuerons également une comparaison de cette stratégie avec deux des stratégies les plus intéressantes parmi celles que nous avons décrites dans la section 6.2.

Tableau 6.3 – Paramètres de simulation

| Paramètres | Valeurs |
|---|---|
| Nombre de datacenters | 14 |
| Nombre de serveurs *Edge* par datacenter | 1 |
| Nombre de VM par serveur Edge/Cloud | 8/4 |
| Nombre de cœurs par CPU de VM Edge/Cloud | 2/4 |
| Mémoire RAM de VM par serveur Edge/Cloud | 2 Go/10 Go |
| Temps de simulation | 33 minutes |
| Bande passante WAN/WLAN | Empirique |
| Bande passante MAN | Modèle MMPP/M/1 [SOE19] |
| Délai de propagation LAN | 5 millisecondes |
| Modèle de mobilité | Nomade [SOE19] |
| Nombre maximal de dispositifs mobiles | 2000 |
| Probabilité de sélection d'un type de localisation | Equiprobable |
| Nombre de type de localisation (1/2/3) | 2/4/8 |
| Temps de séjour moyen par type de localisation | 2/5/8 minutes |

### 6.6.1 Configuration de l'expérimentation

Pour mener à bien cette expérimentation, nous avons utilisé comme simulateur EdgeCloudSim [SOE18]. Cette plateforme de simulation permet de réaliser des modèles réalistes d'environnement d'*Edge Computing*. Basé sur CloudSim [CRB+11], EdgeCloudSim a une architecture modulaire lui permettant de prendre en charge plusieurs éléments importants dans la simulation réaliste d'infrastructure *Edge Computing* : la prise en charge de modèles dynamiques de communication de réseaux locaux sans fil ou de réseaux étendus, la génération de charges réalistes, les modèles de mobilité, etc.

Les paramètres de simulation utilisés dans l'expérimentation sont listés dans le tableau 6.3. L'ensemble des simulations ont été réalisées sur un serveur configuré avec Ubuntu 20.04.3 LTS, Intel (R) Xeon (R) Gold 6230R 2.10 GHz et possédant 2 sockets, et une RAM de 32Go.

Afin de réaliser une simulation de la stratégie de placement de service proposée et adaptée au scénario considéré dans ce chapitre, nous avons considéré le déploiement uniformément reparti des serveurs *Edge* dans la ville. Chaque zone de couverture est gérée par un datacenter de proximité et constitué d'au moins un serveur Edge. Chaque serveur *Edge* comporte jusqu'à huit machines virtuelles ayant chacun la capacité d'exé-

Tableau 6.4 – Blocs du service CASPaaS utilisés dans les simulations

| **Propriétés** | **Gest. Conn.** | **Gest. Séc. Cont.** | **Vie privée** | **AuthZ** | **Auth** | **Comm. Séc.** |
|---|---|---|---|---|---|---|
| Utilisation (%) | 30 | 30 | 10 | 10 | 10 | 10 |
| Période d'activité/ inactivité | 90 / 10 | 45 / 90 | 60 / 120 | 30 / 45 | 30 / 45 | 30 / 45 |
| Données (Ko)(Envoi / Réception) | 100 / 1200 | 20 / 1250 | 2500 / 200 | 25 / 1000 | 25 / 500 | 25 / 1000 |
| Taille (Tâche) (GI) | 20000 | 30000 | 45000 | 15000 | 15000 | 15000 |
| Utilisation de la VM Edge | 5 | 5 | 5 | 5 | 5 | 5 |

*Légende : Gest. Conn. :Gestion de la connaissance, Gest. Séc. Cont. : Gestion de la Sécurité Contexutelle, AuthZ : Authorisation, Auth : Authentification, Comm. Séc. :Communication sécurisée*

cuter plusieurs dizaines de tâches. Nous avons utilisé le modèle de mobilité Nomade [SOE18], dans lequel un utilisateur mobile se déplace d'un endroit à un autre et y passe un certain temps selon l'attractivité de l'endroit (section 7.4.1). Les modèles de bande passante utilisés pour les communications de réseaux local, métropolitain et étendu sont basés sur les modèles MMPP/M/1 et empirique proposés dans [SOE19].

Par ailleurs, EdgeCloudSim permet de simuler le placement de diverses configurations d'application, notamment des applications de réalité augmentée, de streaming vidéo, d'e-santé, etc. Les blocs du service de sécurité et de protection de la vie privée sensibles au contexte pour l'IoT (CASPaaS cf. chapitre 3) considéré dans cette simulation et leurs caractéristiques importantes sont listés dans le tableau 6.4. L'utilisation représente le pourcentage des utilisateurs qui utilisent l'application parmi l'ensemble des utilisateurs. L'envoi et la réception de données illustrent les données (en Kilo octets) envoyées et reçues par les différentes applications (blocs du service). La taille est utilisée pour déterminer les ressources processeur requises pour chaque application. Elle est exprimée en Giga Instructions (GI). Enfin l'utilisation de la machine virtuelle (VM) *Edge* représente la charge entrainée par l'exécution de l'application sur la machine virtuelle.

### 6.6.2 Résultats de la simulation

Les résultats de la simulation réalisée avec EdgeCloudSim consistent en plusieurs indicateurs de performance des systèmes *Edge Computing* et utilisés dans la littérature. Il s'agit du taux d'échec de placement, du temps moyen d'exécution des services, de l'utilisation moyenne des ressources des VM et des causes d'échecs de placement. Nous avons comparé la stratégie d'orchestration proposée aux stratégies suivantes : la stratégie basée sur l'utilisation des ressources et la stratégie basée sur *Fuzzy Competitor* [SOE18].

***Taux d'échec de placement :*** Une infrastructure *Edge Computing* a pour but d'exécuter des applications IoT et mobiles et de décharger les infrastructures *Cloud* Computing de plusieurs tâches. Cela permet de gagner de la bande passante et de réduire fortement la latence. Les applications ainsi exécutées sont fortement dynamiques, c'est-à-dire, leur lancement (placement/migration) ou arrêt peut s'effectuer à tout moment, par exemple selon la mobilité des utilisateurs. Le taux d'échec de placement de tâche est un indicateur de performance très important à prendre en compte dans la conception d'une stratégie d'orchestration dans l'*Edge Computing*. La figure 6.6 illustre les taux d'échec des algorithmes comparés. Lorsque l'infrastructure est moins chargée, avec un nombre d'utilisateurs compris entre 200 et 400, les stratégies d'orchestration comparées ont un taux d'échec de placement assez proche. Néanmoins, la stratégie basée sur l'utilisation des ressources présente un taux d'échec légèrement en dessous des autres stratégies. Cela s'explique par la mise en œuvre de l'algorithme de recherche du nœud *Edge* le moins chargé (*Least Load Algorithm*) lors du placement des services.

Lorsque l'infrastructure est moyennement chargée, avec un nombre d'utilisateurs compris entre 600 et 1200, la stratégie d'orchestration basée sur la fonction d'utilité et la logique floue présente un meilleur taux d'échec de placement comparé aux deux autres stratégies. En effet, la stratégie proposée a un taux d'échec moyen relativement faible comparé à la stratégie *Fuzzy Competitor* et à la stratégie basée sur l'utilisation des ressources. Le taux d'échec relativement faible de la stratégie proposée par rapport aux autres stratégies quand la charge du réseau est élevée s'explique par la considération de l'état des connexions réseau et des ressources des serveurs *Edge* de la zone de destination avant le placement des services.

***Temps moyen d'exécution :*** Le temps moyen d'exécution est un critère d'évaluation

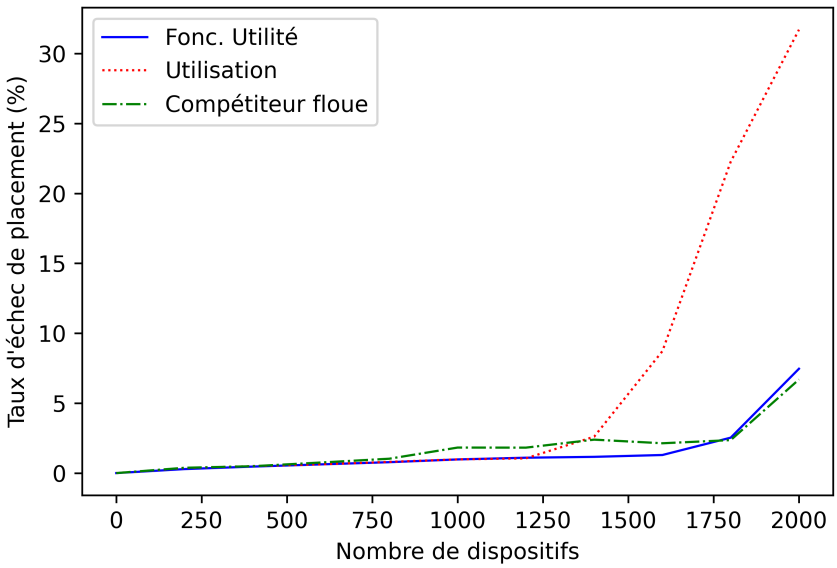



FIGURE 6.6 – Taux d'échec du placement de service en fonction de la charge de l'infrastructure

important des performances d'une infrastructure *Edge Computing*. Il englobe le temps d'exécution du service et le temps des communications réseau. Il permet aux opérateurs d'infrastructure *Edge Computing* de jauger le niveau de satisfaction des clients. En effet, l'exécution complète et sans interruption des applications des clients mobiles est un critère d'évaluation de la qualité de beaucoup de clients. La figure 6.7 représente les temps moyens d'exécution des services dans l' *Edge* selon les algorithmes de placement comparés.

Notre stratégie d'orchestration réalise un meilleur résultat par rapport aux stratégies basées sur l'utilisation des ressources et sur *Fuzzy Competitor*. Ce bon résultat s'explique par l'utilisation de la fonction d'utilité pour le classement des nœuds et la logique floue pour la prise de meilleures décisions de placement. Ainsi, le système étant faiblement chargé, l'absence de congestion réduit fortement l'impact des délais des communications sur le temps d'exécution moyen des applications. Lorsque le système est chargé, le temps d'exécution ainsi que le délai des communications augmentent. Cette augmentation du temps moyen d'exécution est due à la congestion des communications réseau, notamment les communications sur réseau étendu WAN et au manque de disponibilité des ressources de calcul sur les nœuds Edge. Toutefois, notre stratégie d'orchestration

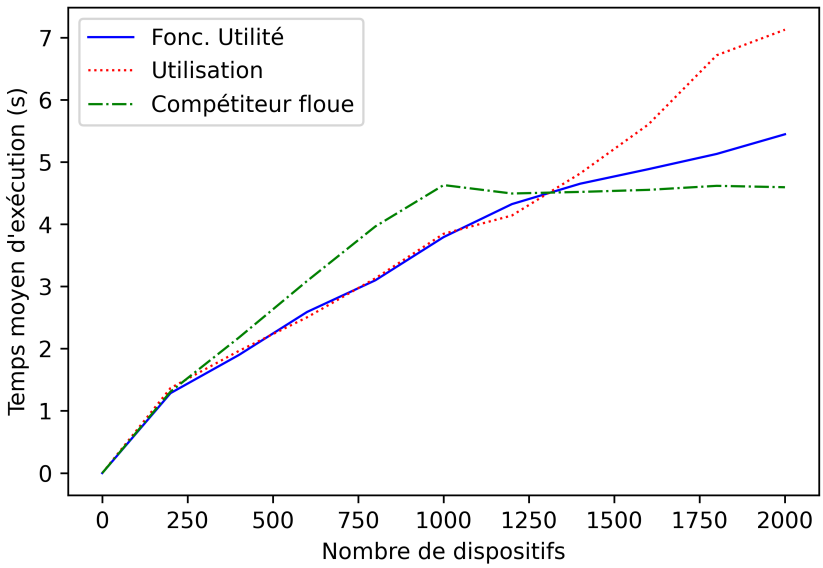
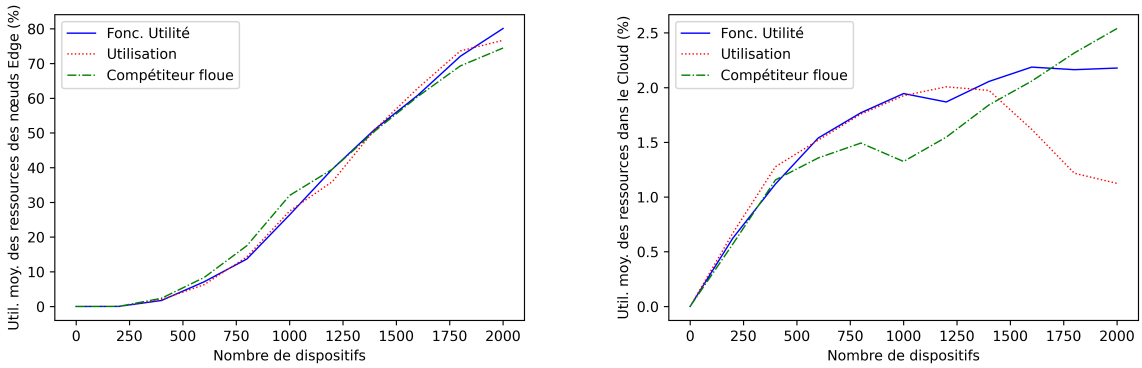



FIGURE 6.7 – Temps moyen d'exécution des services dans l'Edge

produit un temps d'exécution moyen inférieur par rapport aux autres stratégies.

