

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre :.....



THÈSE

Présentée en vue de l'obtention du diplôme de
DOCTORAT LMD EN INFORMATIQUE

Spécialité : INTELLIGENCE ARTIFICIELLE

Titre

Une approche cloud computing basée IoT pour le smart House

Présentée par :
Meftah ZOUAI

Soutenue devant le jury composé de :

Président :	Mohamed Benmohammed	Professeur	Université de constantine 2
Rapporteur :	Okba Kazar	Professeur	Université de Biskra
Co-rapporteur :	Belgacem haba	Professeur	Invensas Corp, USA
Examineurs :	Samir Bourekkache	M.C.A	Université de Biskra
	Soheyb Ayad	M.C.A	Université de Biskra

Année universitaire : **2019 – 2020**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Remerciements

*En premier lieu, ma profonde gratitude et mes sincères remerciements vont à mon directeur de thèse monsieur le Professeur **Okba KAZAR** pour m'avoir proposé ce sujet et m'avoir dirigé, conseillé et encouragé tout au long de la réalisation de ce travail et surtout pour sa patience et sa compréhension afin de donner le fruit d'un travail long et difficile. Veuillez croire à l'expression de ma grande admiration et mon profond respect.*

*Je remercie également le Professeur **Belgacem Haba** d'avoir suivi mon travail et aussi pour ses précieux conseils.*

*Je tiens à remercier le Professeur **Guadalupe Ortiz** du Laboratoire UCASE, Cádiz en Espagne de m'avoir accueilli dans son Laboratoire ainsi que pour l'attention qu'elle m'a apportée pendant mon séjour et aussi pour la collaboration scientifique et pour ses précieux conseils.*

*Mes vifs remerciements vont aussi au Professeur **Nadia Kabachi** de l'université Lyon 1 pour m'avoir accueilli dans le laboratoire ERIC et pour la collaboration scientifique.*

*Je tiens à remercier sincèrement le président de jury le Professeur **Mohamed Benmohamed** de l'université de Constantine 2, qui m'a fait le grand honneur de présider ce jury.*

*Mes remerciements vont aussi aux membres du jury, le docteur **Samir Bouekkache**, maître de conférences A de l'université de Biskra et le docteur **Soheyb Ayad**, maître de conférences A de l'université de Biskra, pour m'avoir fait l'honneur de participer à mon jury de soutenance en tant qu'examineurs.*

*Je ne peux oublier de remercier les enseignants, le docteur **Abdelhak Merizig**, le docteur **Houcine Belouaar** et le docteur **Hamza Saouli** du laboratoire LINFI pour leur aide précieuse pendant la réalisation de cette thèse.*

Meftah Zouai

Dédicaces

Je dédie ce modeste travail :

À mes parents, que le Dieu me les gardes.

À mes chers et adorable frères et sœurs.

À ma chère épouse.

À mes chers petits neveux et nieces.

À tous mes amis.

À la mémoire de mon ami Bouzidi Djouher.

ملخص

قام الإنسان القديم بإنشاء أكواخ (بيوت من شجر و جلد) لحمايته من العوامل الطبيعية و الحيوانات المفترسة لحاجته للأمن ثم تطورت هاته الاكواخ حسب حاجيات الإنسان و المواد المتوفرة المستخدمة للبناء الى ان وصلنا الى شكل المنازل الحالي و مع بداية ظهور الآلات الكهرومنزلية اصبحت المنازل مزودة بالكثير من الاجهزة الالكترونية و الكهربائية و بعد ان أصبحت الانترنت في السنوات الاخيرة عنصرا رئيسيا في كل البيوت مع وجود كل هاته العوامل التكنولوجية و البنية التحتية تطورت حاجيات الانسان و اصبح يفكر في إنشاء منازل الية تماما توفر له بيئة أكثر راحة و أمان و بهذا ظهر مصطلح "المنازل الذكية"

المنازل الذكية هو بيت تحكمه التكنولوجيا بالكامل و يتم التحكم به و إدارته من خلال مجموعة من الأفعال و اجهزة التحكم عن بعد (الهواتف الذكية و الحواسيب) و يتم ادارة المنزل بالكامل من خلال شبكة الانترنت أو الشبكة المحلية اللاسلكية. و بظهور مفهوم إنترنت الأشياء و الحوسبة السحابية اخذت المنازل الذكية خصائص جديدة مثل: المنزل الذكي التفاعلي ، و اصبحت محتويات هاته المنازل تمتاز بالاستقلالية القرار ، و الذكاء و التفاعل مع المحيط من خلال مستشعرات و اجهزة الكترونية دقيقة مدمجة.

من خلال هاته المذكرة سنقوم بدراسة مشكلة التواصل بين مكونات المنزل بحيث لا تصبح متصلة بالانترنت فقط بل تصبح لديها القدرة على التواصل مع بعضها البعض و تتفاعل مع المتغيرات التي تحدث في البيئة المحيطة بصفة تلقائية من خلال استخدام نظام متعدد الوكلاء المتنقلين و المستقرين، حيث قمنا بتقديم بنية جديدة للكائنات (الأشياء) من خلال إضافة طبقة الوكيل إلى البنية القاعدية للكائن، تمنح هاته الطبقة للكائن القدرة على التواصل و التعاون و التنسيق و مشاركة المعلومات و المعارف و تمنحه درجة من الذكاء و استقلالية القرار، إن ذكاء الجماعي الموزع بين الكائنات ينشئ نظام ذكي قادر على اتخاذ قرارات صحيحة و بصفة مستقلة.

Abstract

In antiquity, human beings created shelters of woods, stones, and mud to protect themselves from natural factors and predatory animals. Then they developed them according to their needs and available materials until they got to their current shape. Nowadays, houses became equipped with a lot of electronic devices, and the Internet has become indispensable in all homes. These caused humans to think of creating fully automated homes to provide a safer and more comfortable lifestyle. This is where the term as **smart homes** was introduced.

The smart house is a house that is fully controlled by a group of remote-control devices (smartphones and computers) and managed through the Internet or a wireless local network. With the advent of **Internet of Things** and **cloud computing** concepts, smart homes have taken on new features such as interactive smart houses. The components of these houses became more independent in decision-making, intelligent, and interactive with the environment through sensors and built-in microelectronic devices.

Through this thesis, we will study the problems of communication between the components of houses and the Internet, the interaction between these components, and the changes that occur in the surrounding environment automatically through the use of multi-agents (mobile and fixed) system. We will present a new architecture for the objects (things) by adding the **agent layer** to their base architecture. This layer gives them the ability to communicate, cooperate, coordinate, and share information and knowledge. It also grants them **independence** in decision-making and a degree of **intelligence**. The **collective intelligence distributed** among the objects creates an intelligent system capable of making correct and independent decisions.

Résumé

A l'antiquité, l'homme a créé des huttes (maisons d'arbres et de peau) pour le protéger des facteurs naturels et des animaux prédateurs pour répondre aux ses besoins de sécurité, puis ces huttes se sont développées en fonction des besoins humains et matériaux disponibles de construction jusqu'à l'apparition des maisons actuels. Les maisons de nos jours sont devenues trop équipées des appareils électroniques et électriques avec l'émergence des machines domestiques, et avec l'immense diffusion de l'internet, avec tous ces facteurs technologiques et infrastructures, les besoins humains ont évolué et il est devenu possible de créer des maisons entièrement automatisées qui fournissent un environnement plus confortable et plus sûr. Le couplage de ces technologies avec l'intelligence artificielle se donné naissance à un nouveau modèle de maison dite **maison intelligente**.

La maison intelligente est une maison entièrement contrôlée et gérée par la technologie à travers un groupe de serrures et de dispositifs de télécommande (smartphones et ordinateurs). La maison est entièrement gérée via Internet ou un réseau local sans fil. Avec l'avènement du concept de l'**Internet des objets** et du **cloud computing**, les maisons intelligentes ont adopté de nouvelles fonctionnalités telles que : **la maison intelligente interactive**, et le contenu de ces maisons est devenu une prise de décision indépendante, une intelligence et une interaction avec l'environnement via des capteurs et dispositifs microélectroniques intégrés.

Dans cette thèse, nous avons étudié le problème de la communication entre les composants de la maison afin qu'ils ne se connectent pas seulement à Internet mais aient également la capacité de communiquer entre eux et d'interagir automatiquement avec les changements qui se produisent dans l'environnement. Grâce à l'utilisation d'un système multi-agents (mobiles et située), où nous avons présenté une nouvelle architecture pour les objets en ajoutant **la couche d'agent** à la architecture de base de l'objet. Cette couche donne à l'objet la capacité de communiquer, de coopérer, coordonner et de partager des informations et des connaissances. Elle lui confère un **degré d'intelligence et d'indépendance de décision, l'intelligence collective répartie** entre les objets crée un système intelligent capable de prendre des décisions correctes et indépendantes.

Table des matières

Remerciements	ii
Dédicaces	iii
Résumés	vi
Introduction générale	1
1 Internet des Objets	5
1.1 Introduction	5
1.2 Internet des objets	6
1.3 Architecture de l'internet des objets	7
1.3.1 Couche de perception	8
1.3.2 Couche réseau	8
1.3.3 Couche d'application	8
1.4 Objet connecté	8
1.4.1 Types d'informations	9
1.4.2 Types des objets connectés	10
1.4.2.1 Les objets portés	10
1.4.2.2 Les objets indépendants	10
1.4.2.3 Les objets passifs	10
1.4.2.4 Les objets actifs	11
1.4.3 Capteurs et actionneurs	11
1.4.3.1 Capteurs	11
1.4.3.2 Actionneur	14
1.4.4 Caractéristiques fondamentales de l'internet des objets	15

1.4.5	Axes de l'internet des objets	15
1.5	Domaine d'application de l'internet des objets	17
1.5.1	Maison intelligente	17
1.5.2	La santé	17
1.5.3	Transport	18
1.5.4	Énergie	18
1.5.5	Industrie	18
1.5.6	Environnement	19
1.6	Technologies de communication	20
1.6.1	RFID et NFC (Near-Field Communication)	20
1.6.2	Bluetooth Low Energy (BLE)	21
1.6.3	LiFi	22
1.6.4	6LowPAN	22
1.6.5	ZigBee	23
1.6.6	Z-Wave	23
1.6.7	LoRa	24
1.7	Protocoles de l'internet des objets	25
1.7.1	Hypertext Transfer Protocol	25
1.7.2	WebSocket	25
1.7.3	MQ Telemetry transport (MQTT)	26
1.7.4	Constrained application Protocole (CoAP)	26
1.7.5	Extensible Messaging and Presence Protocol (XMPP)	26
1.8	Système embarqué	26
1.8.1	Définition du système embarqué	27
1.8.1.1	Microcontroller	28
1.8.1.2	Les entrées	28
1.8.1.3	Les sorties	29
1.8.1.4	Interfaces de communication	29
1.8.2	Caractéristiques d'un système embarqué	29
1.8.3	Structure de base d'un système embarqué	30

1.8.4	Tendances futures des systèmes embarqués	31
1.9	Conclusion	31
2	Cloud Computing	32
2.1	Introduction	32
2.2	Définition du Cloud Computing	33
2.3	Caractéristiques du cloud computing	34
2.3.1	Abstraction sur la localisation des données	34
2.3.2	Souscription en ligne	35
2.3.3	Tarifcation « Freemium »	35
2.3.4	Accès depuis n'importe quel appareil	35
2.4	Techniques de virtualisation	35
2.4.1	Définition	36
2.4.2	Virtualisation	36
2.5	Services Cloud	39
2.5.1	Software as a Service (SaaS)	40
2.5.1.1	Les avantages	40
2.5.1.2	Les inconvénients	40
2.5.2	Platform as a Service (PaaS)	40
2.5.2.1	Les avantages	40
2.5.2.2	Les inconvénients	41
2.5.3	Infrastructure as a Service (IaaS)	41
2.5.3.1	Les avantages	41
2.5.3.2	Les inconvénients	41
2.6	Modèles de cloud computing	42
2.6.1	Cloud privé	42
2.6.2	Cloud public	43
2.6.3	Hybride cloud	43
2.6.4	Cloud communautaire	43
2.7	Avantages et inconvénients du Cloud Computing	44

2.8	Edge computing	46
2.8.1	Fondamentaux de l'edge computing	46
2.8.1.1	Edge	46
2.8.1.2	Edge device	46
2.8.1.3	La passerelle Edge	47
2.8.2	Edge computing et Fog computing	47
2.8.3	Domaines d'application pour les architectures d'edge computing	48
2.8.4	Avantages et inconvénients de l'edge computing	49
2.9	Conclusion	50
3	Approches de l'IoT dans la maison intelligente	52
3.1	Introduction	52
3.2	Historique	52
3.3	Définition de la maison intelligente	58
3.4	Travaux connexes	59
3.4.1	Travaux exploitant l'IoT dans le smart house sans le cloud	59
3.4.2	Travaux exploitant l'IoT dans le smart house via le cloud	60
3.4.3	Travaux sur les architecture des smart houses	61
3.4.4	Travaux sur les architecture des objets	62
3.4.5	Travaux sur les robots basés internet des objets	63
3.5	Conclusion	64
4	Modélisation de la nouvelle approche	66
4.1	Introduction	66
4.2	Architecture générale du système	67
4.2.1	Description des couches et des composants	68
4.2.1.1	Edge nodes	69
4.2.1.2	Fog nodes	70
4.2.1.3	Cloud Node	70
4.3	Architecture détaillée de système	71
4.3.1	IoT Gateway	71

4.3.2	Agent de sécurité	72
4.3.2.1	Diagramme d'état	73
4.3.3	Dispositif IoT	74
4.3.4	Agent de santé	75
4.3.5	La modélisation de l'IoT Robot	77
4.3.5.1	Description de IoT Robot	78
4.3.5.2	Diagramme d'état	79
4.3.6	Simulateur de maison intelligente	79
4.3.6.1	Conception de l'API Simulateur	80
4.3.6.1.1	Composantes de l'API	80
4.3.6.1.1.1	Interface utilisateur graphique (GUI) du simulateur :	80
4.3.6.1.1.2	interface de programmation d'application (API) :	81
4.3.6.1.1.3	User Application	81
4.4	Communications des composants de notre architecture	82
4.4.1	Communication des edge nœuds	82
4.4.2	Perspicacité et Communications sur les edge nœuds	83
4.5	Modèle de coopération utilisé	84
4.5.1	Communication entre les objets	85
4.5.1.1	Scénario de détection des voisinages :	86
4.5.1.2	Scénario de détection de défaut :	86
4.6	Conclusion	88
5	Résultat et mise en oeuvre	89
5.1	Introduction	89
5.2	Outils et plateformes utilisés	89
5.3	Plateforme JADE	90
5.3.1	Présentation générale	90
5.3.2	Architecture du logiciel	90
5.4	Simulateur smart house	91
5.4.1	Présentation des interfaces de simulateur	93

5.4.1.1	Lancement de l'interface graphique du simulateur	93
5.4.1.2	Manipulation des états IoT VD	95
5.4.1.3	Manipulation de l'IoT VS	96
5.4.1.4	Afficher le tableau des statuts	97
5.5	IoT Robot	98
5.6	IoT device	99
5.6.1	Composantes électroniques de IoT device	101
5.6.1.1	PIR Motion Sensor (HCSR501 PIR)	101
5.6.1.2	Capteur d'humidité et de température (DHT-11)	101
5.6.1.3	Raspberry Pi Model B	102
5.6.2	Câblage et circuit électronique	102
5.6.3	Fonctionnalités de IoT device	103
5.7	Conclusion	107
	Conclusion générale et perspectives	108
	A Liste des publications	110
A.1	Revue Internationale	110
A.2	Conférences Internationales	110
	Annexe	110
	Bibliographie	112