Par ailleurs, lorsque les services sont exécutés sur les nœuds Edge, le temps total d'exécution comprend le temps d'exécution des services et le délai des communications réseau (LAN et MAN). Dans ce cas, le délai des communications WAN est négligeable. La majeure partie du temps total d'exécution est le temps d'exécution des services. Ainsi, plus la charge du système augmente, plus le temps d'exécution des services sur les nœuds augmentent. Par contre, la latence augmente beaucoup plus lentement par rapport au temps d'exécution des services. Les VM configurées dans le *Cloud* pour l'exécution des services sont puissantes, par conséquent le temps d'exécution des services dans le *Cloud* est faible. La majeure partie du temps total d'exécution correspond au délai WAN. Ce délai augmente proportionnellement au nombre d'utilisateurs. Utilisation moyenne des ressources des nœuds *Edge* : L'utilisation moyenne des ressources de calcul des nœuds *Edge* telle que l'utilisation moyenne du processeur indique si les ressources sont utilisées de manière efficace ou gaspillées. La figure 6.8a illustre l'utilisation moyenne des ressources des nœuds *Edge* lors de la mise en œuvre des trois stratégies d'orchestration évaluées. Lorsque le niveau de charge du système est faible, par exemple avec 400 utilisateurs mobiles, la stratégie proposée présente une utilisation plus ou moins faible des ressources des nœuds par rapport aux deux autres stratégies.

La stratégie basée sur l'utilisation des ressources a un taux d'utilisation des ressources *Cloud* meilleur que la stratégie basée sur *Fuzzy Competitor*. En effet, lorsque les nœuds *Edge* commencent à être chargés, la stratégie basée sur l'utilisation des ressources envoie systématiquement les nouvelles tâches vers le *Cloud*. En revanche, cela peut causer des échecs si les connexions WAN sont congestionnées.




a b

FIGURE 6.8 – Utilisation moyenne des ressources *Edge* et Cloud par les stratégies de placement

Lorsque, le système est très chargé, les différents algorithmes étudiés ont une utilisation assez similaire des ressources des nœuds. Cela peut s'expliquer par la possibilité des différents algorithmes d'orchestration de profiter de la grande disponibilité du réseau métropolitain pour placer les nouveaux services arrivant vers les nœuds *Edge* situés dans des zones moins denses en utilisateurs. Ainsi, les différents algorithmes d'orchestration étudiés peuvent s'adapter aux changements dynamiques de l'infrastructure *Edge* si le réseau MAN est hautement disponible.

***Principales causes des échecs du placement des services*** : L'un des avantages d'Edge*Cloud*Sim est la production détaillée des résultats de simulation des causes d'échec de placements et des exécutions des services. L'analyse des causes de placement permet d'optimiser la stratégie d'orchestration et l'infrastructure Edge. La figure 6.9 illustre une synthèse des taux échecs par cause. Il y a trois principales causes d'échec. Ce sont les échecs dus à la disponibilité des ressources de calcul, à la mobilité et aux réseaux. Les échecs dus à la disponibilité des ressources comprennent les échecs causés par le manque de ressources dans l'infrastructure *Edge* et le manque de ressources dans le *Cloud*. En effet, le manque de ressources dans l' *Edge* peut causer le rejet de nou-

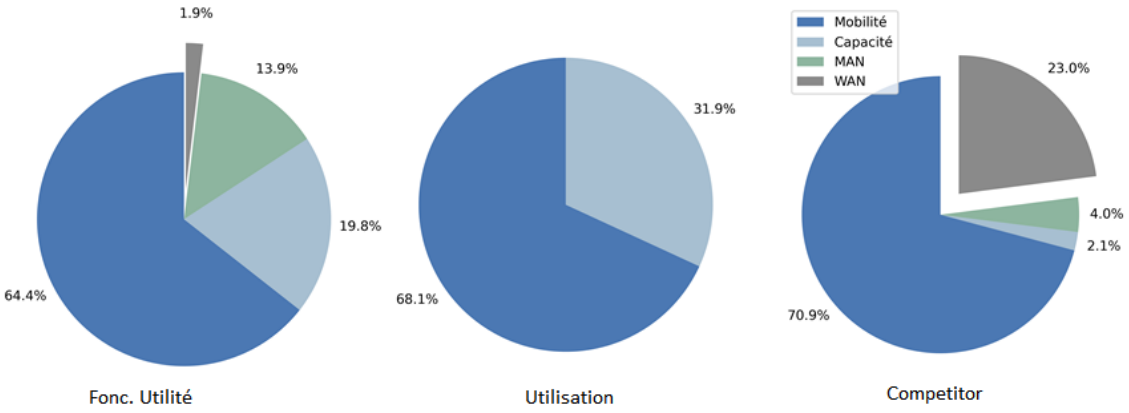



FIGURE 6.9 – Causes des échecs de placement

velles requêtes de placement. Toutefois, les échecs dus au manque de disponibilité des ressources sont moindres que ceux causés par la mobilité et les réseaux.

Les échecs dus aux réseaux comprennent les échecs causés par les réseaux locaux sans fil, MAN et WAN. Ces échecs s'expliquent généralement par la non considération de l'état des réseaux par les algorithmes d'orchestration. En effet, lorsque le nombre d'utilisateurs est proche de la saturation, les liens WAN sont congestionnés. Néanmoins, notre stratégie permet un taux d'échec faible dû au réseau, grâce à la considération par l'algorithme d'orchestration de la latence moyenne et de la bande passante avant de prendre une décision de placement.

Les échecs dus à la mobilité représentent la majorité des échecs. Dans l'*Edge Computing*, les échecs de placement de services causés par la mobilité sont inévitables. C'est pourquoi, la prédiction de la mobilité de l'utilisateur a été intégrée dans la stratégie d'orchestration pour réduire les échecs dus à la mobilité. Grâce à la prédiction de la mobilité et la considération de la latence, la stratégie d'orchestration proposée réalise un taux d'échec faible dû à la mobilité par rapport aux stratégies comparées.

## 6.7 Conclusion

L'*Edge Computing* et les nouvelles architectures réseaux promettent un bel avenir aux systèmes informatiques ubiquitaires tels que l'IoT. Cependant, il reste beaucoup à faire pour surmonter les défis liés au placement et à la migration dynamique de services en fonction de la mobilité des utilisateurs dans ces infrastructures. Afin de relever ce défi, nous avons proposé dans ce chapitre, une nouvelle stratégie de placement dynamique

de blocs d'un service dans une infrastructure *Edge Computing*. L'avantage de cette stratégie est qu'elle utilise les fonctions d'utilité, outils de sélection multicritères légers et ayant peu d'impact sur le temps de placement, pour la sélection de la meilleure zone de destination parmi les zones futures identifiées grâce à la prédiction de la mobilité. Cette prédiction permet aussi de prendre les décisions de placement au bon moment et d'éviter les éventuelles coupures de services liées au différents problèmes identifiés (surcharge de nœuds, faible bande passante, distance par rapport aux nœuds exécutant les services, etc.). Ainsi, cela peut permettre de réduire le taux d'échec de placement.

Il est possible que lors du placement d'un service, les ressources disponibles sur un nœud ne soient pas suffisantes. Pour répondre à ce problème, la stratégie proposée utilise la logique floue, une technique d'intelligence artificielle pour contrôler le nœud *Edge* qui peut recevoir le service en un ou en plusieurs blocs. Les évaluations effectuées ont prouvé les performances de notre proposition, notamment avec un très faible taux d'échec de placement dû à la mobilité.

Cependant, cette contribution de placement de service dans les infrastructures *Edge Computing* présente quelques limites. En effet, d'une manière générale, la couverture réseau d'une ville est assurée par les centres de données *Edge* de différents opérateurs. Par conséquent, une multitude de domaines *Edge* coexistent. Cet aspect n'a pas été considéré dans notre proposition. Une autre perspective intéressante consisterait à mettre en œuvre la stratégie d'orchestration dans un environnement réel pour en estimer les performances réelles.

# Conclusion et perspectives

Les applications IoT de la ville intelligente étendent les limites de l'informatique classique en interconnectant le monde physique grâce aux données récoltées par les dispositifs embarqués, les capteurs, ou tout autre appareil électronique. Cependant, la mise en œuvre de ces applications implique plusieurs problèmes de sécurité et de protection de la vie privée. Plusieurs travaux ont été effectués et ont permis de développer des solutions de sécurité. Une solution intéressante est la mise en œuvre de la sécurité sensible au contexte. En effet, compte tenu des changements très fréquents caractérisant les environnements IoT, il peut être pertinent d'adapter la mise en œuvre des mécanismes de sécurité et de protection de la vie privée aux différents contextes induits par ces changements.

Dans cette thèse, nous nous sommes intéressés à la sécurité et à la protection de la vie privée centrées sur l'utilisateur dans l'IoT. Dans ce qui suit, nous résumons les principales contributions qui ont été détaillées dans ce manuscrit. Enfin, nous introduisons quelques perspectives intéressantes qui peuvent être explorées dans la continuité des travaux de cette thèse.