Table des figures

1.1	Architecture à trois couches de l'IoT [70]	7
1.2	Axes de l'internet des objets [70]	16
1.3	Révolutions industrielles [119]	19
1.4	Diagramme schématique d'un système embarqué typique	28
1.5	Structure de base d'un système embarqué	30
2.1	Architecture de l'isolation	37
2.2	Architecture de la paravirtualisation	37
2.3	Architecture de la virtualisation complète	38
2.4	Modèle de référence du cloud computing	39
2.5	Différents types de services de cloud computing	42
2.6	Modèles de déploiement cloud	44
3.1	Maisons de Néandertal	53
3.2	Maisons rondes de bois	53
3.3	Maisons rectangulaires ou carrées	54
3.4	Une grande ville méditerranéen	54
3.5	Modèle d'une maison gauloise	55
3.6	Maison de la 12 ^{ème} siècle	56
3.7	Cité ouvrière	56
3.8	Cité du 20 ^{ème} siècle	57
4.1	Architecture en couches du système	69
4.2	Architecture proposée IoT Gateway	72

4.3	Architecture concrète de l'agent de sécurité	73
4.4	Diagramme d'état de l'agent de sécurité	74
4.5	Architecture IoT Device	75
4.6	Architecture concrète de l'agent «Health Care»	76
4.7	Modules d'IoT Robot	78
4.8	Diagramme d'état de IoT Robot	79
4.9	Architecture de l'API du simulateur	80
4.10	Edges nœuds et de leur communication avec le fog	83
4.11	Architecture proposée pour le fonctionnement global	85
4.12	Diagramme de séquence de détection des voisins	86
4.13	diagramme de séquence de détection des désactive et échoue	87
5.1	LinFiSim GUI	94
5.2	IoT VD status.	96
5.3	API du simulateur avec la table d'état.	98
5.4	Robot IoT	99
5.5	Circuit électronique de IoT device	103
5.6	Agent sécurité IoT détecte la présence d'une personne à la maison.	104
5.7	Communication entre agents	105
5.8	Réaction de l'agent sécurité IoT	106
5.9	Agent de sécurité IoT envoie un message à l'utilisateur.	107

Introduction Générale

Contexte du travail

L'Internet des Objets ou Internet of things (IoT) se définit comme un réseau mondial de services interconnectés et d'objets intelligents de toutes natures destinés à soutenir les humains dans les activités de la vie quotidienne grâce à leurs capacités de détection, de calcul et de communication. Leur aptitudes à observer le monde physique et à fournir des informations pour la prise de décision, seront partie intégrante de l'architecture de l'Internet du futur. Ces objets doivent s'intégrer dans un système plus global qu'est le monde digital et s'y adapter. L'IoT comprend une grande diversité de dispositifs intégrant capteurs et actionneurs. Le monde réel et numérique tend vers une plus grande osmose. Les composants logiciels et les objets physiques sont profondément corrélés, interagissant entre eux et avec les utilisateurs. Via les capteurs, l'IoT observe, mesure l'état du monde réel, ce qui est essentiel pour la prise de décision. Via les actionneurs, l'IoT agit sur le monde réel. Elle doit s'intégrer dans un système plus global qu'est l'écosystème digital.

Actuellement, nous ne pouvons pas intégrer un objet connecté dans une société d'objets (écosystème digital) de manière flexible, car il nécessite une configuration difficile, et la présence d'un groupe d'objets connectés les uns aux autres ne signifie pas qu'ils constituent une société qui travaille pour atteindre des objectifs et des intérêts individuels et collectifs communs et cela est dû à l'interopérabilité et les différentes structures et protocoles qui composent les objets.

Problématique et objectifs

L'interaction des éléments de cet écosystème digital entre eux et le sensing du monde physique par transformation des états d'une grandeur physique observée en une grandeur utilisable, pour le stockage et le traitement numériques, ce qui produit une énorme quantité des données (Big-data). Les ressources et les capacité de stockage et de traitement limitées des objets ne parviennent pas à gérer cette quantité des données, ce besoin en matière de stockage et de traitement de grande masse de données est pris en charge par l'infrastructure de calcul via le cloud.

Actuellement, la capacité limitée du réseau internet (la bande passante) pour transmettre cette quantité des informations représente un problème majeur, en attendant le lancement de la technologie de cinquième génération (5G) d'internet et sa couverture du monde, nous proposons d'autres solutions pour réduire la pression sur le réseau en laissant l'objet en partie traitant des données et en utilisant Edge et Fog computing . Nous proposons une approche de coopération entre objets au niveau du réseau local de l'écosystème, ce modèle est basé sur le paradigme agents coopératifs intégrés dans les couches existantes de l'objet.

Contributions

Pour répondre à la problématique décrite précédemment, notre thèse apporte les propositions suivant :

- Proposition d'une approche IoT basée sur les systèmes multi-agents pour l'intelligence ambiante, En effet, le système proposé est basé sur un système multi-agents (situé et mobile), en gardant l'avantage offert par ce paradigme et afin de maintenir le problème de communication entre les objets (IoT).
- Nous allons proposer une extension "Ambiance Intelligence Approach Using IoT and Multi-Agent System" en quatre couches. En particulier, nous avons ajouté une couche d'agent dans l'architecture. Nous avons intégré cette couche pour garantir les caractéristiques d'autonomie et d'intelligence de l'architecture IoT. De cette manière, les objets conservent les caractéristiques d'autonomie et d'intelligence de l'agent.
- Nous allons proposer un Robot IoT basé sur la deuxième contribution. Ce robot est doté

d'une caméra rotatable et des capteurs intégrés. Son rôle principal est de percevoir l'environnement à la demande des objets, autrement, il offre un service de sensing (sensing as service).

- Nous allons créer une API qui simule une maison intelligente, cette maison intelligente consiste à connecter les différents appareils et systèmes de la maison afin qu'ils puissent être contrôlés de n'importe où et provoquer l'interaction souhaitée entre eux. Ces appareils sont des objets connectés via un réseau qui a une fenêtre sur Internet.

Structure de la thèse

Cette thèse est organisée en cinq chapitres dont les thèmes sont donnés ci-dessous :

Le premier chapitre présente l'état de l'art sur l'internet des objets en commençant par une introduction, ensuite, nous donnons la définition de l'internet des objets ainsi que l'architecture de l'internet des objets suivie des différents types des objets connecté. De plus, il présente les domaines d'applications de l'internet des objets. Ce chapitre se termine par la présentation des systèmes embarqué.

Le deuxième chapitre présente l'état de l'art sur la technologie du Cloud Computing en commençant par un historique, ensuite, nous donnons la définition du Cloud Computing ainsi que les différents modèles de service et de déploiements. De plus, il présente les composants essentiels pour un contrat de Cloud avec les majeurs fournisseurs du Cloud. Nous citons à la fin de ce chapitre les avantages et les inconvénients de cette technologien.

Le troisième chapitre donne les approches de l'IoT dans le smart house et une synthèse bibliographique sur les travaux réalisés pour résoudre le problème explicite au début. Il présente aussi les méthodes utilisées pour résoudre les problèmes de communication entre les objets.

Dans le quatrième chapitre, nous discutons notre contribution qui consiste à résoudre le problème de l'interopérabilité des protocoles et hétérogénéité des architectures des objets, il commence par un exemple qui montre l'utilité de cette solution. Nous commençons par présenter l'architecture globale du système en détaillant son fonctionnement à l'aide d'un diagramme de séquence UML pour simplifier la compréhension de notre travail. Ensuite, nous expliquons l'architecture détaillée de chaque composant (sous-système) et son fonctionnement.

Le dernier chapitre concerne la mise en œuvre de notre approche, en commençant par la description des outils et l'environnement de développement. Ensuite, nous présentons quelques interfaces qui montrent les résultats obtenus d'après l'implémentation de notre modèle, nous terminons par une discussion sur les résultats obtenus.

Enfin, une conclusion clôture cette thèse. Elle synthétise les contributions globales et met en évidence les perspectives de cette recherche.

Chapitre 1

Internet des Objets

1.1 Introduction

Le terme Internet des objets ou en anglais (Internet of Things) a été utilisé pour la première fois en 1990. Mais l'idée réelle d'appareils connectés existe depuis plus longtemps, du moins depuis les années 70. A l'époque, l'idée était souvent appelée "Internet embarqué" ou "informatique omniprésente". Le terme actuel «Internet of Things» a été inventé par Kevin Ashton en 1999 lors de son travail au centre MIT (Massachusetts Institute of Technology), qui travaillait à l'optimisation de la chaîne logistique, souhaitait attirer l'attention de la haute direction sur une nouvelle technologie passionnante appelée RFID. Parce qu'Internet était la nouvelle tendance la plus en vogue en 1999 et qu'il avait un sens, il a appelé sa présentation «Internet of Things».

Même si Kevin a suscité l'intérêt de certaines entreprises. Le terme Internet des objets n'a pas retenu l'attention au cours des dix prochaines années. Le terme Internet des objets a encore émergé à l'été 2010, Selon des informations confidentielles, le service StreetView de Google n'avait pas seulement pris des photos à 360 degrés, il avait également stocké des tonnes de données sur les réseaux Wi-Fi de personnes. Les spécialistes se demandaient si c'était le début de la nouvelle stratégie de Google, non seulement pour l'indexation d'Internet, mais également pour l'indexée le monde physique.

La même année, le gouvernement chinois a annoncé qu'il ferait de l'Internet des objets une priorité stratégique de son plan quinquennal.

En 2011, Gartner, la société d'études de marché qui a inventé le fameux « hype-cycle for

emerging technologies » a ajouté un nouveau phénomène émergent à sa liste : «l'Internet des objets». L'année suivante, le thème de la plus grande conférence Internet européenne, Le Web, était l'Internet des objets. Parallèlement, des magazines populaires axés sur la technologie, tels que Forbes, Fast Company et Wired, ont commencé à utiliser l'IoT comme vocabulaire pour décrire le phénomène.

En octobre 2013, International Data Corporation (IDC) a publié un rapport indiquant que l'Internet des objets représenterait un marché de 8 900 milliards de dollars en 2020.

Le terme Internet of Things a atteint la notoriété sur le marché de masse quand, en janvier 2014, Google a annoncé l'achat de Nest pour 3,2 milliards de dollars. Au même moment, le Consumer Electronics Show (CES) de Las Vegas s'est tenu sous le thème de l'IoT.

Actuellement, il y a plus de 26 milliards d'appareils IoT connectés, devrait dépasser 75 milliards en 2025. La valeur du marché mondial de l'IoT en 2019 dépasse 1,7 billion de dollars et le matériel représente 35% de cette valeur.

1.2 Internet des objets

L'objet connecté est un objet électronique qui peut transmettre des informations en temps réel via une liaison sans fil à une autre dispositif connecté. Ces informations peuvent être de plusieurs types [58] .

Les objets connectés sont des équipements du quotidien qui dotés de capteurs et de dispositifs d'échange de données, permettent de disposer à distance d'informations sur une partie de notre environnement, ou échangent de l'information pour produire un service [105, 7].

Un objet connecté est un matériel, disposant de composants électroniques lui permettant de communiquer des informations vers un autre objet, un serveur informatique, un ordinateur, une tablette ou un smartphone, en utilisant une liaison sans fil vers un réseau dédié (le plus souvent Internet) [19].

Un objet connecté tend aussi à avoir une capacité de traitement qui lui est propre sur les données qu'il capte ou mesure. Ce traitement local permettant d'alléger la quantité des informations transmises pour s'adapter à la capacité du lien de communication ou des systèmes de traitement distants [123].

La capacité de traitement des objets connectés tend aussi à intégrer des logiques dites « intelligentes » c'est-à-dire capables d'agir sur le comportement [75] de l'objet lui-même ou de moduler la transmission d'informations, en fonction des informations captées ou des mesures effectuées [95].

Un objet peut être une entité physique ou virtuelle ayant des identités et des personnalités virtuelles, opérant dans des espaces intelligents et utilisant des interfaces intelligentes pour se connecter et communiquer au sein de contextes d'usages variés [51].

Lorsque qu'un ensemble d'objets connectés communiquent et interagissent entre eux ou avec des serveurs de traitement via le réseau Internet, on parle alors d'Internet des Objets (IdO) ou Internet of Things (IoT).

1.3 Architecture de l'internet des objets

Dans l'IoT, chaque couche est définie par ses fonctions et les périphériques utilisés dans cette couche. L'architecture de l'IoT est généralement divisée en trois couches, la couche perception, la couche réseau et la couche application [126].

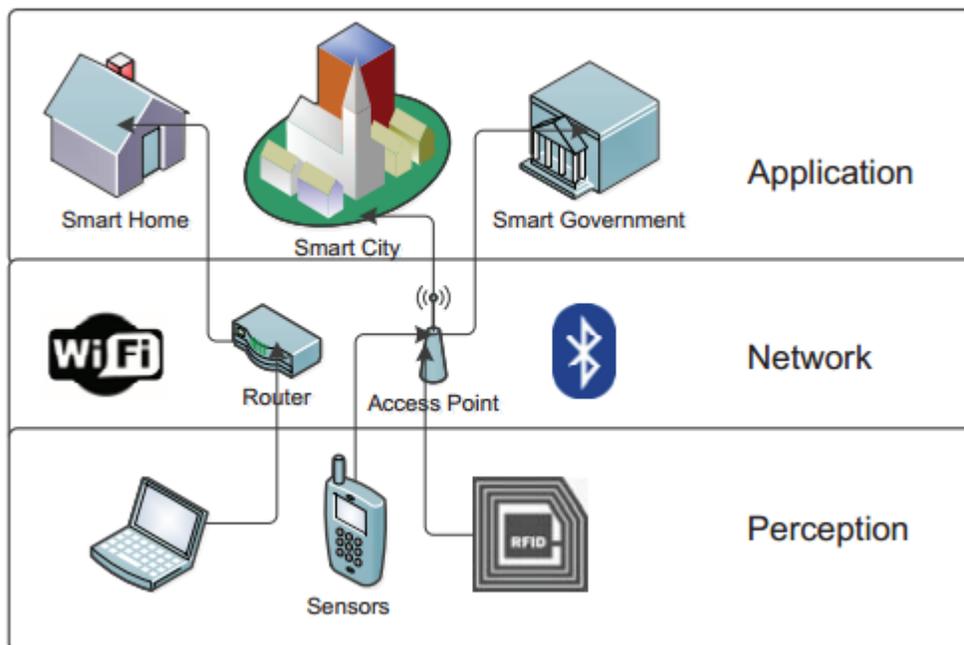


FIGURE 1.1 – Architecture à trois couches de l'IoT [70]

1.3.1 Couche de perception

La couche de perception est également connue sous le nom de couche «Capteurs» dans l’IoT. Le but de cette couche est d’acquérir les données de l’environnement à l’aide de capteurs et d’actionneurs. Cette couche détecte, collecte et traite les informations, puis les transmet à la couche réseau. Cette couche effectue également la collaboration des nœuds IoT dans les réseaux locaux et à courte portée [13].

1.3.2 Couche réseau

La couche réseau de l’IoT sert à la fonction de routage et de transmission des données vers différents hubs et appareils IoT sur Internet. À cette couche, les plates-formes de cloud computing, les passerelles Internet, les dispositifs de commutation et de routage ...etc. fonctionnent en utilisant certaines technologies très récentes telles que WiFi, LTE, Bluetooth, 3G, Zigbee, etc. Les passerelles réseau servent de médiateur entre différents IoT nœuds en agrégeant, filtrant et transmettant des données vers et depuis différents capteurs [64].

1.3.3 Couche d’application

La couche application est considérée comme une couche supérieure de l’architecture IoT conventionnelle. Cette couche fournit des services personnalisés en fonction des besoins des utilisateurs [110]. La responsabilité principale de cette couche est de relier l’écart majeur entre les utilisateurs et les applications. Cette couche IoT combine l’industrie pour atteindre les solutions de type d’applications intelligentes de haut niveau telles que la surveillance des catastrophes, la surveillance de la santé, la transposition, la fortune, l’environnement médical et écologique et la gestion globale gérée pertinente pour toutes les applications de type intelligent.

1.4 Objet connecté

Un «objet» dans «l’Internet des objets» est une unité de traitement capable de se connecter à Internet et d’échanger des données avec le cloud. Les appareils sont souvent appelés «appareils

intelligents» ou «appareils connectés». Ils communiquent deux types de données : la télémétrie et l'état [133].