## Principales contributions

Dans le contexte de la sécurité et de la protection de la vie privée centrées sur l'utilisateur dans l'IoT, notre objectif était de définir une solution de sécurité et de protection de la vie privée qui permet de prendre en considération les caractéristiques des utilisateurs d'applications IoT. Les principales contributions sont :

— **la définition d'une architecture de sécurité et de protection de la vie privée sensibles au contexte en tant que service adaptée à l'IoT :** cette première contribution correspond à la définition d'une architecture de sécurité et de protection de la vie privée sensibles au contexte en tant que service permettant d'adapter le niveau de sécurité des applications en fonction du contexte de l'utilisateur. Pour cela, une analyse des solutions existantes de sécurité sensible au contexte dans

l'IoT et des besoins de sécurité et de protection de la vie privée dans les applications IoT a été réalisée. Cela a permis la proposition d'une nouvelle architecture adaptée à l'IoT et aux nouvelles architectures réseaux. La solution proposée s'appuie sur l'approche 'as a service' pour la prise en charge d'une diversité d'applications IoT de la ville intelligente. Cette conception comprend deux plans. Le plan de connaissance mettant en œuvre l'intelligence artificielle pour la détection et la prédiction du contexte, l'évaluation du niveau de risque et la gestion des préférences des utilisateurs. Ensuite, le plan de sécurité et de protection de la vie privée mettant en œuvre les décisions d'adaptation du niveau de sécurité à travers divers mécanismes de sécurité et de protection de la vie privée sensibles au contexte ;

— **la définition et l'implémentation d'un système de gestion sécurisée et fiable de la sensibilité au contexte basée sur des dispositifs de confiance :** notre deuxième contribution dans cette thèse est la définition, l'implémentation et l'évaluation d'un système de gestion sécurisée et fiable de la sensibilité au contexte dans un environnement IoT. Le système proposé permet d'assurer la sécurité des échanges d'informations contextuelles et de détecter les sources de contexte aux comportements malveillants. Dans un premier temps, un mécanisme de sécurisation des échanges d'informations contextuelles est utilisé pour la collecte d'information contextuelle. Cela permet d'empêcher les accès non autorisés aux informations et le rejeu d'informations contextuelles. Ensuite, un mécanisme de confiance intégré analyse la fiabilité des informations contextuelles et le comportement des sources de contexte. Ainsi, le système de sécurité sensible au contexte peut prendre des décisions d'adaptation fiables ;

— **la définition et l'implémentation d'un système de gestion décentralisée des autorisations sensibles au contexte en tant que service :** cette troisième contribution correspond à la définition, l'implémentation et l'évaluation d'un système de gestion décentralisée des autorisations sensibles au contexte en tant que service pour l'IoT. Cette proposition, étend le framework ACE sur deux aspects. Premièrement, elle ajoute à ACE un aspect dynamique avec l'introduction des jetons d'accès contextuels. Cela permet aux serveurs de ressources de n'autoriser l'accès à une ressource que lorsque le contexte de l'utilisateur l'autorise. Deuxièmement, elle décentralise le serveur d'autorisation d'ACE avec l'introduction de la technologie blockchain. Grâce à cette décentralisation, notamment, les utilisateurs peuvent

aisément et dynamiquement gérer les autorisations (ajout, modification et révocation) sans passer par une autorité de confiance tierce. Enfin, la conception en tant que service permet l'intégration du système proposé dans tout service de sécurité sensible au contexte pour l'IoT ;

— **la définition et l'implémentation d'une stratégie de placement de blocs de service dans une infrastructure *Edge Computing* :** la quatrième et dernière contribution de cette thèse correspond à la définition, l'implémentation et l'évaluation d'une stratégie d'orchestration de blocs de service dans une infrastructure Edge Computing. La solution proposée permet de garantir un taux d'échec de placement faible et une bonne utilisation des ressources lorsque l'infrastructure est saturée. En effet, la prédiction de la mobilité des utilisateurs permet d'éviter les tentatives de placement soudaines de services. Pour mieux utiliser les ressources disponibles dans certaines zones saturées d'une infrastructure Edge Computing, la stratégie proposée utilise la logique floue pour contrôler le placement des blocs de services sur plusieurs nœuds ou sur un seul nœud.

## Perspectives et nouveaux défis

Les travaux exposés dans cette thèse contribuent aux efforts fournis afin d'assurer une sécurité et une protection de la vie privée permettant aux utilisateurs d'avoir confiance en leurs interactions avec l'IoT et de favoriser son adoption. Cependant, ces travaux ne permettent pas de répondre à l'ensemble des problématiques posées par la sécurité et la protection de la vie privée centrées sur l'utilisateur dans l'IoT. Plusieurs perspectives peuvent être identifiées :

— **l'évaluation en condition réelle du module de gestion de confiance des dispositifs :** dans cette thèse, nous avons réussi à définir un système de gestion sécurisée et fiable de la sensibilité au contexte basé sur des dispositifs de confiance. Ce système intègre l'architecture proposée dans le chapitre 3 comme module de gestion de la confiance des dispositifs (GCD). Les évaluations effectuées sur ce système (cf. chapitre 4, section 4.4.2), bien qu'elles aient été réalisées avec des dispositifs physiques, ont été fait en laboratoire en période de pandémie de la Covid19. Ainsi, il conviendra de réaliser une série de tests en conditions réelles avec des utilisateurs utilisant quotidiennement les différents dispositifs dans le respect du RGPD.

Nous pouvons aussi étendre le réseau bayésien proposé pour ce système (cf. chapitre 4, section 4.3.3.2.1) avec d'autres informations contextuelles et l'alimenter dynamiquement avec les informations du profil de l'utilisateur. ;

— **l'implémentation et l'évaluation du module gestionnaire de l'évaluation des risques du plan de connaissance :** le plan de connaissance de l'architecture de sécurité et de protection de la vie privée sensibles au contexte en tant que service proposé dans cette thèse comprend un module de gestion d'évaluation des risques (cf. chapitre 3, section 3.4.3). Ce module a pour objectif d'utiliser l'intelligence artificielle pour dynamiquement évaluer les niveaux de risques de sécurité et de non-respect de la vie privée associés à un contexte en fonction des menaces associées à ce contexte. Ce mécanisme doit être implémenté et ses performances doivent être évaluées en conditions réelles ;

— **l'implémentation et l'évaluation de l'architecture :** l'architecture de sécurité et de protection de la vie privée sensibles au contexte proposée dans cette thèse comprend deux principaux plans. Nous avons démontré les bénéfices de ces deux plans pour l'adaptation dynamique du niveau de sécurité des applications IoT de la ville intelligente tout en prenant en compte les préférences des utilisateurs. Ainsi, l'implémentation et l'évaluation de cette architecture comme un service est une perspective intéressante. En effet, le service qui sera implémenté pourra être éclaté en plusieurs services représentants différentes parties de l'architecture et pouvant être placés de manière indépendante ou ensemble dans une infrastructure Edge Computing selon la disponibilité des ressources. Le service proposé devra également exposer des API qui seront consommées par les applications IoT pour le déchargement de leurs tâches de sécurité et de protection de la vie privée ;

— **le déploiement et l'évaluation des performances en conditions réelles de l'architecture en tant que service dans une infrastructure 5G :** il serait très intéressant d'évaluer l'architecture proposée et implémentée comme un service en conditions réelles, par exemple dans un réseau 5G. Cela permettra de déterminer les performances en conditions réelles de l'architecture, notamment en termes de sécurité et de protection de la vie privée adaptative en temps réel ;

— **la définition, l'implémentation et l'évaluation d'une stratégie de placement de services dans les infrastructures *Edge Computing* multi-domaines :** dans cette thèse, nous avons défini et évaluer une stratégie de placement de service dans une infrastructure *Edge Computing*. La stratégie de placement proposée a été implémentée et testée sur notre plateforme de simulation. Le succès de ce test permet de valider notre approche et de montrer son utilité pour les infrastructures *Edge Computing*. Cependant, les infrastructures *Edge Computing* qui assurent la couverture géographique d'une ville n'appartiennent pas au même domaine. En effet, elles sont gérées par différents opérateurs. Il est possible qu'un utilisateur d'un service d'un opérateur X arrive dans une zone géographique peu couverte par son opérateur. En revanche, cette zone est bien couverte par les opérateurs Y et Z. Par conséquent, proposer et évaluer en conditions réelles une stratégie de placement de service dans les infrastructures *Edge Computing* multi-domaines est une perspective intéressante.

# Bibliographie


[3GP19] 3GPP. Release 15. https ://www.3gpp.org/release-15, April 2019.

[3GP20] 3GPP. Release 16. https ://www.3gpp.org/release-16, 2020.

[AAC+18] Olivier Alphand, Michele Amoretti, Timothy Claeys, Simone Dall'Asta, Andrzej Duda, Gianluigi Ferrari, Franck Rousseau, Bernard Tourancheau, Luca Veltri, and Francesco Zanichelli. IoTChain : A blockchain security architecture for the Internet of Things. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, Barcelona, April 2018. IEEE.

[AALS16] Tatiana Aubonnet, Boubendir Amina, Frederic Lemoine, and Noemie Simoni. Controlled Components for Internet of Things As-A-Service. *Open Journal of Internet Of Things (OJIOT)*, 2(1) :16–33, 2016.

[AAOW18] Vangalur Alagar, Alaa Alsaig, Olga Ormandjiva, and Kaiyu Wan. Context-Based Security and Privacy for Healthcare IoT. In *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pages 122–128, Xi'an, China, August 2018. IEEE.

[ACK+19] Amel Arfaoui, Soumaya Cherkaoui, Ali Kribeche, Sidi Mohammed Senouci, and Mohamed Hamdi. Context-Aware Adaptive Authentication and Authorization in Internet of Things. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6, Shanghai, China, May 2019. IEEE.

[ADB+99] Gregory D. Abowd, Anind K Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a Better Understanding of Context and Context-Awareness. In *HUC '99 Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, pages 304–307, Karlsruhe, Germany, September 27 - 29, 1999. Springer-Verlag.

[AEM+18] Fahad Alkurdi, Ibrahim Elgendi, Kumudu S. Munasinghe, Dharmendra Sharma, and Abbas Jamalipour. Blockchain in IoT Security : A Survey. In *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, pages 1–4, Sydney, NSW, November 2018. IEEE.

[AIM10] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things : A survey. *Computer Networks*, 54(15) :2787–2805, October 2010.

[AK19] Jinesh Ahamed and Firoz Khan. An Enhanced Context-aware Capability-based Access Control Model for the Internet of Things in Healthcare. In *2019 Sixth HCT Information Technology Trends (ITT)*, pages 126–131, Ras Al Khaimah, United Arab Emirates, November 2019. IEEE.

[AKM17] Yosef Ashibani, Dylan Kauling, and Qusay H. Mahmoud. A context-aware authentication service for smart homes. In *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 588–589, Las Vegas, NV, USA, January 2017. IEEE.

[ALC+19a] Tatiana Aubonnet, Frédéric Lemoine, A Cadzow, B Dupré, and N Simoni. User Group ; User Centric Approach ; Guidance for providers and standardization makers. Technical Report TR 103 603, European Telecommunications Standards Institute (ETSI), France, April 2019.

[ALC+19b] Tatiana Aubonnet, Frédéric Lemoine, A Cadzow, B Dupré, and N Simoni. User Group ; User centric approach in Digital Ecosystem. Technical Report TR 103 438, European Telecommunications Standards Institute (ETSI), France, April 2019.

[AM00] Gregory D. Abowd and Elizabeth D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, 7(1) :29–58, March 2000.

[ANRH15] Muhammad Harith Amaran, Nazmin Arif Mohd Noh, Mohd Saufy Rohmad, and Habibah Hashim. A Comparison of Lightweight Communi-

cation Protocols in Robotic Applications. *Procedia Computer Science*, 76 :400–405, 2015.

[APK19] Haripriya A. P. and Kulothungan K. Secure-MQTT : An efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things. *EURASIP Journal on Wireless Communications and Networking*, 2019(1) :90, December 2019.

[ARC18] Mahmoud Ammar, Giovanni Russello, and Bruno Crispo. Internet of Things : A survey on the security of IoT frameworks. *Journal of Information Security and Applications*, 38 :8–27, February 2018.

[ASEJY18] Adnan Ahmed Abi Sen, Fathy Albouraey Eassa, Kamal Jambi, and Mohammad Yamin. Preserving privacy in internet of things : A survey. *International Journal of Information Technology*, 10(2) :189–200, June 2018.

[ASL+19] Iqbal Alam, Kashif Sharif, Fan Li, Zohaib Latif, Md Monjurul Karim, Boubakr Nour, Sujit Biswas, and Yu Wang. IoT Virtualization : A Survey of Software Definition & Function Virtualization Techniques for Internet of Things. *arXiv :1902.10910 [cs]*, February 2019.

[Aut] Auth0. Access Tokens. https ://auth0.com/docs/.

[BBBG19] Andrew Banks, Ed Briggs, Ken Borgendale, and Rahul Gupta. MQTT Version 5.0 OASIS Standard. https ://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html, July 2019.

[BBC+19] Paolo Bellavista, Javier Berrocal, Antonio Corradi, Sajal K. Das, Luca Foschini, and Alessandro Zanni. A survey on fog computing for the Internet of Things. *Pervasive and Mobile Computing*, 52 :71–99, January 2019.

[BBDL+13] Martin Bauer, Nicola Bui, Jourik De Loof, Carsten Magerkurth, Andreas Nettsträter, Julinda Stefa, and Joachim W. Walewski. IoT Reference Model. In Alessandro Bassi, Martin Bauer, Martin Fiedler, Thorsten Kramp,

Rob van Kranenburg, Sebastian Lange, and Stefan Meissner, editors, *Enabling Things to Talk*, pages 113–162. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[BBF+13] Alessandro Bassi, Martin Bauer, Martin Fiedler, Thorsten Kramp, Rob van Kranenburg, Sebastian Lange, and Stefan Meissner, editors. *Enabling Things to Talk*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[BBHMSG14] Jorge Bernal Bernabe, Jose Luis Hernández, M. Victoria Moreno, and Antonio F. Skarmeta Gomez. Privacy-Preserving Security Framework for a Social-Aware Internet of Things. In Ramón Hervás, Sungyoung Lee, Chris Nugent, and José Bravo, editors, *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services*, volume 8867, pages 408–415. Springer International Publishing, Cham, 2014.

[BBS18] Amina Boubendir, Emmanuel Bertin, and Noemie Simoni. Flexibility and dynamicity for open network-as-a-service : From VNF and architecture modeling to deployment. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6, Taipei, April 2018. IEEE.

[BCFS19] Paolo Bellavista, Antonio Corradi, Luca Foschini, and Domenico Scotece. Differentiated Service/Data Migration for Edge Services Leveraging Container Characteristics. *IEEE Access*, 7 :139746–139758, 2019.

[BCR+18] Elaine Barker, Lily Chen, Allen Roginsky, Apostol Vassilev, and Richard Davis. Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography. Technical Report NIST Special Publication (SP) 800-56A Rev. 3, National Institute of Standards and Technology, April 2018.

[BEK14] C. Bormann, M. Ersue, and A. Keranen. Terminology for Constrained-Node Networks. Technical Report RFC 7228, RFC Editor, May 2014.

[Ber21] Olaf Bergmann. Libcoap : C-Implementation of CoAP. *URL : https ://libcoap.net/*, 2021.

[BF16] Joe Biron and Jonathan Follett. *Foundational Elements of an IoT Solution*. O'Reilly Media, Inc., United State of America, first edition edition, 2016.

[BH08] Alessandro Bassi and Geir Horn. Internet of Things in 2020 : Roadmap for the future. Technical report, May 2008.

[BH13] C. Bormann and P. Hoffman. Concise Binary Object Representation (CBOR). https ://www.rfc-editor.org/info/rfc7049, October 2013.

[BHJ+18] J. Bradley, P. Hunt, M. Jones, H. Tschofenig, and M. Mihály. OAuth 2.0 Proof-of-Possession : Authorization Server to Client Key Distribution. https ://tools.ietf.org/id/draft-ietf-oauth-pop-key-distribution-04.html#rfc.section.1, 2018.

[BHNEA15] Alireza Baratloo, Mostafa Hosseini, Ahmed Negida, and Gehad El Ashal. Part 1 : Simple Definition and Calculation of Accuracy, Sensitivity and Specificity. *Emergency (Tehran, Iran)*, 3(2) :48–49, 2015.

[BM04] P. Brézillon and G.K. Mostéfaoui. Context-based security policies : A new modeling approach. In *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*, pages 154–158, Orlando, FL, USA, 2004. IEEE.

[BM07] Carsten Bormann and Geoffrey Milligan. IPv6 over Low power WPAN (6lowpan), September 2007.

[Bor18] Carsten Bormann. Object Security for Constrained RESTful Environments (OSCORE), September 2018.

[BPGB18] Mahmoud Barhamgi, Charith Perera, Chirine Ghedira, and Djamal Benslimane. User-centric Privacy Engineering for the Internet of Things. *arXiv :1809.00926 [cs]*, September 2018.

[BRKK18] Muhammad Burhan, Rana Rehman, Bilal Khan, and Byung-Seo Kim. IoT Elements, Layered Architectures and Security Issues : A Comprehensive Survey. *Sensors*, 18(9) :2796, August 2018.

[Bro] Daniel R L Brown. SEC 1 : Elliptic Curve Cryptography. page 151.

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-Policy Attribute-Based Encryption. In *2007 IEEE Symposium on Security and Privacy (SP '07)*, pages 321–334, Berkeley, CA, May 2007. IEEE.

[BTTK14] Sinda Boussen, Nabil Tabbane, Sami Tabbane, and Francine Krief. A context aware vertical handover decision approach based on fuzzy logic. In *Fourth International Conference on Communications and Networking, ComNet-2014*, pages 1–5, Hammamet, Tunisia, March 2014. IEEE.

[Bur17] Samuel Burke. Google admits its new smart speaker was eavesdropping on users. https ://money.cnn.com/2017/10/11/technology/google-home-mini-security-flaw/index.html, October 2017.

[BZL+21] Abdul Basit, Maham Zafar, Xuan Liu, Abdul Rehman Javed, Zunera Jalil, and Kashif Kifayat. A comprehensive survey of AI-enabled phishing attacks detection techniques. *Telecommunication Systems*, 76(1) :139–154, January 2021.

[BZWL19] Aniruddha Bhattacharjya, Xiaofeng Zhong, Jing Wang, and Xing Li. Security Challenges and Concerns of Internet of Things (IoT). In Song Guo and Deze Zeng, editors, *Cyber-Physical Systems : Architecture, Security and Application*, pages 153–185. Springer International Publishing, Cham, 2019.

[CCT20] Giulia Cisotto, Edoardo Casarin, and Stefano Tomasin. Requirements and Enablers of Advanced Healthcare Services over Future Cellular Systems. *IEEE Communications Magazine*, 58(3) :76–81, March 2020.

[CD16] Konstantinos Christidis and Michael Devetsikiotis. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access*, 4 :2292–2303, 2016.

[CFN17] Petr Cintula, Christian G. Fermüller, and Carles Noguera. Fuzzy Logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2017 edition, 2017.

[CGB16] Ing-Ray Chen, Jia Guo, and Fenye Bao. Trust Management for SOA-Based IoT and Its Application to Service Composition. *IEEE Transactions on Services Computing*, 9(3) :482–495, May 2016.

[Cis14] Cisco. The Internet of Things Reference Model White Paper. Technical report, IoT World Forum, June 2014.

[CLD+14] Sophie Chabridon, Romain Laborde, Thierry Desprats, Arnaud Oglaza, Pierrick Marie, and Samer Machara Marquez. A survey on addressing privacy together with quality of context for context management in the Internet of Things. *annals of telecommunications - annales des télécommunications*, 69(1-2) :47–62, February 2014.

[CLF+19] Min Chen, Wei Li, Giancarlo Fortino, Yixue Hao, Long Hu, and Iztok Humar. A Dynamic Service Migration Mechanism in Edge Cognitive Computing. *ACM Transactions on Internet Technology*, 19(2) :1–15, April 2019.

[CLR+18] Yuling Chen, Min Lei, Wei Ren, Yi Ren, and Zhiguo Qu. RoFa : A Robust and Flexible Fine-Grained Access Control Scheme for Mobile Cloud and IoT based Medical Monitoring. *Fundamenta Informaticae*, 157(1-2) :167–184, January 2018.

[CLST20] Jimmy Carrión, Patricia Ludeña-González, Francisco Sandoval, and Rommel Torres. Evaluation of Utility Function Algorithm for Congestion Control in Computer Networks. In Germania Rodriguez Morales, Efraín R. Fonseca C., Juan Pablo Salgado, Pablo Pérez-Gosende, Marcos Orellana Cordero, and Santiago Berrezueta, editors, *Information and Communication Technologies*, volume 1307, pages 453–467. Springer International Publishing, Cham, 2020.

[CMS18] Pushpinder Kaur Chouhan, Sally McClean, and Mark Shackleton. Situation Assessment to Secure IoT Applications. In *2018 Fifth International Conference on Internet of Things : Systems, Management and Security*, pages 70–77, Valencia, October 2018. IEEE.

[Cou16] Council and Parliament of European Union. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), April 2016.

[CRB+11] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajkumar Buyya. CloudSim : A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software : Practice and Experience*, 41(1) :23–50, January 2011.

[CRT17] Timothy Claeys, Franck Rousseau, and Bernard Tourancheau. Securing Complex IoT Platforms with Token Based Access Control and Authenticated Key Establishment. In *2017 International Workshop on Secure Internet of Things (SIoT)*, pages 1–9, Oslo, September 2017. IEEE.

[CTC+19] Juan Chen, Zhihong Tian, Xiang Cui, Lihua Yin, and Xianzhi Wang. Trust architecture and reputation evaluation for internet of things. *Journal of Ambient Intelligence and Humanized Computing*, 10(8) :3099–3107, August 2019.

[CWX+20] Hao Chen, Wunan Wan, Jinyue Xia, Shibin Zhang, Jinquan Zhang, Xizi Peng, and Xingjie Fan. Task-Attribute-Based Access Control Scheme for IoT via Blockchain. *Computers, Materials & Continua*, 65(3) :2441–2453, 2020.

[Dai19] Wei Dai. Crypto++ Library 8.2 | Free C++ Class Library of Cryptographic Schemes. https ://www.cryptopp.com/, April 2019.

[DBN+01] Morris J. Dworkin, Elaine B. Barker, James R. Nechvatal, James Foti, Lawrence E. Bassham, E. Roback, and James F. Dray Jr. Advanced Encryption Standard (AES). November 2001.

[dFW20] Cyprien Delpech de Saint Guilhem, Marc Fischlin, and Bogdan Warinschi. Authentication in Key-Exchange : Definitions, Relations and Composition. In *2020 IEEE 33rd Computer Security Foundations Symposium (CSF)*, pages 288–303, Boston, MA, USA, June 2020. IEEE.

[DGR+19] Ikram Ud Din, Mohsen Guizani, Joel J.P.C. Rodrigues, Suhaidi Hassan, and Valery V. Korotaev. Machine learning in the Internet of Things : Designed techniques for smart cities. *Future Generation Computer Systems*, 100 :826–843, November 2019.

[DK17] Parwinder Kaur Dhillon and Sheetal Kalra. A lightweight biometrics based remote user authentication scheme for IoT services. *Journal of Information Security and Applications*, 34 :255–270, June 2017.

[DMRF+18] B. Di Martino, M. Rak, M. Ficco, A. Esposito, S.A. Maisto, and S. Nacchia. Internet of things reference architectures, security and interoperability : A survey. *Internet of Things*, 1-2 :99–112, September 2018.

[DNS+16] Prajit Kumar Das, Sandeep Narayanan, Nitin Kumar Sharma, Anupam Joshi, Karuna Joshi, and Tim Finin. Context-Sensitive Policy Based Security in Internet of Things. In *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 1–6, St Louis, MO, USA, May 2016. IEEE.

[DR02] Joan Daemen and Vincent Rijmen. The design of rijndael : AES - the advanced encryption standard. 2002.

[dTAH18a] Everton de Matos, Ramao Tiago Tiburski, Leonardo Albernaz Amaral, and Fabiano Hessel. Context Interoperability for IoT Through an Edge-Centric Context Sharing Architecture. In *2018 IEEE Symposium on Com-*

*puters and Communications (ISCC)*, pages 00667–00670, Natal, Brazil, June 2018. IEEE.

[dTAH18b] Everton de Matos, Ramao Tiago Tiburski, Leonardo Albernaz Amaral, and Fabiano Hessel. Providing Context-Aware Security for IoT Environments Through Context Sharing Feature. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1711–1715, New York, NY, USA, August 2018. IEEE.

[Dub18] Chantelle Dubois. MIT's Encryption Chip Reduces Public-Key Encryption Power Consumption by 99.75%. https ://www.allaboutcircuits.com/news/energy-efficient-public-key-encryption-chip-solve-iot-security-problems/, 18.03.18.

[Dum16] Sebastien Dumoulin. La 5G lancée dans une course d'obstacles. https ://www.lesechos.fr/24/02/2016/lesechos.fr/021721539125_la-5g-lancee-dans-une-course-d-obstacles.htm, February 16.

[Ecl20] Eclipse/paho.mqtt.embedded-c. Eclipse Foundation, June 2020.

[Eri19] Ericsson. Internet of Things forecast – Ericsson Mobility Report. https ://www.ericsson.com/en/mobility-report/internet-of-things-forecast, 2019.

[Ers13] Mehmet Ersue. ETSI NFV management and orchestration-An overview. *Presentation at the IETF*, 88, 2013.

[Eva11] Dave Evans. *The Internet of Things, How Does the Current Evolution of the Internet Transform the World ?* Livre Blanc. Cisco Internet Business Solutions Group (IBSG), April 2011.