1.4.1 Types d'informations

Chaque appareil peut fournir ou consommer différents types d'informations. Chaque forme d'information peut être gérée au mieux par un système backend différent, et chaque système doit être spécialisé autour du débit de données, du volume et de l'API préférée.

1. **Métadonnées de l'appareil** : les métadonnées contiennent des informations sur un périphérique. La plupart des métadonnées changent rarement, voire jamais. Exemples de champs de métadonnées [9] :
 - Identifiant (ID)
 - Classe ou type
 - Modèle
 - Révision
 - Date de fabrication
 - Numéro de série du matériel
2. **Télémétrie** : les données collectées par l'appareil sont appelées télémétrie. Ce sont les données des yeux et des oreilles que les appareils IoT fournissent aux applications. La télémétrie est une donnée en lecture seule sur l'environnement, généralement collectée via des capteurs [101].
3. **Informations sur l'état** : les informations d'état décrivent l'état actuel de l'appareil, pas celui de l'environnement. Ces informations peuvent être lues / écrites, mis à jour, mais généralement pas fréquemment [60].
4. **Commandes de périphérique** : les commandes sont des actions effectuées par un appareil. Les commandes peuvent être valides pour une période de temps limitée, elles doivent donc inclure une durée de vie ou time to live (TTL) en anglais ou une autre valeur d'expiration [60].
5. **Informations opérationnelles** : les informations opérationnelles sont les données les plus pertinentes pour le fonctionnement de l'appareil par opposition à l'application mé-

tier. Cela peut inclure des éléments tels que la température de fonctionnement du processeur et état de la batterie. Ce type de données peut ne pas avoir de valeur analytique à long terme, mais il a une valeur à court terme pour aider à maintenir l'état de fonctionnement, comme répondre aux pannes et corriger la dégradation des performances du logiciel après les mises à jour. Les informations opérationnelles peuvent être transmises sous forme de données de télémétrie ou d'état [111].

1.4.2 Types des objets connectés

Il y a plusieurs critères de classification des types des objets, Il existe deux grandes modèles des objets connectés qui sont :

1.4.2.1 Les objets portés

De manière générale, les termes «technologie wearable», «dispositifs wearable» et «vêtements intelligents» se réfèrent tous à des technologies électroniques ou à des micro-ordinateurs qui sont incorporés dans des vêtements et des accessoires, qui peuvent être portés sur le corps. Suite au fait que les performances de calcul augmentent constamment, les objets portés contemporains deviennent capables d'effectuer des tâches informatiques similaires à celles des smartphone ou même des ordinateurs portables.

1.4.2.2 Les objets indépendants

Tous les objets non portables avec des capteurs déployés à plusieurs endroits, comme les stations météo, les voitures intelligentes, les drones...etc

Il existe une autre classification pour déterminer les types des objets connecté, ils peuvent être classés en grande partie soit passifs soit actifs en fonction de la source d'énergie détectée.

1.4.2.3 Les objets passifs

Utiliser généralement c'est un tag (carte puc, tag RFID, NFC, code barre) Les technologies de capteurs passifs collectent des données cibles grâce à la détection des vibrations, de la lumière,

du rayonnement, de la température ou d'autres phénomènes se produisant dans l'environnement.

1.4.2.4 Les objets actifs

Un objet a une capacité de stockage et de traitement qui comprennent des émetteurs qui envoient un signal, une longueur d'onde lumineuse ou des électrons à rebondir sur la cible, avec des données recueillies par le capteur lors de leur réflexion.

1.4.3 Capteurs et actionneurs

Les capteurs sont les yeux et les oreilles de l'objet IoT pour voir son environnement, et les actionneurs sont les jambes et les mains qui remplissent ses fonctions.

1.4.3.1 Capteurs

Un capteur est un appareil qui convertit un paramètre physique en une sortie électrique. Un capteur est un type de transducteur. Les capteurs peuvent être divisés en capteurs analogiques et capteurs numériques. Les capteurs analogiques fournissent une sortie sous forme de tensions et de courants. Les microcontrôleurs auront besoin d'un ADC (analogue-to-digital converter) [49] pour lire les données des capteurs analogiques. De nombreux capteurs plus récents sont des capteurs numériques [55], c'est-à-dire qu'ils fournissent une sortie au format numérique, en utilisant des protocoles tels qu'I2C (Inter-Integrated Circuit), SPI (Serial Peripheral Interface) [72] et UART (universal asynchronous receiver/transmitter) [29], etc. Les capteurs numériques sont excellents. Pour les systèmes embarqués, car ils contournent le besoin d'ADC et rendent le circuit beaucoup plus simple. Les exemples incluent les capteurs de température, les capteurs d'humidité, les capteurs de pression, les capteurs de fumée, les capteurs de son et de lumière, ..etc.

Il existe de nombreux capteurs disponibles pour l'IoT et un certain nombre de façons de les catégoriser. Les catégories décrites ci-dessous ne sont qu'un petit échantillon des façons dont les capteurs peuvent être regroupés. Les capteurs peuvent être divisés par :

- 1. leurs besoins en alimentation externe** [108, 77] :

Type	Définition	Exemple
Passive	Ne nécessite pas d'alimentation externe pour fonctionner. Ils répondent aux entrées de leur environnement.	Un capteur de température qui modifie la résistance en réponse aux changements de température
Active	Nécessite une alimentation externe pour fonctionner.	une camera

2. Type de signal produit par le capteur [34, 88] :

Type	Définition	Exemple
Analogique	Émet un signal analogique continu	Accéléromètres, capteurs de température
Numérique	La sortie est convertie en valeurs discrètes (1s et 0s numériques) avant de transmettre à un appareil	Capteur de pression numérique, capteur de température numérique

3. Type d'appareil de mesure [50, 89, 25] :

Type	Définition	Exemple
Chimique	Répond aux changements chimiques dans son environnement	Capteur de gaz
mécanique	Répond aux changements physiques de son environnement	Micro-interrupteur
électrique	Répond aux changements électriques dans son environnement	Capteur optique

Le choix des capteurs pour un projet nécessite une compréhension claire de ce qu'on veut mesurer et de la précision requise.

Lors de la sélection d'un capteur IoT, plusieurs éléments doivent être pris en compte. En règle générale, l'objectif d'un capteur et d'un dispositif IoT est une longue durée de vie avec peu d'interaction humaine. Il faut placer les capteurs et appareils IoT dans l'environnement souhaité et pour les faire fonctionner pendant une période de temps prolongée. Ils peuvent se trouver dans un endroit éloigné ou être enfouis profondément dans un système, inaccessibles aux

humains. Le remplacement d'un capteur et d'un appareil dans cette situation peut être extrêmement coûteux, dangereux, voire impossible; toutes les raisons de bien réfléchir aux décisions concernant le capteur et appareil.

La décision est basée sur de nombreux facteurs. pour concerver le système, il faut soigneusement considérer l'importance de chaque facteur et sa priorité pour la conception globale.

La liste suivante de considérations peut être considérée comme un point de départ pour toute discussion sur les capteurs IoT :

- **Durabilité** : la durabilité doit être prise en compte en ce qui concerne l'environnement du capteur. il faut assurer que l'appareil est aussi durable que nécessaire pour fonctionner pendant une période de temps raisonnable, sans encourir de coûts inutiles [36].

Par exemple, un capteur de température résistant à l'eau peut être acceptable pour une station météorologique à distance, mais il ne conviendrait pas du tout pour surveiller la température de l'eau dans une piscine car il n'est pas étanche.

- **Précision** : afin d'avoir suffisamment de précision pour surveiller correctement un environnement, mais ne pas payer plus que ce dont on a besoin [2].

Par exemple, concevez un système pour réguler la température dans une unité de stockage domestique à distance, il faut probablement être prêt à accepter un capteur qui pourrait être précis avec +/- 2 degrés. Cette précision serait totalement inacceptable dans un système de dispositifs médicaux. Un capteur de température de dispositif médical devrait être précis à +/- 0,2 degré!

- **Polyvalence** : les capteurs doivent pouvoir fonctionner dans des variations raisonnables de l'environnement. Parce que la plupart des conceptions de réseaux IoT ont de nombreux capteurs, dans une variété d'environnements, il est important d'avoir des capteurs qui peuvent fonctionner avec précision dans toutes les variations de l'environnement [100].

Par exemple, pour construire des stations météorologiques éloignées pour les zones sauvages, On doit utiliser des capteurs capables de gérer les températures extrêmes d'été et d'hiver. Il ne serait pas pratique d'avoir des capteurs qui ne fonctionnent avec précision qu'à température ambiante.

- **Consommation d'énergie** : selon la situation, les besoins peuvent concerner un péri-

phérique à faible consommation, voire à très faible consommation. Il faut décider si des fonctions d'économie d'énergie (comme le mode veille ou le réveil rapide) sont nécessaires [71].

Par exemple, un capteur ou un appareil alimenté par des batteries solaires peut avoir besoin de passer une grande partie de sa vie en mode veille pour prolonger la durée de vie de la batterie pendant les périodes de faible éclairage. Il peut également nécessiter des temps de réveil rapides pour capturer avec précision les données.

- **Considérations environnementales spéciales** : le choix du capteur peut même affecter la conception finale du système [57].

Par exemple, lors de la conception d'un système de surveillance de la qualité de l'eau, un capteur qui peut être placé dans la tuyauterie principale d'alimentation en eau est beaucoup plus rentable et précis qu'un capteur qui nécessite de détourner des échantillons d'eau.

- **Coût** : les réseaux IoT impliquent généralement des centaines voire des milliers de capteurs et d'appareils. Tous les aspects de la conception des capteurs doivent être examinés du point de vue des coûts. Ces coûts impliquent plus que le prix du capteur. Il faut tenir compte du coût de placement, d'entretien, de fiabilité,..etc [78].

1.4.3.2 Actionneur

Un actionneur est un dispositif qui convertit un signal électrique en sortie physique [120], c'est-à-dire un mouvement. Un actionneur peut être contrôlé par la tension ou le courant électrique, la pression pneumatique ou hydraulique [10], ou même la puissance humaine. Dans les systèmes embarqués, les actionneurs sont principalement contrôlés par l'électricité. Lorsque le signal de commande est reçu, l'actionneur convertit l'énergie électrique en mouvement mécanique. Les actionneurs peuvent créer un mouvement linéaire, un mouvement rotatif ou un mouvement oscillatoire. Des exemples d'actionneurs comprennent les moteurs électriques, les actionneurs piézoélectriques, les actionneurs pneumatiques, les moteurs pas à pas et les actionneurs de serrure de porte, etc [46].

1.4.4 Caractéristiques fondamentales de l'internet des objets

Les caractéristiques fondamentales de l'IoT sont les suivantes :

1. **Interconnexion** : il facilite l'interconnexion d'appareil et d'appareil à appareil [38].
2. **Détection intelligent** : les appareils connectés à l'IoT auront des capacités de détection intelligente. Par exemple, l'utilisation de détecteurs de mouvement pour allumer ou éteindre les lumières. La technologie de détection aide à créer des expériences qui reflètent une véritable conscience du monde physique, des personnes et des objets [26].
3. **Intelligence** : les appareils connectés à l'IoT peuvent être dotés d'intelligence. Par exemple, les Nest Learning Thermostats sont compatibles Wi-Fi, pilotés par capteur et dotés de capacités d'auto-apprentissage. Le Misfit Shine est un tracker de fitness avec des capacités de surveillance du sommeil. Le Misfit Shine peut distribuer des tâches de calcul entre un smartphone et le cloud [128].
4. **Économiser l'énergie** : les appareils IoT comme Motion Sensor Light ont un détecteur de mouvement intégré qui peut allumer la lumière lorsqu'elle détecte un mouvement. Il peut économiser beaucoup d'énergie électrique du gaspillage et stimuler la récupération d'énergie et l'utilisation efficace de l'énergie [71].
5. **Expressive** : les appareils connectés à l'IoT ont une capacité unique de dire l'état actuel aux autres appareils connectés dans les environs. Ils donnent un meilleur flux de communication entre l'homme et les machines [87].
6. **Sécurisé** : les appareils connectés à l'IoT peuvent contribuer à garantir la sécurité de la vie individuelle. Par exemple, un pneu de voiture en mouvement peut indiquer son état actuel au propriétaire de la voiture ayant des tableaux de bord de voiture intelligents, cela aidera à prévenir les accidents dus à l'éclatement des pneus de voiture en raison d'une surchauffe,..etc [83].

1.4.5 Axes de l'internet des objets

L'Internet des objets représente la troisième vague révolutionnaire de la technologie informatique après l'ordinateur personnel et Internet [19]. Certains chercheurs considèrent l'IoT

comme la prochaine phase de l'évolution d'Internet pour atteindre l'objectif de l'apprentissage machine à machine (M2M) [51]. D'autres chercheurs considèrent l'IoT comme une extension du réseau Internet [43]. Cependant, il existe une différence majeure entre l'IoT et Internet. Les sections suivantes mettent en évidence l'IoT et sa différence avec Internet.

L'IoT est différent d'Internet sur le plan de la communication. La communication Internet peut être établie à tout moment et en tout lieu. Cependant, la communication dans l'IoT a la dimension Any THING comme dimension supplémentaire, comme illustré la figure 1.2 [98]. L'objectif principal de l'IoT est de fournir une connectivité pour tout le monde à tout moment et en tout lieu.

Le concept IoT fonctionne pour unir et relier le monde physique et le monde virtuel [19], et le lien échange des données entre des appareils réels et des cyber-applications dans une connexion sécurisée [1]. La forme physique de l'IoT se compose de périphériques externes, qui sont partagés au sein de l'infrastructure Internet [81].

Cependant, certains appareils sont directement liés à l'IoT comme la RFID, les capteurs et les actionneurs qui comblent le fossé entre le monde physique et le monde de l'information [35]. L'IoT relie différents types d'objets entre eux et leur permet de communiquer intelligemment [15]. Une telle configuration atteint l'objectif d'identité, de suivi et de gestion intelligents [19]

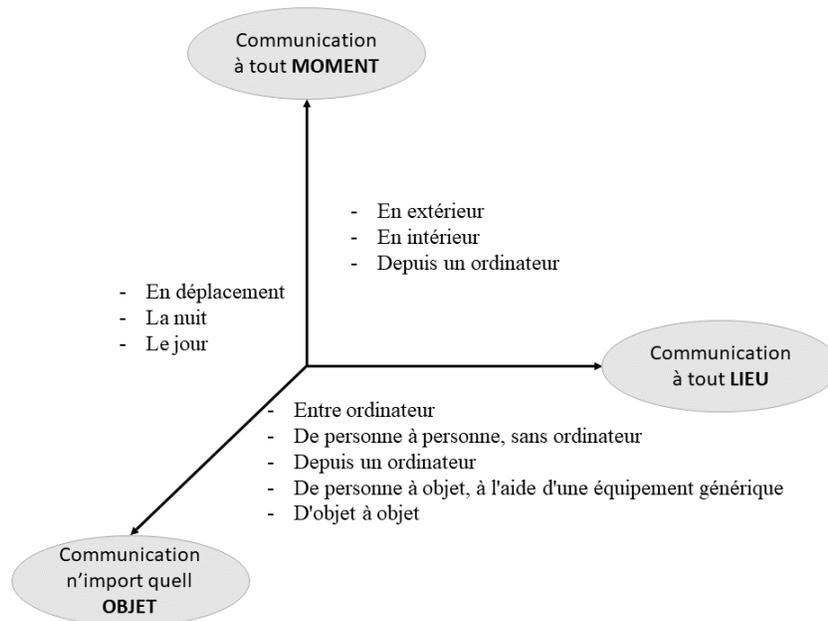


FIGURE 1.2 – Axes de l'internet des objets [70]

1.5 Domaine d'application de l'internet des objets

IoT est utilisé dans plusieurs domaines, nous citons comme exemples :

1.5.1 Maison intelligente

Les maisons intelligentes [39] seront probablement les applications IoT les plus populaires. La maison intelligente, ou domotique, est une extension de l'automatisation du bâtiment, avec laquelle nous pouvons surveiller et contrôler le chauffage, la ventilation et la climatisation (CVC), l'éclairage [121], les appareils électroménagers, les systèmes de sécurité à bande. En connectant tous les appareils électroménagers, nous pouvons automatiser de nombreuses routines quotidiennes, comme allumer et éteindre automatiquement les lumières et le chauffage, démarrer ou arrêter la cuisson et le lavage, etc. Avec le réseau intelligent et les compteurs intelligents, nous pouvons réduire les consommations d'énergie et les factures de services publics, et avec les systèmes de sécurité, nous pouvons rendre la maison plus sûre en détectant automatiquement et en dissuadant, espérons-le, les intrusions en utilisant divers capteurs infrarouges, de mouvement, sonores, de vibration ainsi que des systèmes d'alarme [61].

Une maison intelligente peut également rendre les personnes âgées et les personnes handicapées plus confortables et plus sûres à la maison. Avec l'IoT, nous pouvons collecter et analyser les données des personnes âgées et handicapées pour diagnostiquer les maladies, prédire les risques potentiels, identifier ou prévenir les accidents tels que les chutes, ouvrir ou verrouiller la porte (ou les fenêtres) à distance, et laisser les membres de la famille les surveiller à distance [61]. Avec l'IoT, il est également possible de rapprocher les personnes âgées et handicapées du monde extérieur et de réduire leur sentiment de solitude.

Le marché de la maison intelligente devrait avoir une valeur marchande de plus de 53 milliards de dollars en 2022 [107, 127].

1.5.2 La santé

L'IoT permet la surveillance à distance de la santé et les systèmes de notification d'urgence. Une approche très populaire consiste à utiliser des techniques portables. Ces appareils portables peuvent collecter une gamme de données de santé, telles que la fréquence cardiaque, la

température corporelle et la pression artérielle, qui peuvent ensuite être transmises sans fil à un site distant pour le stockage et une analyse plus approfondie. Cela permet également à la télésanté/télé médecine, c'est-à-dire de diagnostiquer ou de traiter les patients à distance [23].

1.5.3 Transport

L'IoT peut considérablement améliorer les systèmes de transport. Avec toutes les voitures connectées, il est beaucoup plus facile de planifier votre voyage, d'éviter les embouteillages, de trouver une place de parking et de réduire les accidents de la circulation [62]. Les voitures sans conducteur auront sans aucun doute le plus grand impact. De nombreuses entreprises, telles que Tesla, Google, Uber, Volvo, Volkswagen, Audi et General Motors, les développent et les promeuvent activement. Les voitures sans conducteur peuvent rendre notre voyage plus agréable et peut-être beaucoup plus sûr. Obtenir un permis de conduire pourrait bientôt être chose du passé [91]

L'IoT peut également bénéficier aux transports publics. En connectant tous les panneaux d'information et les panneaux publicitaires dans les gares et les aéroports [12], il aide les passagers à obtenir des mises à jour régulières, et en cas d'accident, à détecter rapidement les problèmes et à réduire les coûts de maintenance. En améliorant la visibilité de bout en bout, la gestion des entrepôts et la gestion de la flotte, l'IoT bénéficiera également à l'industrie de la logistique [62].

1.5.4 Énergie

En intégrant des capteurs et des actionneurs, il est susceptible de réduire la consommation d'énergie de tous les appareils consommateurs d'énergie. L'IoT modernisera également l'infrastructure de l'industrie électrique, afin d'améliorer l'efficacité et la productivité [84].

1.5.5 Industrie

L'application de l'IoT dans l'industrie est souvent appelée Industrie 4.0 ou quatrième révolution industrielle (figure 1.3) [30]. La première révolution industrielle a eu lieu au XVIIIe siècle lorsque la machine à vapeur a mobilisé la production industrielle. La deuxième révolution in-

dustrielle a eu lieu au début du XIXe siècle, lorsque l'énergie électrique alimentait la production de masse. La troisième révolution industrielle, ou révolution numérique, a eu lieu à la fin du XIXe siècle lorsque l'électronique et l'informatique ont encore automatisé la production. L'industrie 4.0 s'appuie sur des systèmes cybers - physiques qui intègrent étroitement les machines, les logiciels, les capteurs, Internet et les utilisateurs. Il créera des usines intelligentes, dans lesquelles les machines pourront utiliser l'auto-optimisation, l'auto configuration et même l'intelligence artificielle pour effectuer des tâches complexes afin de fournir des économies de coûts largement supérieures et des biens ou services de meilleure qualité [119].

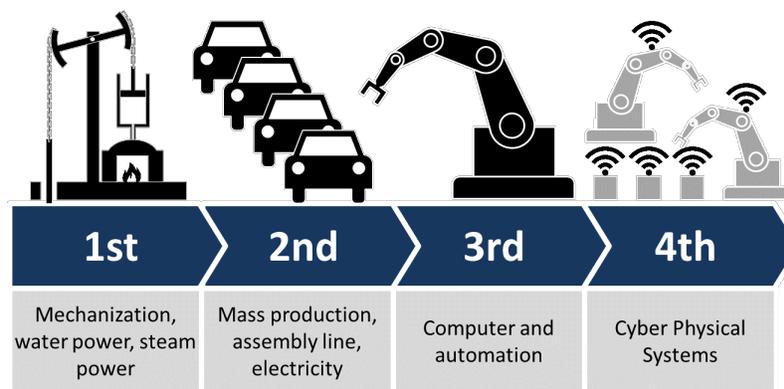


FIGURE 1.3 – Révolutions industrielles [119]

1.5.6 Environnement

En déployant des capteurs environnementaux, nous pouvons mesurer et surveiller la qualité de l'air, la qualité de l'eau, les conditions du sol, le rayonnement et les produits chimiques dangereux plus efficacement. Nous pouvons également mieux prévoir les tremblements de terre et les tsunamis et détecter plus rapidement les incendies de forêt, les avalanches de neige et les glissements de terrain. Tout cela nous aidera à mieux protéger notre environnement [27]. En marquant les animaux sauvages, en particulier les espèces en voie de disparition, nous pouvons étudier et mieux comprendre le comportement des animaux [76, 86], et ainsi offrir une meilleure protection et des habitats plus sûrs. L'IoT permettra également une agriculture intelligente, qui offrira une visibilité 24/7 sur la santé des sols et des cultures, et aidera les agriculteurs à optimiser l'utilisation des engrais et des produits phytopharmaceutiques. Cela aura à nouveau inévitablement un impact positif sur l'environnement [79].

1.6 Technologies de communication

Outre les techniques de communications conventionnelles telles qu’Ethernet, Wi-Fi et Bluetooth, de nombreuses autres technologies peuvent être utilisées pour les communications sur l’Internet des objets.

1.6.1 RFID et NFC (Near-Field Communication)

L’identification par radiofréquence, en anglais Radio-frequency identification (RFID) est une technologie qui peut identifier et suivre de manière unique les étiquettes attachées aux objets à l’aide d’ondes électromagnétiques radiofréquence [3]. Un système RFID comprend généralement une étiquette, un lecteur et une antenne. Le lecteur envoie un signal d’interrogation à l’étiquette via l’antenne, et l’étiquette répond avec ses informations uniques. Les étiquettes RFID peuvent être actives ou passives. Les étiquettes RFID actives ont leur source d’alimentation et peuvent donc être lues sur une longue portée (jusqu’à 100 mètres) [8]. Les étiquettes RFID passives n’ont pas leur source d’alimentation. Ils sont alimentés par l’énergie électromagnétique transmise par le lecteur RFID. Par conséquent, ils ne peuvent être lus que sur une courte distance (<25m) [52]. La RFID fonctionne principalement dans les gammes de fréquences suivantes, comme indiqué dans Tableau 1.1.

La communication en champ proche, en anglais Near-field communication (NFC) est une technologie de communication qui fonctionne à la même fréquence (13,56 MHz) que la RFID HF [4]. Différent de la RFID, le NFC est basé sur une communication d’égal à égal, ce qui signifie qu’un appareil NFC peut être soit un lecteur soit une étiquette.

Cette capacité unique a fait du NFC un choix populaire pour le paiement sans contact, les cartes d’identité et les cartes de voyage, etc. Les appareils NFC communiquent généralement à moins de 4 cm (2 pouces) les uns des autres. NFC est désormais disponible sur la plupart des nouveaux smartphones. Les smartphones NFC peuvent être utilisés pour le paiement sans contact, ainsi que pour transmettre les informations (informations de contact ou photographies) d’un smartphone à l’autre en tapant les deux appareils ensemble [103].

TABLE 1.1 – Bandes de fréquences RFID

Band	Range	Data Speed	Tags
Low frequency (LF) : 125–134.2 kHz	10 m	low	passive
High frequency (HF) : 13.56 MHz	10 cm–1 m	low to moderate	passive
Ultra high frequency (UHF) : 433 MHz	1–100 m	moderate	passive or active
Ultra high frequency (UHF) : 856 MHz–960 MHz	1–12 m	moderate to high	passive or active
Microwave : 2.45–5.8 GHz	1–2 m	high	active
Microwave : 3.1–10 GHz	<200 m	high	active

1.6.2 Bluetooth Low Energy (BLE)

Bluetooth low energy (BLE) est un membre plus récent de la famille Bluetooth, basé sur les normes Bluetooth 4.0. Semblable au Bluetooth classique, le BLE fonctionne également dans la bande ISM 2,4 GHz mais utilise un système de modulation plus simple. Le BLE reste constamment en mode veille, sauf lorsqu’une connexion est établie et consomme donc beaucoup moins d’énergie [24]. BLE est arrivé sur le marché en 2011 et est commercialisé sous le nom de Bluetooth Smart. Le BLE est conçu pour fournir une consommation d’énergie et des coûts très réduits tout en conservant une plage de communication similaire. Le BLE fonctionne généralement à une distance d’environ 100 m, avec un débit de données d’environ 1 Mbits/s [112]. Les applications typiques du BLE sont :

- Moniteurs de fréquence cardiaque
- Tensiomètres
- Moniteurs de glycémie
- Appareils Fitbit
- Capteurs de surveillance industriels
- Promotions ciblées basées sur la géographie (iBeacon)
- Détection de proximité

1.6.3 LiFi

Light Fidelity (LiFi) est une nouvelle technologie de communication haute vitesse sans fil, bidirectionnelle, basée sur la lumière visible à modulation rapide. Il s'agit d'un type de système de communication à lumière visible (Visible Light Communications : VLC). Semblable au WiFi, le LiFi transmet des données à l'aide d'ondes électromagnétiques [44]. Mais au lieu d'utiliser des ondes radio (MHz - GHz), il utilise la lumière visible (THz). LiFi utilise des ampoules LED domestiques (light-emitting diodes) comme émetteurs. En faisant varier le courant électrique fourni à une ampoule LED à des vitesses extrêmement élevées, les données peuvent être encodées à mesure que la luminosité change rapidement, qui peuvent ensuite être captées par un photodétecteur (photodiode). Ces changements rapides sont trop rapides pour être remarqués aux yeux des humains ; par conséquent, le LiFi n'affecte pas la fonction principale des lumières LED - l'éclairage [45].

Le LiFi a un énorme avantage en termes d'infrastructure, car les ampoules LED sont de plus en plus utilisées dans les bâtiments, les rues et les véhicules. Il peut fonctionner à une vitesse impressionnante pouvant atteindre 224 gigabits par seconde et il est insensible aux interférences électromagnétiques. Le LiFi ne peut pas pénétrer les murs, ce qui signifie qu'il ne peut fonctionner qu'à courte portée, mais en même temps, cela le rend moins susceptible d'être piraté. Il existe déjà des produits sur le marché qui peuvent fournir à la fois lumière et connectivité [17].

1.6.4 6LoWPAN

6LoWPAN signifie IPv6 (Internet Protocol Version 6) sur les réseaux personnels sans fil à faible puissance (Personal Area Networks : WPAN) [23]. Il s'agit essentiellement d'un réseau maillé sans fil à faible puissance et faible débit basé sur les normes IEEE 802.15.4, utilisant IPv6 comme protocole de communication. Comparé à d'autres réseaux locaux, 6LoWPAN a un avantage distinct, c'est-à-dire qu'il est basé sur des normes ouvertes TCP/IP, y compris TCP, UDP, HTTP, CoAP, MQTT et WebSockets etc [94]. Il a des nœuds adressables IPv6 de bout en bout et peut être facilement connecté à Internet directement. Il est également auto-réparateur en raison du routage du maillage. 6LoWPAN a été utilisé dans les réseaux de capteurs sans fil, les lumières et les compteurs [41].

1.6.5 ZigBee

ZigBee est une technologie de communication de haut niveau pour les réseaux personnels à faible puissance et à faible débit de données, tels que les réseaux de capteurs [37], les automatismes domestiques et les dispositifs médicaux. ZigBee est basé sur la norme IEEE 802.15.4. Il a une distance de transmission de 10 à 100 mètres et doit être en ligne de mire. Il opère dans les bandes radio industrielles, scientifiques et médicales (ISM), à savoir 868 MHz en Europe, 915 MHz aux États-Unis et en Australie, 784 MHz en Chine et 2,4 GHz dans le reste du monde [73]. ZigBee a un débit de données allant de 20 kbit / s (bande 868 MHz) à 250 kbit / s (bande 2,4 GHz). Les réseaux ZigBee sont normalement moins chers que les autres réseaux sans fil tels que Bluetooth ou WiFi. ZigBee a été utilisé pour les interrupteurs d'éclairage sans fil, les compteurs électriques (réseau intelligent, réponse à la demande, etc.), la surveillance des équipements industriels, etc [48].

La différence entre ZigBee et 6LoWPAN, ZigBee a été plus longtemps, et a donc été adopté plus largement que 6LoWPAN [69]. ZigBee est sans aucun doute la norme de réseautique maillée sans fil à faible coût et à faible consommation d'énergie la plus populaire actuellement. Cependant, 6LoWPAN rattrape son retard et devient plus attractif, car il est basé sur IP, en particulier avec le support IPv6. Cela facilite l'intégration avec le reste d'Internet. À ce jour, de nombreuses sociétés de semi-conducteurs (par exemple, Texas Instruments, Freescale et Atmel, etc.) fabriquent des puces 802.15.4 qui prennent en charge à la fois ZigBee et 6LoWPAN [33].

1.6.6 Z-Wave

Z-Wave est une technologie de communication sans fil qui est principalement utilisée pour la domotique, comme le contrôle et l'automatisation des lumières et des appareils. Il peut être utilisé comme système de sécurité ou pour surveiller et contrôler votre propriété à distance [115]. Z-Wave fonctionne dans la bande industrielle, scientifique et médicale (ISM) non autorisée, à savoir 868,42 MHz en Europe, 908,42 MHz aux États-Unis et au Canada, et d'autres fréquences dans d'autres régions. Z-Wave est conçu pour fournir une transmission fiable à faible latence à une portée d'environ 100 mètres, avec des débits de données jusqu'à 100 kbit/s. Un réseau Z-Wave comprend normalement un contrôleur principal et une collection de périphé-

riques (jusqu'à 232) [114].

1.6.7 LoRa

LoRa est une technologie de communication à longue portée conçue pour les communications à faible puissance et à longue distance des dispositifs IoT alimentés par batterie, c'est-à-dire un réseau étendu à faible puissance (LPWAN). Il prend en charge les communications bidirectionnelles sécurisées des réseaux avec des millions et des millions d'appareils [116, 117].

Le tableau 1.2 présente une comparaison rapide des différentes technologies de communication sans fil. Le LiFi et le WiFi offrent potentiellement les débits de données les plus élevés, tandis que le cellulaire et le LoRa offrent les plus longues distances.