[Eys01] G Eysenbach. What is e-health ? *Journal of Medical Internet Research*, 3(2), June 2001.

[FBM12] D. Fraga, Z. Bankovic, and J.M. Moya. A Taxonomy of Trust and Reputation System Attacks. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 41–50, Liverpool, United Kingdom, June 2012. IEEE.

[Fer05] Niels Ferguson. Authentication weaknesses in GCM. May 2005.

[FPS+19] Nikos Fotiou, Iakovos Pittaras, Vasilios A. Siris, Spyros Voulgaris, and George C. Polyzos. Secure IoT access at scale using blockchains and smart contracts. *arXiv :1907.03904 [cs]*, July 2019.

[FTC15] FTC Staff. Internet of Things : Privacy and Security in a Connected World. Technical report, Federal Trade Commission, January 2015.

[GBKB20] Kshitiz Goel, Abhishek Bhaumick, Deepika Kaushal, and Saurabh Bagchi. Reliability Analysis of Edge Scenarios Using Pedestrian Mobility. In *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*, pages 61–62, Valencia, Spain, June 2020. IEEE.

[GBMP13] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of Things (IoT) : A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7) :1645–1660, September 2013.

[GH04] Henri Gilbert and Helena Handschuh. Security Analysis of SHA-256 and Sisters. In Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Mitsuru Matsui, and Robert J. Zuccherato, editors, *Selected Areas in Cryptography*, volume 3006, pages 175–193. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[GKS18] O. Garcia-Morchon, S. Kumar, and M. Sethi. State-of-the-Art and Challenges for the Internet of Things Security. https ://tools.ietf.org/id/draft-irtf-t2trg-iot-seccons-13.html, April 2018.

[GM19] Ghada Glissa and Aref Meddeb. 6LowPSec : An end-to-end security protocol for 6LoWPAN. *Ad Hoc Networks*, 82 :100–112, January 2019.

[GMBC19] Amitabha Ghosh, Andreas Maeder, Matthew Baker, and Devaki Chandramouli. 5G Evolution : A View on 5G Cellular Technology Beyond 3GPP Release 15. *IEEE Access*, 7 :127639–127651, 2019.

[GMS14] Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva. Network-layer security for the Internet of Things using TinyOS and BLIP : Network-Layer Security For The IoT using TinyOS and BLIP. *International Journal of Communication Systems*, 27(10) :1938–1963, October 2014.

[GMSS23] Jorge Granjal, Edmundo Monteiro, and Jorge Sa Silva. Security for the Internet of Things : A Survey of Existing Protocols and Open Research Issues. *IEEE Communications Surveys & Tutorials*, 17(3) :1294–1312, 23.

[GoE21] Go Ethereum. https ://github.com/ethereum/go-ethereum/releases, March 2021.

[GOP12] Carles Gomez, Joaquim Oller, and Josep Paradells. Overview and Evaluation of Bluetooth Low Energy : An Emerging Low-Power Wireless Technology. *Sensors*, 12(9) :11734–11753, August 2012.

[Hap18] HappyEmu. https ://github.com/HappyEmu/ace, 2018.

[Har12] D. Hardt. The OAuth 2.0 Authorization Framework. Technical Report RFC6749, RFC Editor, October 2012.

[HDA+13] Eiji Hayashi, Sauvik Das, Shahriyar Amini, Jason Hong, and Ian Oakley. CASA : Context-aware scalable authentication. In *Proceedings of the Ninth Symposium on Usable Privacy and Security - SOUPS '13*, page 1, Newcastle, United Kingdom, 2013. ACM Press.

[HDMP+21] Hamed HaddadPajouh, Ali Dehghantanha, Reza M. Parizi, Mohammed Aledhari, and Hadis Karimipour. A survey on internet of things security : Requirements, challenges, and solutions. *Internet of Things*, 14 :100129, June 2021.

[Hen03] Karen Henricksen. *A Framework for Context-Aware Pervasive Computing Applications*. Computer Science, School of Information Technology and Electrical Engineering, University of Queensland, September 2003.

[HJ12] Dick Hardt and Michael Jones. The OAuth 2.0 Authorization Framework : Bearer Token Usage. https ://tools.ietf.org/html/rfc6750#page-2, October 2012.

[Hou07] Russell Housley. Using AES-CCM and AES-GCM Authenticated Encryption in the Cryptographic Message Syntax (CMS). https ://tools.ietf.org/html/rfc5084, November 2007.

[HTS08] Urs Hunkeler, Hong Linh Truong, and Andy Stanford-Clark. MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, pages 791–798, Bangalore, India, January 2008. IEEE.

[Hyp21] Hyperledger Fabric. https ://github.com/hyperledger/fabric/releases, March 2021.

[IEE16] IEEE Standard for Low-Rate Wireless Networks. Technical report, IEEE, April 2016.

[IEE20a] IEEE. 802.15.4-2020 - IEEE Standard for Low-Rate Wireless Networks. Technical report, IEEE, July 2020.

[IEE20b] IEEE. 802.15.4z-2020 - IEEE Standard for Low-Rate Wireless Networks– Amendment 1 : Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques. Technical report, IEEE, August 2020.

[ISO14] JTC1 ISO/IEC. Internet of Things (IoT), Preliminary report. https ://www.iso.org/files/live/sites/isoorg/files/developing_standards/docs/en/inter jtc1.pdf, 2014.

[ITU12] ITU. ITU-T Recommendation Y.4000/Y.2060, June 2012.

[IVH16] Cvitic Ivan, Miroslav Vujic, and Sinisa Husnjak. Classification of Security Risks in the IoT Environment. In Branko Katalinic, editor, *DAAAM Proceedings*, volume 1, pages 0731–0740. DAAAM International Vienna, first edition, 2016.

[JBA+20] Abdul Rehman Javed, Mirza Omer Beg, Muhammad Asim, Thar Baker, and Ali Hilal Al-Bayatti. AlphaLogger : Detecting motion-based side-channel attack using smartphone keystrokes. *Journal of Ambient Intelligence and Humanized Computing*, February 2020.

[JBS15] Michael Jones, John Bradley, and Nat Sakimura. JSON Web Signature (JWS). Technical Report RFC 7515, Internet Engineering Task Force (IETF), October 2015.

[JET18] M. Jones, S. Erdtman, and H. Tschofenig. CBOR Web Token (CWT). https ://datatracker.ietf.org/doc/html/rfc8392, May 2018.

[JH15] Michael Jones and Joe Hildebrand. JSON Web Encryption (JWE). Technical Report RFC 7516, Internet Engineering Task Force (IETF), May 2015.

[JJB12] Peter B. Jensen, Lars J. Jensen, and Søren Brunak. Mining electronic health records : Towards better research applications and clinical care. *Nature Reviews Genetics*, 13 :395, May 2012.

[JK16] Mary James and Deepa S. Kumar. An Implementation of Modified Lightweight Advanced Encryption Standard in FPGA. *Procedia Technology*, 25 :582–589, 2016.

[JMV01] Don Johnson, Alfred Menezes, and Scott Vanstone. The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security*, 1(1) :36–63, August 2001.

[Jon15a] Michael Jones. JSON Web Algorithms (JWA). Technical Report RFC 7518, Internet Engineering Task Force (IETF), May 2015.

[Jon15b] Michael Jones. JSON Web Key (JWK). RFC 7517, Internet Engineering Task Force (IETF), May 2015.

[JQM+18] Qi Jiang, Yuanyuan Qian, Jianfeng Ma, Xindi Ma, Qingfeng Cheng, and Fushan Wei. User centric three-factor authentication protocol for cloud-assisted wearable devices. *International Journal of Communication Systems*, page e3900, December 2018.

[JVB92] R. Jager, H.B. Verbruggen, and P.M. Bruijn. The Role of Defuzzification Methods in the Application of Fuzzy Control. *IFAC Proceedings Volumes*, 25(6) :75–80, May 1992.

[KAA+18] Akram Khan, Abdullah Al-Zahrani, Safwan Al-Harbi, Soliman Al-Nashri, and Iqbal A. Khan. Design of an IoT smart home system. In *2018 15th Learning and Technology Conference (L&T)*, pages 1–5, Jeddah, February 2018. IEEE.

[KAH+19] Wazir Zada Khan, Ejaz Ahmed, Saqib Hakak, Ibrar Yaqoob, and Arif Ahmed. Edge computing : A survey. *Future Generation Computer Systems*, 97 :219–235, August 2019.

[Ker83] Auguste Kerckhoffs. La cryptographie militaire. *Journal des Sciences Militaires XI*, pages 161–191, 1883.

[KIRM18] Amritpal Kaur, Isha, Gaurav Rai, and Arun Malik. Authentication and Context Awareness Access Control in Internet of Things : A Review. In *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 630–635, Noida, India, January 2018. IEEE.

[KJ17] Amandeep Kaur and Ashish Jasuja. Health monitoring based on IoT using Raspberry PI. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pages 1335–1340, Greater Noida, May 2017. IEEE.

[KKS20] Vladik Kreinovich, Olga Kosheleva, and Shahnaz N. Shahbazova. Why Triangular and Trapezoid Membership Functions : A Simple Explanation. In Shahnaz N. Shahbazova, Michio Sugeno, and Janusz Kacprzyk, editors, *Recent Developments in Fuzzy Logic and Fuzzy Sets*, volume 391, pages 25–31. Springer International Publishing, Cham, 2020.

[KM07] Nandakishore Kushalnagar and Gabriel Montenegro. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. https ://tools.ietf.org/html/rfc4944#page-6, September 2007.

[KMS07] N Kushalnagar, G Montenegro, and C Schumacher. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) : Overview, Assumptions, Problem Statement, and Goals. https ://datatracker.ietf.org/doc/rfc4919/, August 2007.

[KRA14] Pavlos Kosmides, Angelos Rouskas, and Miltiades Anagnostou. Utility-based RAT selection optimization in heterogeneous wireless networks. *Pervasive and Mobile Computing*, 12 :92–111, June 2014.

[KS13] G. S. Khekare and A. V. Sakhare. A smart city framework for intelligent traffic system using VANET. In *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, pages 302–305, Kottayam, March 2013. IEEE.

[KW15] Habib Kashif and Leister Wolfgang. Context-Aware Authentication for the Internet of Things. In *ICAS 2015 - The Eleventh International Conference on Autonomic and Autonomous Systems*, Rome, Italy, 2015. IARIA.

[LL15] Knud Lasse Lueth. The 10 most popular Internet of Things applications right now. https ://iot-analytics.com/10-internet-of-things-applications/, February 2015.

[LL19] YongJoo Lee and Keon Myung Lee. Blockchain-based RBAC for user authentication with anonymity. In *Proceedings of the Conference on Research in Adaptive and Convergent Systems*, pages 289–294, 2019.

[LLJ+17] Wei-Han Lee, Changchang Liu, Shouling Ji, Prateek Mittal, and Ruby Lee. Quantification of De-anonymization Risks in Social Networks :. In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, pages 126–135, Porto, Portugal, 2017. SCITEPRESS - Science and Technology Publications.

[LMD+17] Shi-Wan Lin, Bradford Miller, acques Durand, Graham Bleakley, Amine Chigani, Robert Martin, Brett Murphy, and Mark Crawford. The Industrial Internet of ThingsVolume G1 : Reference Architecture. Technical Report IIC :PUB :G1 :V1.80 :20170131, IIC, 2017.

[LMQ+14] Qiujian Lv, Zongshan Mei, Yuanyuan Qiao, Yufei Zhong, and Zhenming Lei. Hidden Markov Model based user mobility analysis in LTE network. In *2014 International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 379–384, Sydney, Australia, September 2014. IEEE.

[LNLLE+20] Bao Le Nguyen, E. Laxmi Lydia, Mohamed Elhoseny, Irina V. Pustokhina, Denis A. Pustokhin, Mahmoud Mohamed Selim, Gia Nhu Nguyen, and K. Shankar. Privacy Preserving Blockchain Technique to Achieve Secure and Reliable Sharing of IoT Data. *Computers, Materials & Continua*, 65(1) :87–107, 2020.