TABLE 1.2 – Comparaison de différentes technologies

	Standard	Frequency	Range	Data Rate
LiFi	Similar to 802.11	400–800 THz	<10 m	<224 Gbps
WiFi	802.11a/b/g/n/ac	2.4 GHz and 5 GHz	~50 m	<1 Gbps
Cellular	GSM/GPRS/EDGE (2G), UMTS/HSPA (3G), LTE (4G), 5G	900, 1800, 1900, and 2100MHz 2.3, 2.6, 5.25, 26.4, and 58.68 GHz	<200 km	<500 kps (2G), <2 Mbps (3G), <10 Mbps (4G) <100 Mbps (5G)
Bluetooth	Bluetooth 4.2	2.4 GHz	50–150 m	1 Mbps
RFID/NFC	ISO/IEC 18000-3	13.56 MHz	10 cm	100–420 kbps
6LowPAN	RFC6282	2.4 GHz and ~1 GHz	<20 m	20–250 kbps
ZigBee	ZigBee 3.0 based on IEEE 802.15.4	2.4 GHz	10–100 m	250 kbps
Z-Wave	Z-Wave Alliance ZAD12837 / ITU-T G.9959	868.42 MHz and 908.42 MHz	<100 m	<100 kbps
LoRa	LoRaWAN	868 MHz and 915 MHz	<15 km	0.3–50 kbps

1.7 Protocoles de l'internet des objets

Les protocoles, ou les protocoles de communication, sont un ensemble de règles qui permettent aux appareils de communiquer entre eux. Les protocoles définissent la syntaxe, la sémantique et la synchronisation de la communication. Une analogie étroite avec les protocoles est celle des langues humaines. Il existe de nombreux protocoles de communication disponibles pour les applications IoT. Les protocoles suivants sont couramment utilisés : HTTP, WebSocket et MQTT.

1.7.1 Hypertext Transfer Protocol

Le protocole de transfert hypertexte (Hypertext Transfer Protocol : HTTP) est le protocole de communication derrière le World Wide Web (WWW). Il est basé sur une architecture client-serveur et fonctionne à la manière des demandes et des réponses. HTTP utilise TCP (protocole de contrôle de transmission) pour fournir des connexions fiables [124].

HTTP est un protocole sans état, car le client et le serveur ne maintiennent pas de connexion pendant la communication. La version actuelle est HTTP/1.1 et la version précédente est HTTP/1.0. La nouvelle version HTTP/2 arrive bientôt, qui aura de nombreuses nouvelles fonctionnalités, telles que la poussée du serveur, pour minimiser le nombre de demandes des clients et augmenter la vitesse [32].

1.7.2 WebSocket

WebSocket est un protocole de communication conçu pour les navigateurs Web et les serveurs Web, mais contrairement à HTTP, WebSocket fournit une communication en duplex intégral sur une seule connexion TCP. WebSocket est avec état, car le client et le serveur maintiennent une connexion pendant la communication. Le WebSocket permet une plus grande interaction entre un navigateur et un serveur Web, permet un transfert de données en temps réel et des flux de messages. À ce jour, WebSocket est implémenté dans tous les principaux navigateurs Web, par exemple Firefox 6, Safari 6, Google Chrome 14, Opera 12.10 et Internet Explorer 10 [67, 102].

1.7.3 MQ Telemetry transport (MQTT)

MQ Telemetry transport (MQTT) est un protocole de communication machine à machine légère conçue pour les appareils IoT par IBM. MQTT est basé sur un modèle publisher-subscriber, où publisher publie des données sur un serveur (également appelé broker), et le subscriber s'abonne au serveur et reçoit des données du serveur. Le broker MQTT est responsable de la distribution des messages et peut-être quelque part dans le Cloud [63, 122].

1.7.4 Constrained application Protocole (CoAP)

Constrained application Protocole (CoAP) est un protocole de couche application spécialisé pour les dispositifs IoT contraints, c'est-à-dire les devices avec une puissance de calcul, une consommation d'énergie et une connectivité réseau limitée [104], etc. Il est basé sur des messages de demande et de réponse, similaires à HTTP, mais il utilise UDP (user datagram protocole) plutôt TCP (transmission contrôle protocole). Bien que UDP ne fournisse pas de transmissions fiables, il est beaucoup plus simple, à un sur débit beaucoup plus petit, et donc il est beaucoup plus rapide. CoAP est conçu pour les applications de machine à machine (M2 M) telles que l'énergie intelligente et l'automatisation de la maison/du bâtiment building automation [18].

1.7.5 Extensible Messaging and Presence Protocol (XMPP)

Extensible Messaging and Presence Protocol (XMPP) est un protocole de communication en temps réel standard ouvert basé sur XML (Extensible Markup Language). Il peut fournir une large gamme de services, notamment la messagerie instantanée, la présence et la collaboration. Il est décentralisé et dispose de fonctions de sécurité. Il est également extensible, ce qui signifie qu'il est conçu pour grandir et s'adapter aux changements. Le logiciel XMPP comprend des serveurs, des clients et des bibliothèques [96, 16].

1.8 Système embarqué

Le premier système informatique embarqué moderne en temps réel a été l'ordinateur de guidage Apollo, développé dans les années 1960 par le Dr Charles Stark Draper au Massachusetts

Institute of Technology pour le programme Apollo. L'ordinateur de guidage Apollo a été conçu pour collecter automatiquement des données et fournir des calculs critiques pour le module de commande Apollo et le module lunaire.

En 1971, Intel a sorti la première unité de microprocesseur disponible dans le commerce - l'Intel 4004 - un premier microprocesseur qui nécessitait encore des puces de support et une mémoire externe; en 1978, la National Engineering Manufacturers Association a publié une norme pour les microcontrôleurs programmables, améliorant la conception du système embarqué; et au début des années 80, les composants du système de mémoire, d'entrée et de sortie avaient été intégrés dans la même puce que le processeur, formant un microcontrôleur.

Le système intégré basé sur un microcontrôleur continuerait d'être intégré dans tous les aspects de la vie quotidienne des consommateurs, des lecteurs de cartes de crédit et des téléphones portables aux feux de circulation et aux thermostats.

1.8.1 Définition du système embarqué

Un système embarqué est un système informatique à petite échelle qui fait partie d'une machine ou d'un système électrique/mécanique plus grand. Il est souvent conçu pour effectuer certaines tâches dédiées et souvent un système en temps réel. Il est appelé intégré car le système informatique est intégré à un périphérique matériel. Les systèmes intégrés sont importants, car ils sont de plus en plus utilisés dans de nombreux appareils quotidiens [90], tels que les montres numériques, les appareils photo, les fours à micro-ondes, les machines à laver, les chaudières, les réfrigérateurs, les téléviseurs intelligents et les voitures. Les systèmes embarqués doivent également souvent être de petite taille, à faible coût et avoir une faible consommation d'énergie [99].

Un système embarqué comprend trois composants :

- Le matériel.
- Un logiciel d'application.
- Il dispose d'un système d'exploitation en temps réel (SETR) qui supervise le logiciel d'application et fournit un mécanisme permettant au processeur d'exécuter un processus conformément au calendrier en suivant un plan de contrôle des latences. SETR définit le fonctionnement du système. Il définit les règles lors de l'exécution du programme d'ap-

plication. Un système intégré à petite échelle peut ne pas avoir de SETR.

Nous pouvons donc définir un système embarqué comme un système de contrôle en temps réel basé sur un microcontrôleur, piloté par logiciel et fiable.

La figure 1.4 montre le diagramme schématique d'un système embarqué typique qui comprend un microcontrôleur, des entrées/sorties et des interfaces de communication.

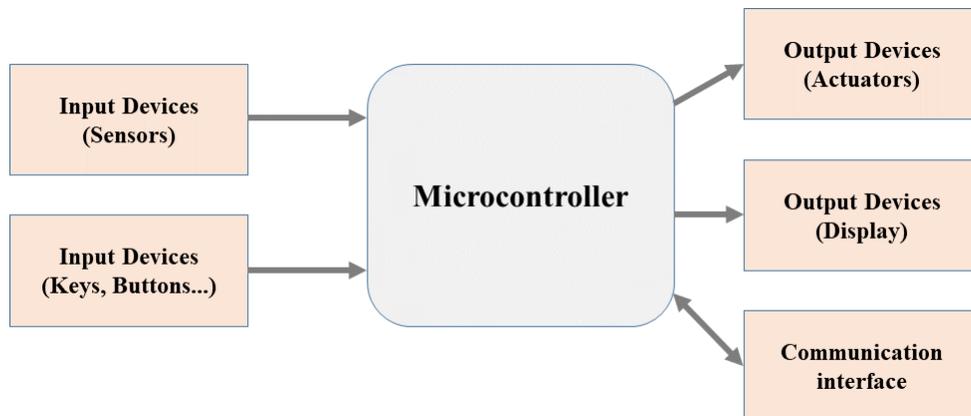


FIGURE 1.4 – Diagramme schématique d'un système embarqué typique

1.8.1.1 Microcontroller

Le microcontrôleur est le cerveau d'un système embarqué, qui orchestre toutes les opérations. Un microcontrôleur est un processeur d'ordinateur avec la mémoire et tous les périphériques d'entrée/sortie. Plus de détails sur les microcontrôleurs seront illustrés dans la section suivante.

1.8.1.2 Les entrées

Un système embarqué interagit avec le monde extérieur via ses entrées et sorties. Les entrées peuvent être des entrées numériques ou des entrées analogiques. Les entrées sont généralement utilisées pour lire les données des capteurs (capteur de température, capteur de lumière, capteur à ultrasons, etc.) ou d'autres types de périphériques d'entrée (touches, boutons, etc...).

1.8.1.3 Les sorties

Les sorties peuvent également être des sorties numériques ou des sorties analogiques. Les sorties sont généralement utilisées pour l’affichage, la commande de moteurs ou d’autres dispositifs (actionneurs).

1.8.1.4 Interfaces de communication

Un système embarqué communique avec d’autres appareils à l’aide d’interfaces de communication, notamment Ethernet, USB (Universal Serial Bus), CAN (Controller Area Network), infrarouge, ZigBee, WiFi et Bluetooth.

1.8.2 Caractéristiques d’un système embarqué

les systèmes embarqués se caractérisent par :

- **Fonction unique** : un système embarqué effectue généralement une opération spécialisée et fait de même à plusieurs reprises. Par exemple, un téléavertisseur fonctionne toujours comme un téléavertisseur.
- **Contraintes strictes** : tous les systèmes informatiques ont des contraintes sur les métriques de conception, mais celles d’un système embarqué peuvent être particulièrement strictes. Les métriques de conception sont une mesure des fonctionnalités d’une implémentation telles que son coût, sa taille, sa puissance et ses performances. Il doit être d’une taille adaptée à une seule puce, doit être suffisamment rapide pour traiter les données en temps réel et consommer un minimum d’énergie pour prolonger la durée de vie de la batterie.
- **Réactif et en temps réel** : de nombreux systèmes embarqués doivent constamment réagir aux changements de l’environnement du système et doivent calculer certains résultats en temps réel sans aucun délai. Prenons un exemple de régulateur de vitesse pour voiture ; il surveille et réagit en permanence aux capteurs de vitesse et de freinage. Il doit calculer à plusieurs reprises l’accélération ou les des accélérations dans un temps limité ; un calcul retardé peut entraîner un échec de contrôle de la voiture.
- **Basé sur des microprocesseurs** : il doit être basé sur un microprocesseur ou un micro-

contrôleur.

- **Mémoire** : il doit avoir une mémoire, car son logiciel est généralement intégré dans la ROM. Il n'a pas besoin de mémoire secondaire sur l'ordinateur.
- **Connecté** : il doit avoir des périphériques connectés pour connecter les périphériques d'entrée et de sortie.
- **Systèmes matériels / logiciels** : le logiciel est utilisé pour plus de fonctionnalités et de flexibilité. Le matériel est utilisé pour les performances et la sécurité.

1.8.3 Structure de base d'un système embarqué

L'illustration suivante montre la structure de base d'un système embarqué

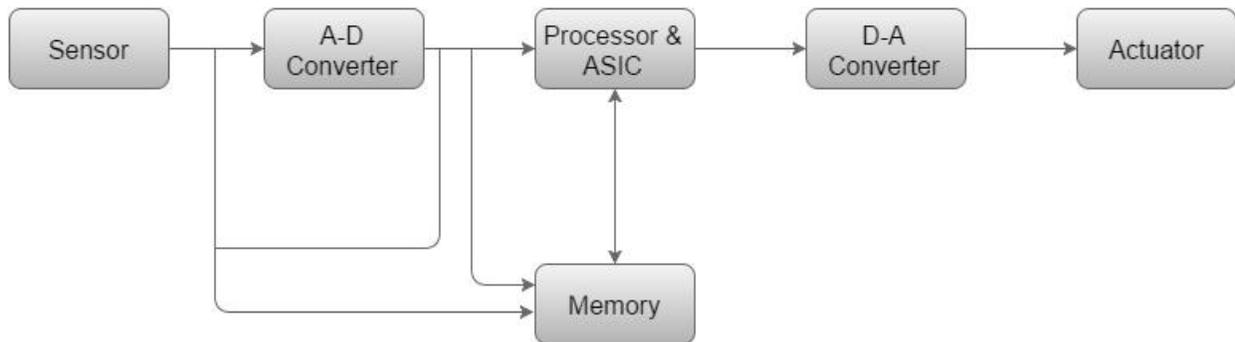


FIGURE 1.5 – Structure de base d'un système embarqué

- **Capteur** : il mesure la quantité physique et la convertit en un signal électrique qui peut être lu par un observateur ou par n'importe quel instrument électronique comme un convertisseur A2D. Un capteur stocke la quantité mesurée dans la mémoire.
- **Convertisseur A-D** : un convertisseur analogique-numérique convertit le signal analogique envoyé par le capteur en un signal numérique.
- **Processeur et ASIC** : les processeurs traitent les données pour mesurer la sortie et les stockent dans la mémoire.
- **Convertisseur D-A** : un convertisseur numérique-analogique convertit les données numériques fournies par le processeur en données analogiques
- **Actionneur** : un actionneur compare la sortie donnée par le convertisseur D-A à la sortie réelle (attendue) qui y est stockée et stocke la sortie approuvée.

1.8.4 Tendances futures des systèmes embarqués

L'industrie des systèmes embarqués devrait continuer de croître rapidement, tirée par le développement continu de l'intelligence artificielle (IA), de la réalité virtuelle (VR) et de la réalité augmentée (AR), de l'apprentissage automatique, de l'apprentissage en profondeur et de l'Internet des objets (IoT).

Le système embarqué cognitif sera au cœur de tendances telles que la réduction de la consommation d'énergie, l'amélioration de la sécurité des appareils embarqués, la connectivité cloud et la mise en réseau maillée, les applications d'apprentissage en profondeur et les outils de visualisation avec des données en temps réel.

1.9 Conclusion

Dans ce chapitre nous avons fait un survol sur l'internet des objets en général en fournissant la définition d'un objet connecté, de son architecture et de l'architecture de l'IoT. On a vu aussi l'historique de l'IoT et les raisons de cette évolution. Ensuite on a expliqué les technologies et les domaines d'application qui seront touchés par l'émergence d'internet des objets ainsi que les défis et comment appliquer le système embarqué.

Dans le chapitre suivant, nous allons présenter dans la première section l'historique du cloud computing, ses propres définitions, leur caractéristiques, ses services et les modèles de déploiement de cloud computing, enfin nous allons citer quelques avantages et inconvénients de cloud computing.

Chapitre 2

Cloud Computing

2.1 Introduction

Il est communément admis que le concept de cloud computing a été initié par le géant d'Amazon en 2002. Le cybermarchand a ensuite investi dans un parc informatique pour pallier les surcharges de serveurs dédiées au commerce en ligne observées pendant les vacances. À cette époque, Internet comptait moins de 600 millions d'utilisateurs, mais l'utilisation du Web et les achats en ligne augmentaient. Malgré cette augmentation, les ressources informatiques d'Amazon sont restées peu utilisées une fois la période des fêtes terminée. Cette dernière a alors eu l'idée de louer ses capacités informatiques le reste de l'année à des clients pour stocker les données et utiliser les serveurs. Ces services étaient accessibles via Internet et avec une adaptation en temps réel des capacités de traitement, toutes facturées à la consommation. Cependant, ce n'est qu'en 2006 qu'Amazon s'est rendu compte qu'un nouveau mode de consommation des ordinateurs et d'Internet émergeait [68].

Avant la naissance du terme cloud computing, utilisé par les informaticiens pour décrire l'immense nébuleuse du net, les services cloud étaient déjà utilisés comme webmail, stockage de données en ligne (photos, vidéos ...) ou partage d'informations sur les réseaux sociaux.

Dans les années 1990, un autre concept avait déjà préparé le terrain pour le Cloud Computing. Il s'agit de l'ASP (Application Service Provider) qui a permis au client de louer l'accès au logiciel installé sur les serveurs distants d'un fournisseur de services, sans installer le logiciel sur ses propres machines. Le Cloud Computing ajoute à cette offre la notion d'élasticité avec la

possibilité d'ajouter de nouveaux utilisateurs et services d'un simple clic de souris [68].

La virtualisation est un concept beaucoup plus ancien qui est à la base du cloud computing. La virtualisation est un ensemble de techniques matérielles ou logicielles utilisées pour exécuter plusieurs configurations informatiques (systèmes d'exploitation, applications, RAM, etc.) sur une seule machine physique pour former plusieurs machines virtuelles qui reproduisent le comportement des machines physiques. C'est le fait de formaliser une offre de services informatiques dématérialisés à la demande en direction des entreprises qui a été le moteur du développement du Cloud Computing en tant que tel.

2.2 Définition du Cloud Computing

Le cloud computing est le mot qui a fait le plus de buzz ces dernières années, vous le trouvez partout, feuillotez un magazine technologique ou visitez n'importe quel site ou blog de l'informatique, vous allez sûrement rencontrer une discussion sur le cloud computing, le seul problème est que non tout le monde s'accorde sur la définition du Cloud, si vous posez la question «Qu'est-ce que le Cloud Computing?» Accélérez les experts dans le domaine Vous aurez dix réponses différentes. Et le cloud est-il à la mode?! Pour certains, ils disent que ce n'est qu'un concept marketing de vendre des produits classiques. En revanche, d'autres y voient l'avenir des applications et une rupture technologique majeure dans l'histoire de l'informatique, pour vous apporter un éclairage objectif nous vous proposons plusieurs définitions par différents acteurs du marché informatique.

En 2009, le NIST (National Institut of Standards and Technologie) a donné une définition du cloud computing largement adoptée et référencée. «Le Cloud Computing est l'ensemble des disciplines, technologies et modèles commerciaux utilisés pour fournir des capacités informatiques (logiciels, plates-formes, matériels) en tant que service à la demande».

Une deuxième définition est donnée en 2009 par l'Université de Californie à Berkeley qui considère que "le Cloud Computing désigne à la fois les applications fournies sur les services Internet et le matériel et les logiciels du système dans les centres de données qui fournissent ces services".

Ces deux définitions commencent à introduire des termes spécifiques au cloud, décrivant le

cloud comme l'ensemble complet des phénomènes allant du faible niveau de matériel au haut niveau de service ou d'applications, ils introduisent également le concept de tout en tant que service référencé. Temps par XaaS (X = Logiciel ou Plateforme ou Matériel ...), dont une partie peut être livrée, mesurée et payable en tant que service [22].

2.3 Caractéristiques du cloud computing

Il y a cinq propriétés qui caractérisent le cloud, nous appelons un service cloud lorsqu'il y a :

2.3.1 Abstraction sur la localisation des données

Le Cloud désigne un ordinateur où les données sont confisquées sans connaître leur localisation géographique. L'application cloud se trouvant à San Francisco, Paris ou Pékin : cela fait peu de différences pour nous. Le mot «Cloud» fait principalement référence à cette abstraction sur le terrain. La métaphore est la suivante : les vrais clouds se déplacent perpétuellement autour de la terre, on ne sait pas les localiser, il en va de même pour le Cloud dont la position géographique des données est inconnue.

Certains acteurs du monde du Cloud jouent sur cette abstraction : ainsi Google entretient un certain mystère autour de la localisation de ses Datacenter. Il est donc impossible de savoir dans quel pays Google stocke vos données, ce qui peut être perturbant pour certains.

Le concept de déplacement perpétuel des clouds peut également avoir un sens avec le cloud : en effet, certains acteurs mettent en place des systèmes de déplacement et de réplication des données entre leurs Datacenters. Ces déplacements ont deux objectifs : disposer de plusieurs copies des données pour assurer leur conservation en cas de panne, et optimiser le remplissage des différents Datacenters, c'est-à-dire éviter les serveurs à moitié pleins. Ces mouvements étant automatiques, personne ne connaît l'emplacement des données, pas même les responsables des Datacenters.

2.3.2 Souscription en ligne

vous vous inscrivez via un formulaire, vous recevez un e-mail de confirmation, et le service peut être utilisé quelques secondes plusieurs tard. . . S'abonner en ligne semble naturel aujourd'hui, à l'heure du web. Mais il ne faut pas oublier qu'il existe encore un certain nombre de services qui ne peuvent pas être souscrits en ligne : les banques en ligne (à quelques exceptions près), l'assurance, etc. Cependant, ces services sont informatisés et ne nécessitent pas de rencontres physiques.

2.3.3 Tarification « Freemium »

Ce terme est la contraction des mots "Gratuit" et "Premium". Cela signifie qu'il existe une offre gratuite, parfois limitée dans le temps ou la capacité, ou offrant moins de fonctionnalités, et une offre payante pour les fonctionnalités avancées. L'offre payante est facturée en fonction des services disponibles ou en fonction de la capacité utilisée. Par exemple, la solution de stockage de fichiers DropBox de Cloud est gratuite pour 2 Go d'espace, puis devient payable au-delà.

2.3.4 Accès depuis n'importe quel appareil

Ordinateur, tablette, iPhone (Apple), Android (Google) ou Windows Phone (Microsoft). Le Cloud permet l'accès depuis n'importe quel appareil : à la maison, au travail, depuis le domicile d'un ami, depuis un cybercafé à l'étranger ... offre donc la certitude qu'il sera possible d'accéder à ses données où que vous soyez, à condition d'avoir une connexion à Internet et un navigateur Web. Cela peut être pratique en cas d'incident à l'étranger : il est astucieux d'avoir une copie de son passeport et de sa licence dans le Cloud pour anticiper les problèmes devant les voyageurs [\[93\]](#).

2.4 Techniques de virtualisation

Dans cette section nous présentons les concepts clé de la virtualisation.

2.4.1 Définition

La virtualisation a été la première pierre de l'ère du cloud computing, c'est l'ensemble des techniques et outils permettant de réaliser plusieurs systèmes d'exploitation sur une même machine physique pour offrir une meilleure utilisation des ressources, cette technologie vient répondre à certains problèmes tels que :

1. **Sous-exploitation des serveurs physiques** : on estime que dans un Datacenter privé, le taux d'utilisation moyen est de 10%. Cela passe à 35% sur une architecture virtuelle.
2. **La croissance du nombre de serveurs physiques** : les ressources physiques d'un serveur seront partagées entre différents serveurs virtuels, ce qui permet de ne pas acheter plusieurs serveurs physiques.
3. **Sécurité et fiabilité** : isoler les services sur différents serveurs dans les systèmes de virtualisation.

2.4.2 Virtualisation

La virtualisation permet de coexister plusieurs systèmes d'exploitation complètement isolés sur un même hôte. On distingue plusieurs techniques de virtualisation à savoir : l'isolement, la para virtualisation, et la virtualisation complète.

1. **L'isolation** : il est possible de diviser un système d'exploitation en plusieurs espaces mémoire ou plusieurs contextes. Chaque contexte est géré par le système d'exploitation hôte. Cette isolation permet d'exécuter plusieurs fois la même application pour ne s'exécuter qu'une seule fois par machine. Les programmes de chaque contexte ne peuvent communiquer qu'avec les processus et les ressources associés à leur contexte. L'isolement n'est lié qu'aux systèmes Linux.

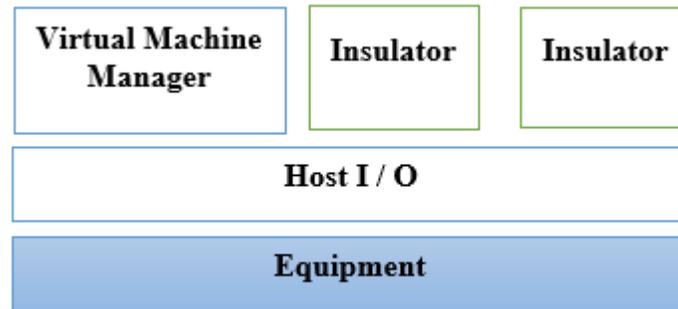


FIGURE 2.1 – Architecture de l'isolation

2. **Para virtualisation (virtualisation de type 1)** : c'est une technique de virtualisation qui présente à la machine invitée une interface logicielle similaire mais non identique au matériel réel. Ainsi, il permet aux systèmes d'exploitation invités d'interagir directement avec le système d'exploitation hôte et, par conséquent, ils seront conscients de la virtualisation.

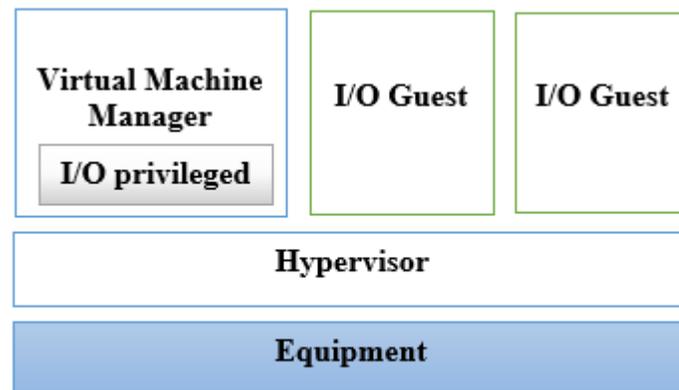


FIGURE 2.2 – Architecture de la paravirtualisation

3. **Virtualisation complète (Virtualisation de type 2 virus)** : il s'agit d'une technique de virtualisation qui crée un environnement entièrement virtuel. En utilisant ces techniques, le système d'exploitation invité n'interagit pas directement avec le système d'exploitation hôte et pense donc qu'il s'exécute sur une vraie machine physique. Cette technique de virtualisation permet uniquement la virtualisation des SE avec la même architecture matérielle que l'hôte.

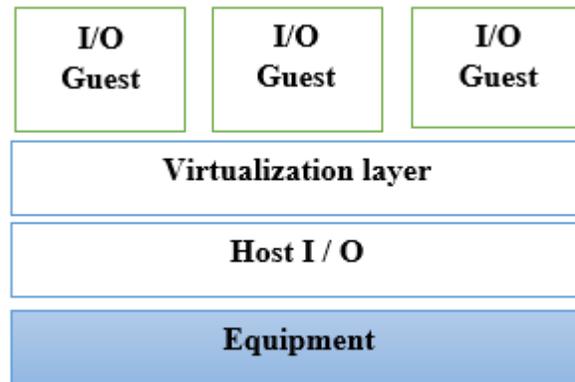


FIGURE 2.3 – Architecture de la virtualisation complète

	isolation	Paravirtualisation	Virtualisation complète
Types invités d'OS	Linux	Type différent mais avec une architecture identique (doit être adapté à la couche de virtualisation -> conscient d'être virtualisé)	Type différent mais avec une architecture identique (non adapté à la couche de virtualisation-> croit dialoguer directement avec le matériel)
Performance	++ (Supplément faible)	+++ (Les invités SE travaillent avec la conscience d'être virtualisés)	+ (L'unité centrale de traitement, c'est-à-dire le CPU, la RAM, ainsi que la mémoire de stockage, sont directement accessibles aux machines virtuelles)
Simplicity	+++	+	++
Exemples	OpenVZ	Xen,HyperV	KVM, VirtualBox

En conclusion, nous pouvons dire que :

- L'isolement est considéré comme la solution la plus efficace, cependant, son inconvénient est qu'il est incomplet, donc le système d'exploitation doit être du même type Linux.
- La paravirtualisation suppose que le noyau de l'OS invité est légèrement modifié. Par conséquent, si le système n'a pas de fonctions dédiées à la paravirtualisation dans son noyau, cette technique devient inutilisable. L'objectif principal de cette technique est de fournir un accès presque identique aux ressources matérielles entre les systèmes hôte et invité.
- La virtualisation complète permet d'exécuter le système d'exploitation invité en mode natif sans modification. En revanche, cette solution est considérée comme la moins efficace car le système invité ne participe pas au processus de virtualisation et doit traverser la couche de virtualisation pour accéder aux ressources matérielles [92].

2.5 Services Cloud

Les services proposés depuis le cloud peuvent être classés en trois grandes catégories, et les définitions choisies sont celles données par le NIST :

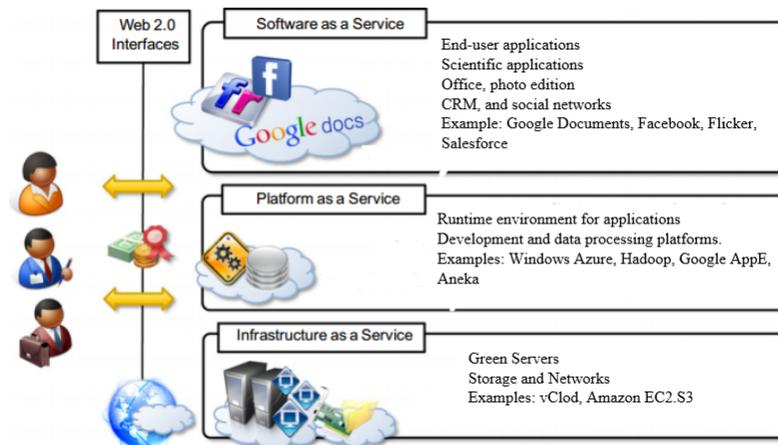


FIGURE 2.4 – Modèle de référence du cloud computing

Les différentes couches de chaque catégorie sont présentées dans la figure 2.5

2.5.1 Software as a Service (SaaS)

Logiciel consommé en tant que service, le fournisseur de SaaS Cloud possède et gère l'ensemble de sa plateforme (du matériel au logiciel). Dans ce modèle, le client cloud utilise le logiciel mais ne s'occupe pas de la pile ci-dessous (plateforme d'application, matériel ...) ou de l'installation du logiciel. Quelques exemples d'utilisation du modèle SaaS : E-mail, CRM (gestion de la relation client). Dans une solution SaaS, le contrôle des données est partagé entre le client (qui crée et utilise les données) et le fournisseur de cloud qui héberge, stocke, sécurise et sauvegarde les données).

2.5.1.1 Les avantages

- Plus d'installation, plus de mises à jour (elles sont continuées chez le fournisseur).
- Payez à l'utilisation.
- Testez facilement de nouveaux logiciels.

2.5.1.2 Les inconvénients

- Limitation par définition du logiciel proposé.
- Aucun contrôle sur le stockage et la sécurisation des données associées
- La réactivité des applications Web n'est pas toujours idéale. Vers le logiciel [85].

2.5.2 Platform as a Service (PaaS)

Une plate-forme sur laquelle les développeurs ou les éditeurs de logiciels peuvent déployer des applications. La pile (la plate-forme d'application, le système d'exploitation, le matériel, le réseau) est détenue et gérée par le fournisseur de services.

2.5.2.1 Les avantages

- Le déploiement est automatisé (aucun logiciel supplémentaire à acheter ou à installer).

2.5.2.2 Les inconvénients

- Limitez une ou deux technologies (par exemple Python ou Java pour Google AppEngine.NET pour Microsoft Azure, propriétaire de force.com).
- Aucun contrôle des machines virtuelles sous-jacentes.
- Convient uniquement aux applications Web.