[LSW+18] L. Seitz, G. Selander, Erik Wahlstroem, Samuel Erdtman, and H. Tschofenig. Authentication and Authorization for Constrained Environments

(ACE) using the OAuth 2.0 Framework (ACE-OAuth). ACE Working Group Internet-draft, IETF, October 2018.

[LTL+19] Fang Liu, Guoming Tang, Youhuizi Li, Zhiping Cai, Xingzhou Zhang, and Tongqing Zhou. A Survey on Edge Computing Systems and Tools. *Proceedings of the IEEE*, 107(8) :1537–1562, August 2019.

[LWL+19] Xiang Li, Qixu Wang, Xiao Lan, Xingshu Chen, Ning Zhang, and Dajiang Chen. Enhancing Cloud-Based IoT Security Through Trustworthy Cloud Service : An Integration of Security and Reputation Approach. *IEEE Access*, 7 :9368–9383, 2019.

[LXI17] Shancang Li, Li D. Xu, and Imed Romdhani. *Securing the Internet of Things*. Syngress, Cambridge, MA, 2017.

[LY18] Gen Liang and Hewei Yu. Network selection algorithm for heterogeneous wireless networks based on service characteristics and user preferences. *EURASIP Journal on Wireless Communications and Networking*, 2018(1) :241, December 2018.

[MAA+14] Yaser Mowafi, Dhiah Abou-Tair, Tareq Al-Aqarbeh, Marat Abilov, Viktor Dmitriyev, and Jorge Marx Gomez. A Context-aware Adaptive Security Framework for Mobile Applications. In *Proceedings of the 3rd International Conference on Context-Aware Systems and Applications*, Dubai, United Arab Emirates, 2014. ICST.

[MAPP12] Parikshit N. Mahalle, Bayu Anggorojati, Neeli Rashmi Prasad, and Ramjee Prasad. Identity driven capability based access control (ICAC) scheme for the Internet of Things. In *2012 IEEE International Conference on Advanced Networks and Telecommunciations Systems (ANTS)*, pages 49–54, Bangalore, India, December 2012. IEEE.

[Mat19] J. and F. Palombini Mattsson. Comparison of CoAP Security Protocols. https ://tools.ietf.org/id/draft-ietf-lwig-security-protocol-comparison-03.html, March 2019.

[MB03] Ghita Kouadri Mostéfaoui and Patrick Brézillon. A Generic Framework for Context-Based Distributed Authorizations. In Patrick Blackburn, Chiara Ghidini, Roy M. Turner, and Fausto Giunchiglia, editors, *Modeling and Using Context*, volume 2680, pages 204–217. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[MB04] G.K. Mostéfaoui and P. Brézillon. Modeling context-based security policies with contextual graphs. In *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*, pages 28–32, Orlando, FL, USA, 2004. IEEE.

[McF15] Duncan McFarlane. The Origin of the Internet of Things, June 2015.

[MCK18] Léo Mendiboure, Mohamed Aymen Chalouf, and Francine Krief. Towards a Blockchain-Based SD-IoV for Applications Authentication and Trust Management. In Andrzej M.J. Skulimowski, Zhengguo Sheng, Sondès Khemiri-Kallel, Christophe Cérin, and Ching-Hsien Hsu, editors, *Internet of Vehicles. Technologies and Services Towards Smart City*, volume 11253, pages 265–277. Springer International Publishing, Cham, 2018.

[MCK20a] Leo Mendiboure, Mohamed Aymen Chalouf, and Francine Krief. Load-Aware and Mobility-Aware Flow Rules Management in Software Defined Vehicular Access Networks. *IEEE Access*, 8 :167411–167424, 2020.

[MCK20b] Leo Mendiboure, Mohamed Aymen Chalouf, and Francine Krief. A Scalable Blockchain-based Approach for Authentication and Access Control in Software Defined Vehicular Networks. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–11, Honolulu, HI, USA, August 2020. IEEE.

[MCK20c] Leo Mendiboure, Mohamed Aymen Chalouf, and Francine Krief. Survey on blockchain-based applications in internet of vehicles. *Computers & Electrical Engineering*, 84 :106646, June 2020.

[MD20] Parikshit N. Mahalle and Prashant S. Dhotre. *Security Issues in Context-Aware Systems*, volume 169, pages 137–149. Springer Singapore, Singapore, 2020.

[Men20] Léo Mendiboure. *Distribution géographique de données dans l'Internet des Véhicules : une approche logicielle et sécurisée utilisant les réseaux cellulaires*. PhD thesis, Université de Bordeaux, September 2020.

[MHB+17] Juan A Martínez, José L Hernández-Ramos, Victoria Beltrán, Antonio Skarmeta, and Pedro M Ruiz. A user-centric Internet of Things platform to empower users for managing security and privacy concerns in the Internet of Energy. *International Journal of Distributed Sensor Networks*, 13(8) :155014771772797, August 2017.

[MK18] M Durairaj and K Muthuramalingam. A New Authentication Scheme with Elliptical Curve Cryptography for Internet of Things (IoT) Environments, 2018.

[MM20] Erfan Farhangi Maleki and Lena Mashayekhy. Mobility-aware computation offloading in edge computing using prediction. In *2020 IEEE 4th International Conference on Fog and Edge Computing (ICFEC)*, pages 69–74, Melbourne, Australia, May 2020. IEEE.

[Mos14] Mosquitto. MQTT v3.1.1 now an OASIS Standard. http ://mqtt.org/2014/11/mqtt-v3-1-1-now-an-oasis-standard, November 2014.

[MRSB18] Sumit Maheshwari, Dipankar Raychaudhuri, Ivan Seskar, and Francesco Bronzino. Scalability and Performance Evaluation of Edge Cloud Systems for Latency Constrained Applications. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 286–299, Seattle, WA, USA, October 2018. IEEE.

[MS13] Prerna Mahajan and Abhishek Sachdeva. A Study of Encryption Algorithms AES, DES and RSA for Security. *Global Journal of Computer Science and Technology*, 2013.

[MS18] Daniel Miessler and Craig Smith. OWASP Internet of Things Project - OWASP. https ://www.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Vu 2018.

[MS21] Leo Mendiboue and Tidiane Sylla. Context Aware Performance Comparison, April 2021.

[MSD+19] Lukas Malina, Gautam Srivastava, Petr Dzurenda, Jan Hajny, and Radek Fujdiak. A Secure Publish/Subscribe Protocol for Internet of Things. In *Proceedings of the 14th International Conference on Availability, Reliability and Security - ARES '19*, pages 1–10, Canterbury, CA, United Kingdom, 2019. ACM Press.

[MXZ+17] Du Mingxiao, Ma Xiaofeng, Zhang Zhe, Wang Xiangwei, and Chen Qijun. A review on consensus algorithm of blockchain. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2567–2572, Banff, AB, October 2017. IEEE.

[MZC19] Huirong Ma, Zhi Zhou, and Xu Chen. Predictive Service Placement in Mobile Edge Computing. In *2019 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 792–797, Changchun, China, August 2019. IEEE.

[Nas17] Lavinia Nastase. Security in the Internet of Things : A Survey on Application Layer Protocols. In *2017 21st International Conference on Control Systems and Computer Science (CSCS)*, pages 659–666, Bucharest, Romania, May 2017. IEEE.

[NGHS17] Michael Nofer, Peter Gomber, Oliver Hinz, and Dirk Schiereck. Blockchain. *Business & Information Systems Engineering*, 59(3) :183–187, June 2017.

[NSB+15] Ricardo Neisse, Gary Steri, Gianmarco Baldini, Elias Tragos, Igor Nai Fovino, and Maarten Botterman. Dynamic Context-Aware Scalable and Trust-based IoT Security, Privacy Framework. In *Internet of Things - From Research and Innovation to Market Deployment*, River Publishers Series in Communication, pages 199–224. River Publishers, 2015.

[NSFB15] Ricardo Neisse, Gary Steri, Igor Nai Fovino, and Gianmarco Baldini. SecKit : A Model-based Security Toolkit for the Internet of Things. *Computers & Security*, 54 :60–76, October 2015.

[ODO17] Alma Oracevic, Selma Dilek, and Suat Ozdemir. Security in internet of things : A survey. In *2017 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6, Marrakech, Morocco, May 2017. IEEE.

[ONF09] ONF. OpenFlow Switch Specification. Technical Report V1.0.0, Open Networking Foundation, December 2009.

[ONF13] ONF. SDN Architecture Overview. Technical Report V1.0, Open Networking Foundation, December 2013.

[OZC18] Tao Ouyang, Zhi Zhou, and Xu Chen. Follow Me at the Edge : Mobility-Aware Dynamic Service Placement for Mobile Edge Computing. *IEEE Journal on Selected Areas in Communications*, 36(10) :2333–2345, October 2018.

[Pau14] Paul. Security Needs Context in IoT| SC Magazine. https ://securityledger.com/2014/11/security-needs-context-in-iot-sc-magazine/, November 2014.

[Pip14] Ankur Pipara. Internet of things (IoT). https ://www.slideshare.net/AnkurPipara/internet-of-things-iot-2014, July 2014.

[Pra17] M Prabhu. The INTERNET OF THINGS (IoT) | LinkedIn. https ://www.linkedin.com/pulse/internet-things-iot-prabhu-m/, June 2017.

[PRZR19] Ioan Petri, Omer Rana, Ali Reza Zamani, and Yacine Rezgui. Edge-Cloud Orchestration : Strategies for Service Placement and Enactment. In *2019 IEEE International Conference on Cloud Engineering (IC2E)*, pages 67–75, Prague, Czech Republic, June 2019. IEEE.

[PSL+19] Jordi Paillisse, Jordi Subira, Albert Lopez, Alberto Rodriguez-Natal, Vina Ermagan, Fabio Maino, and Albert Cabellos. Distributed Access Control with Blockchain. *arXiv :1901.03568 [cs]*, January 2019.

[Puj19] Guy Pujolle. *Software Networks : Virtualization, Sdn, 5g and Security*. ISTE Ltd / John Wiley and Sons Inc, Hoboken, 2nd edition edition, 2019.

[PZCG21] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context Aware Computing for The Internet of Things : A Survey. *IEEE Communications Surveys & Tutorials*, 16(1) :414–454, 21.

[QKCK14] Minh Thao Quach, Francine Krief, Mohamed Aymen Chalouf, and Hicham Khalifé. Fuzzy-based interference level estimation in cognitive radio networks. In *The Tenth Advanced International Conference on Telecommunications, AICT. IARIA XPS*, 2014.

[RBS15] Jose L. Hernandez Ramos, Jorge Bernal Bernabe, and Antonio F. Skarmeta. Managing Context Information for Adaptive Security in IoT Environments. In *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, pages 676–681, Gwangiu, South Korea, March 2015. IEEE.

[Ref14] Référentiel Général de Sécurité. Mécanismes Cryptographiques Annexe B1 Ver.2, Agence nationale de la sécuritédes systèmes d'information, February 2014.

[RJJAM+20] Abdul Rehman Javed, Zunera Jalil, Syed Atif Moqurrab, Sidra Abbas, and Xuan Liu. Ensemble Adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles. *Transactions on Emerging Telecommunications Technologies*, August 2020.

[RLA17] Said Radouche, Cherkaoui Leghris, and Abdellah Adib. MADM methods based on utility function and reputation for access network selection in a multi-access mobile network environment. In *2017 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pages 1–6, Rabat, November 2017. IEEE.

[RMM+20] Swarna Priya R.M., Praveen Kumar Reddy Maddikunta, Parimala M., Srinivas Koppu, Thippa Reddy Gadekallu, Chiranji Lal Chowdhary, and Mamoun Alazab. An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture. *Computer Communications*, 160 :139–149, July 2020.

[RNWK14] Arkham Zahri Rakhman, Lukito Edi Nugroho, Widyawan, and Kurnianingsih. Fall detection system using accelerometer and gyroscope based on smartphone. In *2014 The 1st International Conference on Information Technology, Computer, and Electrical Engineering*, pages 99–104, Semarang, Indonesia, November 2014. IEEE.

[Ros97] E.S. Rosenbloom. A probabilistic interpretation of the final rankings in AHP. *European Journal of Operational Research*, 96(2) :371–378, January 1997.