2.5.3 Infrastructure as a Service (IaaS)

La plateforme sur laquelle les administrateurs informatiques pourront déployer une infrastructure (machine (s) virtuelle (s) + base d'applications + applications ..etc) Ce modèle, qui est une évolution des Datacenters virtualisés, permet au client de ne pas tenir compte du modèle physique (gestion des serveurs physiques, des éléments liés aux centres de données comme l'électricité, la climatisation, la sécurité physique). Dans ce modèle, le fournisseur contrôle le matériel et la couche de virtualisation. Au niveau des données, le contrôle est partagé au niveau de la machine virtuelle (qui est stockée et sauvegardée par le fournisseur de cloud IaaS).

2.5.3.1 Les avantages

- Grande flexibilité.
- Contrôle complet du système (administration à distance par SSH ou Remote Desktop, RDP), qui permet d'installer tout type de logiciel d'entreprise.

2.5.3.2 Les inconvénients

- Besoin d'administrateur système quant aux solutions de serveurs traditionnels sur site [85].

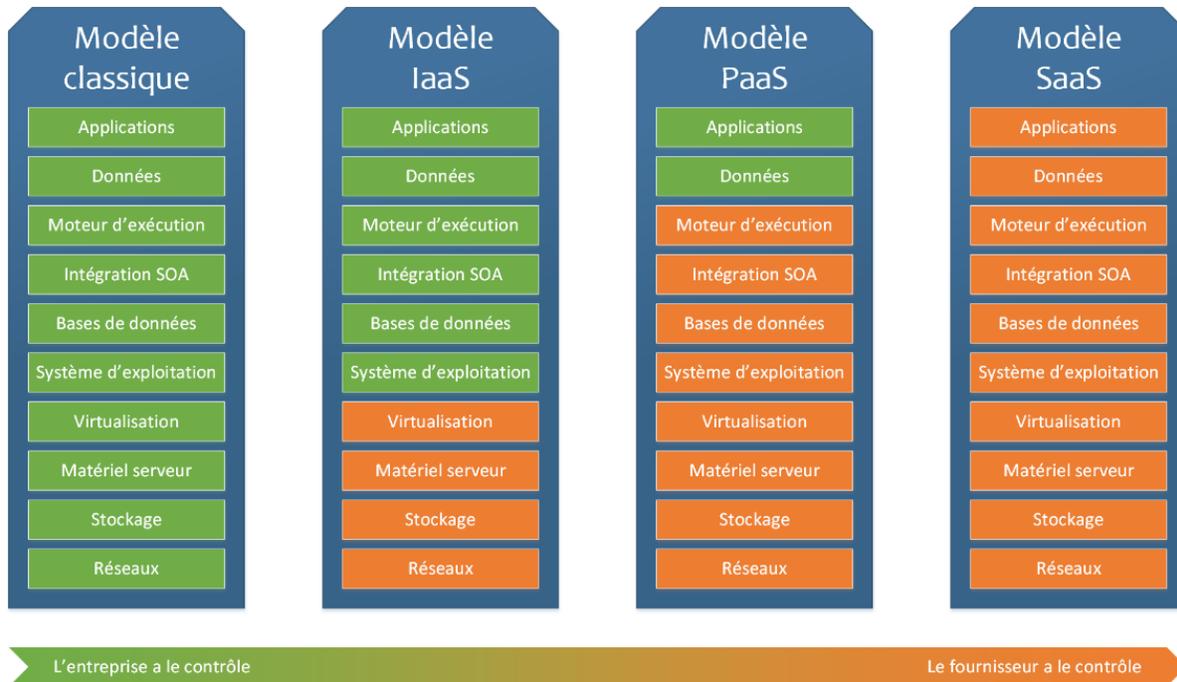


FIGURE 2.5 – Différents types de services de cloud computing

2.6 Modèles de cloud computing

La figure 2.6 montre les différents modèles de Cloud Computing, il en existe quatre modèles :

2.6.1 Cloud privé

Dans lequel toutes les ressources sont exclusivement mises à la disposition d'un seul client. Le cloud privé peut être géré par l'entreprise utilisatrice elle-même ou par un prestataire externe qui met à disposition de l'utilisateur un parc de machines adapté à la demande de l'utilisateur (Cloud privé virtuel). Notez qu'une même infrastructure peut accueillir plusieurs clouds privés virtuels appartenant à différents utilisateurs, chacun pouvant accéder à son cloud privé via son réseau.

2.6.2 Cloud public

Les utilisateurs ont accès aux services cloud via Internet public sans savoir exactement où leurs données sont hébergées ni où leur traitement est effectué. Les ressources informatiques et les bases de données de l'utilisateur peuvent être hébergées dans n'importe quel centre de données du fournisseur et peuvent passer d'un centre de données à un autre pour optimiser les capacités du fournisseur.

TABLE 2.1 – Principaux acteurs mondiaux du cloud public

IaaS	PaaS	SaaS
<ul style="list-style-type: none"> - Amazon-<i>o</i>_res EC2 and AWS - Microsoft-<i>o</i>_re Azur 	<ul style="list-style-type: none"> - Microsoft-<i>o</i>_re Azur - Google-<i>o</i>_re, Google App Engine 	<ul style="list-style-type: none"> - Google-<i>o</i>_re, Google Apps (messagerie et bureautique) - Salesforce-CRM (Gestion de la relation client) - Microsoft-<i>o</i>_re, <i>o</i>_ce 365 (outils collaboratifs)

2.6.3 Hybride cloud

Ils combinent infrastructure et cloud privé et public. Certaines données ou infrastructures sont gérées en interne par l'entreprise, dans ses locaux ou chez un fournisseur et communiquent avec les ressources Cloud. Le cloud hybride permet de différencier les lieux de traitement des données selon qu'elles sont stratégiques ou non : les données sensibles peuvent alors être traitées au sein des murs de l'entreprise tandis que les autres seront traitées par un cloud public plus rentable, plus efficace. Le cloud public peut également être une solution pour lisser un pic d'activité lorsque les capacités de l'entreprise sont obsolètes.

2.6.4 Cloud communautaire

Permettant à plusieurs entreprises où organisations de partager des ressources en mode Cloud, qui sont alors exclusivement dédiées à ces organisations. Le Cloud communautaire peut être géré par des organisations membres ou par un fournisseur externe. Le cloud communau-

taire peut également permettre à plusieurs utilisateurs de construire un cloud aux caractéristiques d'un cloud privé en termes de sécurité et de ressources dédiées, à moindre coût et avec une garantie d'indépendance vis-à-vis d'un prestataire de services de cloud public [68].

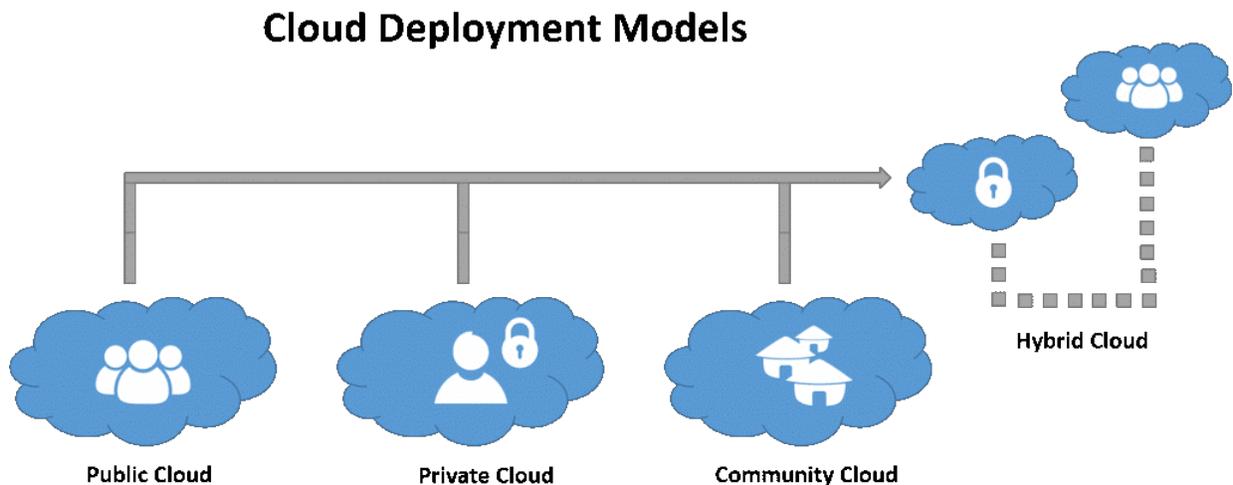


FIGURE 2.6 – Modèles de déploiement cloud

2.7 Avantages et inconvénients du Cloud Computing

Le Cloud Computing représente la révolution dans l'informatique dans ces dernières années, ce paradigme-là arrive avec des avantages qui peuvent corriger les limites des autres technologies existées. Dans cette section, nous allons présenter quelques points forts de ce paradigme. Lorsqu'on parle d'un nouveau paradigme, les chercheurs ou les clients peuvent rencontrer toujours quelques limites [21]. Les avantages et les limites de ce paradigme seront présentés comme suit :

- **Flexibilité** : le Cloud utilise multi-locataire pour ces ressources au cours de l'exécution, une application peut utiliser plusieurs ressources. Cela offre une possibilité de demander d'autres ressources s'il est besoins [21].
- **Optimisation des coûts** : réduction des effectifs informatiques et fixation du prix en fonction de la durée d'utilisation des ressources informatiques sans investissement initiale lourd.
- **Évolutivité** : ça fait référence à un utilisateur Cloud (entreprise) qui peut utiliser d'énormes

calculs de données dans un temps spécifique. A la fin du processus, le système peut retourner aux normes, tous sans nécessiter ces serveurs lourds [21].

- **Portabilité :** les organisations peuvent utiliser leurs puissances informatiques partout où les utilisateurs peuvent avoir un accès dans n'importe quelle localisation géographique [21].
- **Sécurité :** dans le Cloud, les fournisseurs utilisent des stratégies afin de garantir la vie privée de chaque utilisateur par : la réplication des données, plan de reprise d'activité, etc.
- **Simplicité d'utilisation :** le Cloud offre des applications et des services installés et faciles à utiliser à travers des pages web.

Comme nous avons mentionné, chaque nouvelle technologie arrivée porte des avantages et malheureusement suivi par quelques limites rencontrées par des utilisateurs du Cloud. Ces limites sont présentées comme suit :

- **La gestion d'énergie :** afin de définir un plan d'utilisation des ressources, le fournisseur doit définir une stratégie pour la gestion d'énergie (consommation d'électricité) [21]
- **Confidentialité et sécurité :** dans n'importe quelle technologie, la sécurité pose toujours des problèmes. Le problème concerne les attaques lors des opérations du transfert de données. Dans le cloud, ce problème est posé dans les cas des cloud publiques [21].
- **Gestion de ressources :** ce problème représente toujours les limites de chaque technologie. A cause de la nature multidimensionnelle des machines virtuelles, la gestion des ressources sera compliquée .
- **Dépendance :** en cas ou l'entreprise (ou client final) souhaite des fonctionnalités très spécifiques, il est peut être difficile de convaincre le fournisseur de proposer ces fonctionnalités. Le client final doit choisir un fournisseur en qu'il a la confiance.
- **Migration vers une autre offre difficile :** il n'existe pas pour l'instant un standard entre les différents acteurs du domaine, donc, le risque d'incompatibilité du transfert de données.

2.8 Edge computing

L'edge computing se définit comme une architecture informatique destinée aux environnements IoT, dans laquelle les ressources informatiques, la capacité de stockage et la puissance de calcul sont maintenues au plus près des équipements terminaux et des capteurs qui génèrent les données. Le concept représente ainsi une alternative aux solutions de Cloud ordinaires avec des serveurs centralisés.

Le mot « edge » vient de l'anglais et signifie bord ou périphérie. Ce terme fait allusion au fait que le traitement des données ne se fait plus dans le Cloud, mais il est décentralisé, en périphérie du réseau. L'edge computing peut ainsi offrir une option que le Cloud n'est pas capable de proposer, à savoir des serveurs capables d'interpréter sans délai les données de masse générées par des usines, des réseaux de distributions ou des systèmes de circulation « intelligents », et de prendre immédiatement les mesures nécessaires en cas d'incidents.

2.8.1 Fondamentaux de l'edge computing

L'edge computing est une nouvelle forme d'architecture pour les environnements IoT bien qu'elle n'ait pas directement recours à de nouveaux composants de réseau. Au contraire, elle s'appuie sur d'anciennes technologies dans un format compact, mais employées sous une nouvelle dénomination. Voici un aperçu des éléments de base de l'edge computing.

2.8.1.1 Edge

dans le jargon informatique, le mot « edge » désigne la périphérie du réseau. Quant à savoir quels seront les éléments implantés en périphérie du réseau, cela dépendra de la configuration mise en place. Dans des réseaux de télécommunication, ce sera par exemple un téléphone portable qui représentera la périphérie du réseau; et dans un système de voitures autonomes interconnectées, chaque véhicule. On parle dans ce cas d'un edge device.

2.8.1.2 Edge device

on entend par edge device tout appareil situé en périphérie de réseau, et qui génère des données. Les sources de données possibles sont par exemple des capteurs, des machines, des

véhicules ou tous les autres appareils intelligents dans un environnement IoT, comme des lave-linge, des détecteurs d'incendie, des ampoules ou des thermostats pour radiateur.

2.8.1.3 La passerelle Edge

la passerelle Edge est une instance de calcul implantée à la transition entre deux réseaux. Dans des environnements IoT, les passerelles Edge sont utilisées comme nœuds entre l'Internet des objets et le réseau central. On a aussi de puissants routeurs, capables de supporter de fortes puissances de calcul pour assurer le traitement des données de l'IoT. Pour ce faire, les passerelles Edge disposent de diverses interfaces permettant de transférer les données soit par câble, soit par radio, et de standards de communication, comme l'Ethernet, le Wifi, le Bluetooth, la téléphonie 3G, LTE, Zigbee, Z-Wave, CAN-Bus, Modbus, BACnet ou SCADA.

2.8.2 Edge computing et Fog computing

L'approche visant à étendre le Cloud autour des instances de calcul n'est pas une nouveauté. En 2014 déjà, le groupe américain Cisco a créé le terme marketing, baptisé « fog computing ». Ce concept est basé sur un traitement décentralisé des données dans ce qu'on appelle des « nœuds fogs ». Les nœuds fog représentent de mini-centres de calcul positionnés en amont du Cloud, constituant une couche intermédiaire dans le réseau (on parle de « couche fog »). Les données générées dans des environnements IoT ne sont donc pas directement envoyées dans le Cloud. Elles sont d'abord collectées dans des Fog Nodes, où elles sont interprétées avant d'être sélectionnées pour d'autres formes de traitement.

L'edge computing est aujourd'hui considéré comme faisant partie du fog computing, où les ressources informatiques, comme la puissance de calcul et la capacité de stockage sont rapprochées au mieux des équipements IoT, en périphérie du réseau. Dans des architectures de fog computing, le traitement des données se fait d'abord au niveau de la couche fog, tandis que dans des concepts d'edge computing, il est exécuté au niveau de puissants routeurs IoT, et même parfois directement sur les appareils ou sur les capteurs. On peut parfaitement envisager une combinaison des deux concepts. Le graphique ci-dessous montre une telle architecture avec une couche Cloud, fog et edge.

2.8.3 Domaines d'application pour les architectures d'edge computing

Les domaines d'application pour l'edge computing viennent généralement de l'environnement IoT, et représentent des projets d'avenir au même titre que le concept d'une architecture Cloud décentralisée. Un facteur de croissance important de la technologie de l'edge computing est le besoin croissant de systèmes de communication fonctionnant en temps réel. Le traitement décentralisé des données est une technologie clé pour les projets suivants :

- * La communication de véhicule à véhicule
- * Le réseau électrique intelligent
- * Le Smart Factory

Un véhicule connecté sera, à l'avenir, bien plus qu'un véhicule équipé d'une connexion Internet. L'avenir de la mobilité permettra la mise en place de systèmes d'avertissement gérés par le Cloud, permettant une communication de véhicule à véhicule, voire des moyens de transport complètement autonomes dans leurs déplacements. La condition est cependant que l'on puisse disposer d'une infrastructure permettant d'échanger en temps réel des données entre les véhicules et les différents points de communication tout au long de l'itinéraire.