[RZH+19] Ju Ren, Deyu Zhang, Shiwen He, Yaoxue Zhang, and Tao Li. A survey on end-edge-cloud orchestrated network computing paradigms : Trans-

parent computing, mobile edge computing, fog computing, and cloudlet. *ACM Computing Surveys (CSUR)*, 52(6) :1–36, 2019.

[RZH+20] Ju Ren, Deyu Zhang, Shiwen He, Yaoxue Zhang, and Tao Li. A Survey on End-Edge-Cloud Orchestrated Network Computing Paradigms : Transparent Computing, Mobile Edge Computing, Fog Computing, and Cloudlet. *ACM Computing Surveys*, 52(6) :1–36, January 2020.

[SA09] Silvia Schiaffino and Analía Amandi. Intelligent User Profiling. In Max Bramer, editor, *Artificial Intelligence An International Perspective*, volume 5640, pages 193–216. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[SAL+19] N Simoni, Tatiana Aubonnet, Frédéric Lemoine, A Cadzow, B Dupré, and J Monfort. User Group ; User Centric Approach : Guidance for users ; Best practices to interact in the Digital Ecosystem. Technical Report EG 203 602, European Telecommunications Standards Institute (ETSI), France, April 2019.

[Sas18] Rami Sass. AutoSploit : Mass Exploitation Just Got a Lot Easier. https ://www.darkreading.com/threat-intelligence/autosploit-mass-exploitation-just-got-a-lot-easier-/a/d-id/1330982, February 2018.

[SCA16] SCA. Embedded Hardware Security for IoT Applications. White Paper IoTSC-16001, Smart Card Alliance, Princeton Junction, December 2016.

[SCFS21] Tidiane Sylla, Mohamed Aymeb Chalouf, Krief Francine, and Karim Samake. Context-Aware Security in the Internet of Things : A survey. *International Journal of Autonomous and Adaptive Communications Systems*, 14(3) :1, 2021.

[Sci] SciKit-Fuzzy — skfuzzy v0.2 docs. https ://pythonhosted.org/scikit-fuzzy/overview.html.

[SCKS20] Tidiane Sylla, Mohamed Aymen Chalouf, Francine Krief, and Karim Samaké. Towards a Context-Aware Security and Privacy as a Service in

the Internet of Things. In Maryline Laurent and Thanassis Giannetsos, editors, *Information Security Theory and Practice*, volume 12024, pages 240–252. Springer International Publishing, Cham, 2020.

[SCKS21] Tidiane Sylla, Mohamed Aymen Chalouf, Francine Krief, and Karim Samaké. SETUCOM : Secure and Trustworthy Context Management for Context-Aware Security and Privacy in the Internet of Things. *Security and Communication Networks*, 2021 :1–24, April 2021.

[SDK18] Jim Shaad, Roman Danyliw, and Benjamin Kaduk. Authentication and Authorization for Constrained Environments (ACE). https ://tools.ietf.org/html/draft-ietf-ace-oauth-authz-16, March 2018.

[SDL20] Farah Aït Salaht, Frédéric Desprez, and Adrien Lebre. An Overview of Service Placement Problem in Fog and Edge Computing. *ACM Computing Surveys*, 53(3) :1–35, July 2020.

[SFL17] Lei Shi, Xi Fu, and Jing Li. Mobility Prediction-Based Service Scheduling Optimization Algorithm in Cloudlets. In Xingming Sun, Han-Chieh Chao, Xingang You, and Elisa Bertino, editors, *Cloud Computing and Security*, volume 10603, pages 619–630. Springer International Publishing, Cham, 2017.

[SGFW10] Harald Sundmaeker, Patrick Guillemin, Peter Friess, and Sylvie Woelfflé. *Vision and Challenges for Realising the Internet of Things*. Publications Office of the European Union, Luxembourg, 2010.

[SHB14] Z. Shelby, K. Hartke, and C. Bormann. The Constrained Application Protocol (CoAP). https ://www.rfc-editor.org/info/rfc7252, June 2014.

[SHB+17] Hossein Shafagh, Anwar Hithnawi, Lukas Burkhalter, Pascal Fischli, and Simon Duquennoy. Secure Sharing of Partially Homomorphic Encrypted IoT Data. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, pages 1–14, Delft Netherlands, November 2017. ACM.

[SHC+01] Mark Scherling, An-Ni Huynh, Mark Carlson, John Strassner, Steve Waldbusser, Shai Herzog, Andrea Westerinen, Jay Perry, Bob Quinn, and John Schnizlein. Terminology for Policy-Based Management. https ://tools.ietf.org/html/rfc3198#page-17, November 2001.

[SKWT17] Michael Seufert, Brice Kamneng Kwam, Florian Wamser, and Phuoc Tran-Gia. Edgenetworkcloudsim : Placement of service chains in edge clouds using networkcloudsim. In *2017 IEEE Conference on Network Softwarization (NetSoft)*, pages 1–6, Bologna, Italy, July 2017. IEEE.

[SMP19] G. Selander, J. Mattsson, and F. Palombini. Ephemeral Diffie-Hellman Over COSE (EDHOC). https ://datatracker.ietf.org/doc/html/draft-selander-ace-cose-ecdhe-07, September 2019.

[SMPS19] G. Selander, J. Mattsson, F. Palombini, and L. Seitz. Object Security for Constrained RESTful Environments (OSCORE). Technical Report RFC8613, RFC Editor, July 2019.

[SNS16] C. Schmitt, M. Noack, and B. Stiller. TinyTO : Two-way authentication for constrained devices in the Internet of Things. In *Internet of Things*, pages 239–258. Elsevier, 2016.

[SOE18] Cagatay Sonmez, Atay Ozgovde, and Cem Ersoy. EdgeCloudSim : An environment for performance evaluation of edge computing systems : EdgeCloudSim : An environment for performance evaluation of edge computing systems. *Transactions on Emerging Telecommunications Technologies*, 29(11) :e3493, November 2018.

[SOE19] Cagatay Sonmez, Atay Ozgovde, and Cem Ersoy. Fuzzy Workload Orchestration for Edge Computing. *IEEE Transactions on Network and Service Management*, 16(2) :769–782, June 2019.

[Sol15] Cisco Fog Computing Solutions. Unleash the power of the Internet of Things. *Cisco Systems Inc*, 2015.

[SPC+17] Savio Sciancalepore, Giuseppe Piro, Daniele Caldarola, Gennaro Boggia, and Giuseppe Bianchi. OAuth-IoT : An access control framework for the Internet of Things based on open standards. In *2017 IEEE Symposium on Computers and Communications (ISCC)*, pages 676–681, Heraklion, Greece, July 2017. IEEE.

[SS17] Pallavi Sethi and Smruti R. Sarangi. Internet of Things : Architectures, Protocols, and Applications. *Journal of Electrical and Computer Engineering*, 2017 :1–25, 2017.

[SSHY21] Saurabh Singh, A.S.M. Sanwar Hosen, and Byungun Yoon. Blockchain Security Attacks, Challenges, and Solutions for the Future Distributed IoT Network. *IEEE Access*, pages 1–1, 2021.

[SSMP17] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, and Jong Hyuk Park. Advanced lightweight encryption algorithms for IoT devices : Survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing*, May 2017.

[ST94] B.N. Schilit and M.M. Theimer. Disseminating active map information to mobile hosts. *IEEE Network*, 8(5) :22–32, September 1994.

[STB+20] Muhammad Shafiq, Zhihong Tian, Ali Kashif Bashir, Xiaojiang Du, and Mohsen Guizani. CorrAUC : A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine Learning Techniques. *IEEE Internet of Things Journal*, pages 1–1, 2020.

[Suc13] Jose M. Such. Attacks and Vulnerabilities of Trust and Reputation Models. In Sascha Ossowski, editor, *Agreement Technologies*, pages 467–477. Springer Netherlands, Dordrecht, 2013.

[Sun16] B Shanmuga Sundaram. A quantitative analysis of 802.11 ah wireless standard. *International Journal of Latest Research in Engineering and Technology*, 2, 2016.

[Sve19] Oleg Svet. IoT : User-Centric, Privacy Security - DZone IoT. https ://dzone.com/articles/iot-user-centric-privacy-security, May 2019.

[SVK+16] Dario Sabella, Alessandro Vaillant, Pekka Kuure, Uwe Rauschenbach, and Fabio Giust. Mobile-edge computing architecture : The role of MEC in the Internet of Things. *IEEE Consumer Electronics Magazine*, 5(4) :84–91, 2016.

[SWH17] Rashmi Sharan Sinha, Yiqiao Wei, and Seung-Hoon Hwang. A survey on LPWA technology : LoRa and NB-IoT. *ICT Express*, 3(1) :14–21, March 2017.

[SWYS10] Hongbo Si, Yue Wang, Jian Yuan, and Xiuming Shan. Mobility Prediction in Cellular Network Using Hidden Markov Model. In *2010 7th IEEE Consumer Communications and Networking Conference*, pages 1–5, Las Vegas, NV, USA, January 2010. IEEE.

[TBME19] Reza Tourani, Austin Bos, Satyajayant Misra, and Flavio Esposito. Towards security-as-a-service in multi-access edge. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 358–363, Arlington Virginia, November 2019. ACM.

[tea] aGrUM team. aGrUM/pyAgrum. https ://agrum.gitlab.io/.

[TF16] H. Tschofenig and T. Fossati. Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things. Technical Report RFC7925, RFC Editor, July 2016.

[TKSA16] Ilaria Torre, Frosina Koceva, Odnan Ref Sanchez, and Giovanni Adorni. A framework for personal data protection in the IoT. In *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 384–391, Barcelona, Spain, December 2016. IEEE.

[TP18] Punnarumol Temdee and Ramjee Prasad. *Security for Context-Aware Applications*, pages 97–125. Springer International Publishing, Cham, 2018.

[TSM+23] Tarik Taleb, Konstantinos Samdanis, Badr Mada, Hannu Flinck, Sunny Dutta, and Dario Sabella. On Multi-Access Edge Computing : A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. *IEEE Communications Surveys & Tutorials*, 19(3) :1657–1681, 23.

[TTC17] Michal Trnka, Martin Tomasek, and Tomas Cerny. Context-Aware Security Using Internet of Things Devices. In Kuinam Kim and Nikolai Joukov, editors, *Information Science and Applications 2017*, volume 424, pages 706–713. Springer Singapore, Singapore, 2017.

[TTHG+17] Amy J.C. Trappey, Charles V. Trappey, Usharani Hareesh Govindarajan, Allen C. Chuang, and John J. Sun. A review of essential standards and patent landscapes for the Internet of Things : A key enabler for Industry 4.0. *Advanced Engineering Informatics*, 33 :208–229, August 2017.

[TYMR17] Slavica Tomovic, Kenji Yoshigoe, Ivo Maljevic, and Igor Radusinovic. Software-Defined Fog Network Architecture for IoT. *Wireless Personal Communications*, 92(1) :181–196, January 2017.

[uRKI+21] Saif ur Rehman, Mubashir Khaliq, Syed Ibrahim Imtiaz, Aamir Rasool, Muhammad Shafiq, Abdul Rehman Javed, Zunera Jalil, and Ali Kashif Bashir. DIDDOS : An approach for detection and identification of Distributed Denial of Service (DDoS) cyberattacks using Gated Recurrent Units (GRU). *Future Generation Computer Systems*, 118 :453–466, May 2021.

[VACM17] Karima Velasquez, David Perez Abreu, Marilia Curado, and Edmundo Monteiro. Service placement for latency reduction in the internet of things. *Annals of Telecommunications*, 72(1-2) :105–115, February 2017.

[VAH+11] J Vasseur, Navneet Agarwal, Jonathan Hui, Zach Shelby, Paul Bertrand, and Cedric Chauvenet. RPL : The IP routing protocol designed for low power and lossy networks. *Internet Protocol for Smart Objects (IPSO) Alliance*, 36, 2011.

[VKK+18] Jeffrey Voas, Rick Kuhn, Constantinos Kolias, Angelos Stavrou, and Georgios Kambourakis. Cybertrust in the IoT Age. *Computer*, 51(7) :12–15, July 2018.

[VLG+17] Ricard Vilalta, Victor Lopez, Alessio Giorgetti, Shuping Peng, Vittorio Orsini, Luis Velasco, Rene Serral-Gracia, Donal Morris, Silvia De Fina, Filippo Cugini, Piero Castoldi, Arturo Mayoral, Ramon Casellas, Ricardo Martinez, Christos Verikoukis, and Raul Munoz. TelcoFog : A Unified Flexible Fog and Cloud Computing Architecture for 5G Networks. *IEEE Communications Magazine*, 55(8) :36–43, August 2017.

[VTR+15] Mališa Vučinić, Bernard Tourancheau, Franck Rousseau, Andrzej Duda, Laurent Damon, and Roberto Guizzetti. OSCAR : Object security architecture for the Internet of Things. *Ad Hoc Networks*, 32 :3–16, September 2015.

[WK21] Lusheng Wang and Geng-Sheng G.S. Kuo. Mathematical Modeling for Network Selection in Heterogeneous Wireless Networks — A Tutorial. *IEEE Communications Surveys & Tutorials*, 15(1) :271–292, 21.

[WM17] Xiaonan Wang and Yi Mu. Communication security and privacy support in 6LoWPAN. *Journal of Information Security and Applications*, 34 :108–119, June 2017.

[Woo16] Nicky Woolf. DDoS attack that disrupted internet was largest of its kind in history, experts say, October 2016.

[WTB+12] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, and R. Alexander. RPL : IPv6 Routing Protocol for Low-Power and Lossy Networks. https ://tools.ietf.org/html/rfc6550, March 2012.

[XTH+19] Junfeng Xie, Helen Tang, Tao Huang, F. Richard Yu, Renchao Xie, Jiang Liu, and Yunjie Liu. A Survey of Blockchain Technology Applied to Smart

Cities : Research Issues and Challenges. *IEEE Communications Surveys & Tutorials*, pages 1–1, 2019.

[YCT15] Xuanxia Yao, Zhi Chen, and Ye Tian. A lightweight attribute-based encryption scheme for the Internet of Things. *Future Generation Computer Systems*, 49 :104–112, August 2015.

[YZL+16] Zhe Yang, Qihao Zhou, Lei Lei, Kan Zheng, and Wei Xiang. An IoT-cloud Based Wearable ECG Monitoring System for Smart Healthcare. *Journal of Medical Systems*, 40(12), December 2016.

[YZV14] Zheng Yan, Peng Zhang, and Athanasios V. Vasilakos. A survey on trust management for Internet of Things. *Journal of Network and Computer Applications*, 42 :120–134, June 2014.

[ZAR21] Moubarak Zoure, Toufik Ahmed, and Laurent Réveillère. VeriNeS : Runtime verification of outsourced network services orchestration. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pages 1138–1146, Virtual Event Republic of Korea, March 2021. ACM.

[ZDWM17] Junqing Zhang, Trung Duong, Roger Woods, and Alan Marshall. Securing Wireless Communications of the Internet of Things from the Physical Layer, An Overview. *Entropy*, 19(8) :420, August 2017.

[ZKS+19] Yuanyu Zhang, Shoji Kasahara, Yulong Shen, Xiaohong Jiang, and Jianxiong Wan. Smart Contract-Based Access Control for the Internet of Things. *IEEE Internet of Things Journal*, 6(2) :1594–1605, April 2019.

[ZLG+20] Jinxin Zuo, Yueming Lu, Hui Gao, Ruohan Cao, Ziyv Guo, and Jim Feng. Comprehensive Information Security Evaluation Model Based on Multi-Level Decomposition Feedback for IoT. *Computers, Materials & Continua*, 65(1) :683–704, 2020.

[ZM17] Almudena Diaz Zayas and Pedro Merino. The 3GPP NB-IoT system architecture for the Internet of Things. In *2017 IEEE International Conference*

*on Communications Workshops (ICC Workshops)*, pages 277–282, Paris, France, May 2017. IEEE.

[ZXD+18] Zibin Zheng, Shaoan Xie, Hong Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities : A survey. *International Journal of Web and Grid Services*, 14(4) :352, 2018.

# Acronymes

| Acronyme | Signification |
|---|---|
| 3GPP | Third Generation Partnership Project |
| 6LoWPAN | IPv6 for Low-power Wireless Personal Area Networks |
| ABI | Application Binary Interface |
| AC | Acquisition de Contexte |
| ACE | Authentication and Authorization for Constrained Environments |
| ACL | Access Control List |
| ADSL | Asymmetric Digital Subscriber Line |
| AES | Advanced Encryption Standard |
| AHP | Analytic Hierarchy Processor |
| ANSSI | Agence Nationale de la Sécurité des Systèmes d'Information |
| AOS | Architecture Orientée Service |
| API | Application Programing Interface |
| APSC | Applicateur des Politiques de Sécurité Contextuelles |
| ARIB | Association of Radio Industries and Businesses, Japon |
| ATIS | Alliance for Telecommunications Industry Solutions |
| BC | Base de Contextes |
| BIC | Base d'Information Contextuelle |
| BLE | Bluetooth Low Energy |
| BPSC | Base de Politiques de Sécurité Contextuelle |
| CAABAC | Context-Aware Attribute-Based Access Control |
| CASA | Context-Aware Scalable Architecture |
| CASFMA | Context-Aware Security Framework for Mobiles Applications |
| CASM | Context-Aware Security Manager |
| CASPaaS | Context-Aware Security and Privacy as a Service |
| CASSH | Context-Aware Authentication Service for Smart Homes |
| CBOR | Consise Binary Object Representation |
| CBSPHIoT | Context-Based Security and Privacy for Healthcare IoT |
| CCSA | China Communications Standards Association |
| CoAP | Constrained Application Protocol |
| CoRE | Constrained RESTful Environments |
| COSE | Concise Object Signing and Encryption |
| CP-ABE | Ciphertext-Policy Attribute Based Encryption |
| CRBAC | Contextual Role Based Access Control |
| CWT | CBOR Web Token (CWT) |
| DC | Distributeur de Contexte |
| DCASTBISPF | Dynamic Context-Aware Scalable and Trust-Based IoT Security, Privacy Framework |
| DDoS | Distributed Denial of Service |
| DODAG | Destination Oriented Directed Acyclic Graph |
| DoS | Denial of Service |

| Acronyme | Signification |
|---|---|
| DTLS | Datagram Transport Layer Security |
| ECC | Elliptic Curve Cryptography |
| ECCAPAC | Enhanced Context-aware Capability based Access Control |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| ECG | Electrocardiogramme |
| ECIES | Elliptic Curve Integrated Encryption Scheme |
| ECSA | Edge-centric Context Sharing Architecture |
| EDHOC | Ephemeral Diffie Hellman over Cose |
| eMTC | Machine-Type Communication |
| EPR | Electronic Patient Record |
| ETSI | European Telecommunication Standard Institute |
| EXI | Efficient XML Interchanges |
| GCD | Gestionnaire de Confiance des Dispositifs |
| GER | Gestionnaire de l'Evaluation des Risques |
| GPS | Global Positioning System |
| GPSC | Gestionnaire des Politiques de Sécurité Contextuelles |
| GPU | Gestion des Préférences Utilisateur |
| GRU | Gate Recurrent Unit |
| GSPC | Gestion de la Sécurité et de la protection de la vie Privée Contextuelles |
| HIoT | Healthcare Internet of Things |
| HMAC | Keyed-Hash Message Authentication Code |
| HMM | Hidden Markov Model |
| HSDPA | High Speed Downlink Packet Access |
| HSPA | High Speed Packet Access |
| HSUPA | High-Speed Uplink Packet Access |
| HTTP | HyperText Transfer Protocol |
| IDS | Intrusion Detection System |
| IEC | International Electrotechnical Commission |
| IEEE | International Electrical and Electronics Engineers |
| IETF | Internet Engineering Task Force |
| IIC | Industrial Internet Consortium |
| IIRA | Industrial Internet Reference Architecture |
| IMT | International Mobile Telecommunication |
| IoT | Internet of Things |
| IoV | Internet of Vehicles |
| IPv4 | Internet Protocol version 4 |
| IPv6 | Internet Protocol version 6 |
| ISO | International Organization for Standardization |
| ITS | Intelligent Transport System |
| JOSE | JSON Object Signing and Encryption |
| JSON | Javascript Object Notation |
| JTC1 | Joint Technical Committee 1 |

| Acronyme | Signification |
|---|---|
| JWT | JSON Web Token |
| KDF | Key Derivation Function |
| LAN | Local Area Network |
| LLN | Low power and Lossy Networks |
| LoRaWAN | Long Range Wide Area Network |
| LPWAN | Low Power Wide Area Network |
| LR-WPAN | Low Rate – Wireless Personal Area Networks |
| LTE | Long-Term Evolution |
| LTE-A | Long-Term Evolution Advanced |
| LTE-M | Long-Term Evolution for Machines |
| M2M | Machine-To-Machine |
| MAC | Medium Access Control |
| MAC | Message Authentication Code |
| MAN | Metropolitan Area Network |
| MANO | Management and Orchestration |
| MC | Modélisation du Contexte |
| MCIASIoTE | Managing Context Information for Adaptive Security in IoT Environments |
| MEC | Multi Access Mobile Edge Computing |
| MIT | Massachusetts Institute of Technology |
| MITM | Man In The Middle |
| MQTT | Message Queue Telemetry Transport |
| MTU | Maximum Transmission Unit |
| NB-IoT | Narrowband Internet of Things |
| NFC | Near Field Communication |
| NFV | Network Function Virtualization |
| NFVI | NFV Infrastructure |
| NIST | National Institute of Standards and Technology |
| NR | New Radio |
| OAUTH | Open Authorization |
| ONF | Open Networking Foundation |
| OSCAR | Object Security Architecture |
| OSCORE | Object Security for Constrained RESTful Environments |
| OTA | On The Air |
| OWASP | Open Web Application Security Project |
| PC | Plan de Connaissance |
| PIN | Personal Identification Number |
| PMC | Pervasive Mobile Computing |
| PoP | Proof of Possession |
| PPSFSAIoT | Privacy Preserving Security Framework for a Social-Aware Internet of Things |
| PSC | Politique de Sécurité Contextuelle |
| PSPVP | Plan de Sécurité et de Protection de la Vie Privée |

| Acronyme | Signification |
|---|---|
| QoC | Quality of Context |
| QoE | Quality of Experience |
| QoS | Quality of Service |
| RAN | Radio Access Network |
| REST | REpresentational State Transfer |
| RFC | Request For Comments |
| RFID | Radio Frequency Identifier |
| RGPD | Règlement Général sur la Protection des Données |
| ROLL | Routing over low-power and Lossy networks |
| RPC | Raisonnement et Prédiction du Contexte |
| RPL | Routing Protocol for Low-power and lossy networks |
| RSU | Road Side Unit |
| SDN | Software Defined Networking |
| SETUCOM | SEcure and TrUstworthy COntext Management |
| SGX | Software Gard Extension |
| SLA | Service Level Agreement |
| SOA | Service Oriented Architecture |
| SRA | Sécurité du Réseau et de l'Architecture |
| SSL | Secure Socket Layer |
| SSP | Services de Sécurité et protection de la vie Privée |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| TIA | Telecommunications Industry Association |
| TIC | Technologies de l'Information et de la Communication |
| TLS | Transport Layer Security |
| TSDSI | Telecommunications Standards Development Society |
| TTA | Telecommunications Technology Association |
| TTC | Telecommunication Technology Committee, Japon |
| UDP | User Datagram Protocol |
| UIT | Union Internationale des Télécommunications |
| UIT-T | Branche de l'UIT chargée de la normalisation du secteur des télécommunications |
| UMTS | Universal Mobile Telecommunications System |
| UN DESA | United Nation Department of Economic and Social Affairs |
| VIM | Virtualized Infrastructure Manager |
| VNF | Virtualised Network Functions |
| VPN | Virtual Private Network |
| WAN | Wide Area Network |
| WBAN | Wireless Body Area Network |
| WOL | Web Ontology Language |
| WPAN | Wireless Personal Area Network |
| WSN | Wireless Sensor Network |
| XACML | eXtensibe Access Control Markup Language |