Le réseau électrique de l'avenir sera lui aussi adaptatif, capable de s'auto-réguler automatiquement en fonction des besoins grâce à des systèmes de gestion d'énergie décentralisés. Dans le cadre de la transition énergétique, le réseau électrique intelligent représentera une technologie clé. En effet, la conversion vers des énergies renouvelables impose de nouveaux défis aux réseaux d'électricité. Au lieu d'avoir quelques gros producteurs centralisés, on aura de nombreux petits producteurs d'énergie décentralisés, avec des dispositifs de stockage qui devront être connectés avec les consommateurs. Certains d'entre eux seront eux-mêmes producteurs d'énergie, notamment grâce aux panneaux solaires. Les réseaux intelligents ne se contentent plus de transporter le courant électrique : ils délivrent également des données applicables à sa production, son stockage et sa consommation. Ceci permet à chacun de réagir en temps réel à la moindre modification. L'objectif est de maintenir la stabilité des réseaux électriques malgré leur complexité croissante, et d'assurer une meilleure efficacité grâce à une compensation intelligente des charges. Pour pouvoir saisir, sauvegarder et traiter dans les meilleurs délais toutes ces données générées, nous avons besoin de nouveaux concepts de Cloud, comme l'edge computing et le fog computing.

On entend par Smart Factory (usine du futur) des sites de production et des systèmes de logistique qui s'organisent eux-mêmes. Dans l'idéal, on n'a aucune intervention humaine dans de tels concepts. Une usine intelligente est un système interconnecté d'appareils, de machines et de capteurs qui communiquent entre eux par Internet pour mener à terme des processus de fabrication. Le système de communication Smart Factory inclut le produit fini dans son système et peut donc réagir automatiquement face à des demandes de devis. Grâce à des systèmes d'intelligence artificielle et à un apprentissage autonome, on a des processus de maintenance qui optimisent la production. Ceci demande une infrastructure informatique capable de traiter sans délai de gros volumes de données, et de réagir rapidement à des imprévus. Les systèmes de Cloud traditionnels échouent pour des raisons de latence. Les architectures d'edge computing et de fog computing peuvent résoudre ce problème grâce à un traitement des données partagé.

2.8.4 Avantages et inconvénients de l'edge computing

Le tableau ci-dessous présente les avantages et les inconvénients d'une architecture d'edge computing, comparée à un environnement Cloud traditionnel [80].

Avantages	Inconvénients
<p>Traitement des données en temps réel : dans les architectures d'edge computing, les unités de calcul sont rapprochées au mieux des sources de données, favorisant une communication en temps réel. On évite ainsi le problème récurrent de latence rencontré avec les solutions de Cloud plus classiques.</p>	<p>Des structures de réseau plus complexes : un système de répartition est bien plus compliqué qu'une architecture Cloud centralisée. Un environnement edge computing est un ensemble hétérogène de plusieurs composants de réseau, venant en partie de divers fabricants, et qui communiquent les uns avec les autres grâce à un grand nombre d'interfaces.</p>
<p>Débit utile réduit : l'edge computing privilégie un traitement des données en local au niveau de passerelles Edge. Seules les données qui ne peuvent pas être traitées localement, ou qui doivent être mises en lignes, sont téléchargées dans le Cloud.</p>	<p>Les frais d'acquisition pour du matériel Edge : les architectures de Cloud se distinguent avant tout par le fait qu'il y a beaucoup moins d'équipement matériel à installer localement. On perd cet avantage si on opte pour des systèmes à répartition.</p>
<p>La sécurité des données : avec une solution d'edge computing, la majeure partie des données reste dans le réseau local. Dans une telle configuration, les entreprises auront plus de facilité à se conformer aux exigences de conformité.</p>	<p>Un niveau de maintenance plus élevé : un système décentralisé, composé de plusieurs nœuds de calcul, nécessite plus d'entretien et d'administration qu'un centre de données.</p>

2.9 Conclusion

Dans ce chapitre, nous avons donné un aperçu sur la technologie du Cloud Computing. Comme nous avons vu que le Cloud pense à résoudre les problèmes posés dans les autres technologies. De plus, le Cloud représente une révolution car il offre des services de plusieurs types. Grâce aux bénéfices offerts par le fournisseur du Cloud, les utilisateurs finaux et les entreprises trouvent que le Cloud est un bon choix pour l'utilisation de ces services.

Avec toutes ces solutions et les avantages produits dans le Cloud, il existe toujours des li-

mites que nous avons détaillés dans ce chapitre. On a parlé aussi de la sécurité par ce qu'elle représente la vie privée des clients. Mais tout le monde maintenant utilise le Cloud Computing que ce soit personnel ou partagé avec les membres des équipes de la société.

Dans le prochain chapitre nous allons présenté un état de l'art sur les approches de l'Internet des objets dans le smart house.

Chapitre 3

Approches de l'IoT dans la maison intelligente

3.1 Introduction

Dans ce chapitre, nous allons donner un aperçu sur l'évolution des constructions des maisons jusqu'à la naissance du terme maison intelligente (Smart home), ensuite nous allons définir ce qui est une maison intelligente, En plus, nous allons présenter un état de l'art sur les approches utilisant la technologie de l'Internet des objets dans un smart house.

3.2 Historique

Les constructions ont évolué en fonction des besoins de l'homme, l'homme de l'histoire et nomade et se déplace au gré des saisons et des migrations animales, pour se mettre l'abri il fabrique des huttes faites de branchages d'ossements et de peau.

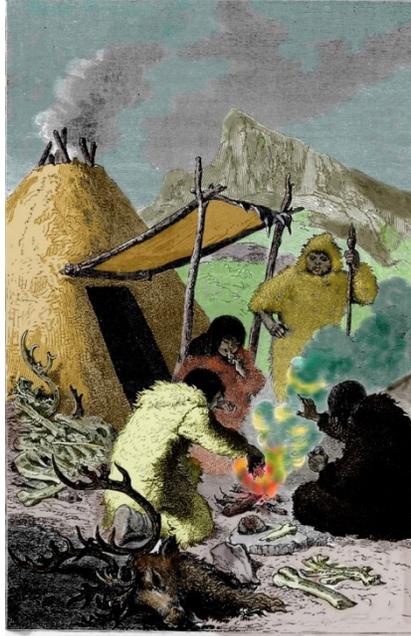


FIGURE 3.1 – Maisons de Néandertal

Il y a environ 12 mille ans les hommes inventent l'élevage et l'agriculture, n'ayant plus besoin de se déplacer pour trouver leur nourriture, ils bâtissent des habitats fixes et se regroupent dans les premiers villages dont les maisons sont rondes construite en bois ou en terre et recouvertes de feuillages,



FIGURE 3.2 – Maisons rondes de bois

C'est en Mésopotamie il y a cinq mille ans que naissent les premières villes progressivement

les maisons deviennent rectangulaire ou carrés formé ou pratiques pour être cloisonné en différents espaces et permettent d'assembler les maisons les unes contre les autres autour de petites rues.



FIGURE 3.3 – Maisons rectangulaires ou carrées

En Egypte les maisons traditionnelles sont construites en briques de terre et en paille et possèdent déjà plusieurs pièces plus tard les riches se font construire des demeures de plusieurs étages, peu les grandes villes du bassin méditerranéen s'organisant en quartier, séparant habitations ateliers culte religieux marché; elles sont déjà pourvus de canalisations qui assurent l'arrivée d'eau dans les maisons.

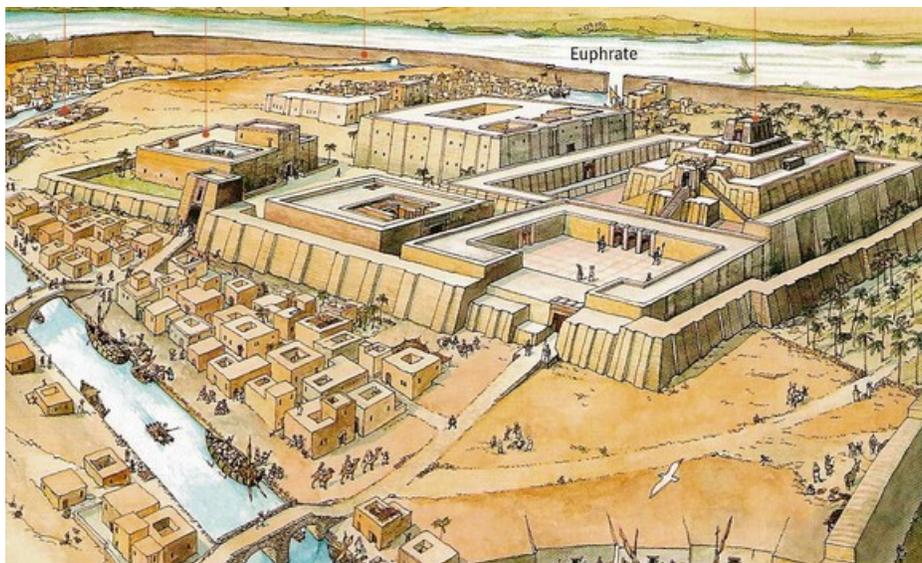


FIGURE 3.4 – Une grande ville méditerranéen

La maison gauloise est construite avec les matériaux disponibles proximité du bois pour sa structure, il faut de la charpente du torchis qui est une mélange de terre et de paille pour les murs, de la paille pour le toit, sans fenêtre cette maison est sombre, et est constitué d'une pièce qui accueille un foyer pour s'éclairer et cuisine et d'une réserve provisions et d'un grenier plus chaud pour y dormir.



FIGURE 3.5 – Modèle d'une maison gauloise

En Europe c'est à partir du 12^{ème} siècle que la pierre remplace peu à peu le bois jugé trop fragile face aux incendies dans les bourgs et les villes. L'époque est propice pour la construction de fortifications devant assurer la défense de la population en cas d'attaque, la cité médiévale compte de nombreux édifices, habitations, ateliers, boutiques, tandis que les plus riches habitent des logements spacieux et confortables les pauvres vivent dans des pièces étroites sombre et souvent insalubres.



FIGURE 3.6 – Maison de la 12^{ème} siècle

Au 16^{ème} siècle, l'architecture renaissance venus d'Italie se propage en Europe, les riches demeures rappelle l'architecture romaine par leurs formes leurs colonnes leur proportion les façades sont plus régulières et pourvue de grandes fenêtres en verre. Au 19^{ème} siècle les grands industriels implantent des cités ouvrières pour loger leur main d'œuvre proximité des usines, ces petites maisons en briques sont toutes identiques et sans confort.



FIGURE 3.7 – Cité ouvrière

Le 20^{ème} siècle est marqué par l'exode rural et le développement des villes. Pour faire face au manque de place on construit en hauteur c'est l'ère des gratte-ciel avec des matériaux nouveaux, béton, acier, aluminium, verre. Dans les banlieues apparaissent des tours et des barres d'immeubles, les habitants de plus en plus éloignés des centres villes et passent beaucoup de temps dans les transports.



FIGURE 3.8 – Cité du 20^{ème} siècle

Aujourd'hui on cherche à diminuer notre consommation d'énergie ce qui nécessite de réduire les déplacements en voiture et de construire des logements écologiques, pratiques, plus économes, confortables et respectueux de l'environnement.

Cette évolution des constructions a été toujours liée avec le déroulement des besoins humains, et avec l'émergence de l'IT comme concept de confort, il est devenu nécessaire de le joindre dans la vie quotidienne pour fournir un environnement plus confortable pour l'habitant. Ces dernières années, un nouveau type de maisons a vu le jour, que nous appelons les maisons automatiques et les maisons intelligentes (smart home).

Le terme «maisons intelligentes» a été utilisé pour la première fois de manière officielle en 1984 par l'Association américaine des constructeurs de maisons, bien que les premières "maisons câblées" aient été en fait construites par des amateurs au début des années 1960. Et ce développement est la clé de ce que l'on entend par "maisons intelligentes". Car une maison n'est pas intelligente parce qu'elle est bien construite, ni parce qu'elle utilise efficacement l'espace; ni parce qu'elle est écologique, en utilisant l'énergie solaire et en recyclant les eaux usées, par exemple. Une maison intelligente peut inclure ces objets, et c'est d'ailleurs souvent le cas,

mais ce qui la rend intelligente, ce sont les technologies interactives qu'elle contient.

Les objets d'une maison intelligente sont reliés entre eux et accessibles par un point d'accès (Gateway) - un smartphone, une tablette, un ordinateur portable ou une console de jeu. Les serrures de porte, les télévisions, les thermostats, les moniteurs domestiques, les caméras, les lumières et même les appareils tels que le réfrigérateur peuvent être contrôlés par un seul système domotique. Le système est installé sur un téléphone portable ou un autre appareil en réseau, et l'utilisateur peut créer des horaires pour que certains changements prennent effet.

En 2016, le marché mondial de la domotique était estimé à environ 36 milliards de dollars, et avec l'adoption croissante des appareils fonctionnant sur Internet, on prévoit que le marché pourrait atteindre des revenus de l'ordre de 80 milliards de dollars d'ici 2020. La croissance du marché de la domotique au sens large a toutefois été confrontée à des défis en raison de sa nature axée sur la commodité, plutôt que sur la nécessité.

3.3 Définition de la maison intelligente

Une maison intelligente est une résidence équipée d'un réseau de communication, reliant des capteurs, des appareils ménagers et d'autres dispositifs électroniques et électriques, qui peut être surveillée, accédée ou contrôlée à distance et qui fournissent des services qui répondent aux besoins de ses habitants[14].

Une "maison intelligente" peut être définie comme une résidence équipée de technologies informatiques qui anticipent et répondent aux besoins des occupants, en s'efforçant de promouvoir leur confort, leur commodité, leur sécurité et leurs loisirs grâce à la gestion de la technologie à l'intérieur de la maison et aux connexions avec le monde extérieur [47].

Une maison intelligente est une installation domestique pratique où les objets et les dispositifs peuvent être automatiquement contrôlés à distance depuis n'importe quel endroit connecté à Internet dans le monde à l'aide d'un smartphone ou d'un autre dispositif en réseau. Les objets d'une maison intelligente sont interconnectés par internet et l'utilisateur peut contrôler des fonctions telles que l'accès sécurisé à la maison, la température, l'éclairage et le cinéma maison. Les termes connexes comprennent la "domotique" et le "bâtiment intelligent".

3.4 Travaux connexes

Afin de mieux analyser des travaux connexes, nous avons détaillé points suivants : Travaux exploitant l'IoT dans le smart house sans le cloud, Travaux exploitant l'IoT dans le smart house via le cloud, Travaux sur les architecture des smart houses, Travaux sur les architecture des objets et Travaux sur les robots basés internet des objets.

3.4.1 Travaux exploitant l'IoT dans le smart house sans le cloud

Dans [40], les auteurs ont proposé un système de contrôle intelligent basé sur les technologies de l'internet des objets. Le système de contrôle domestique intelligent utilise un contrôleur central intelligent pour configurer un réseau de capteurs et d'actionneurs sans fil à fréquence radio 433 MHz (WSAN). Une série de modules de contrôle, tels que des modules de commutation, des modules de contrôle de radiofréquence, ont été développés dans le WSAN pour contrôler directement tous les types d'appareils ménagers. Les serveurs d'applications, les ordinateurs clients, les tablettes ou les smartphones peuvent communiquer avec le contrôleur central intelligent via un routeur sans fil via une interface Wi-Fi. Puisqu'il a WSAN comme couche de contrôle inférieure, un appareil peut être ajouté ou retiré du système de contrôle très facilement. Le système de contrôle intelligent englobe les fonctions de surveillance, de contrôle et de gestion des appareils, de sécurité à domicile, de statistiques et d'analyses énergétiques.

Dans [11], les auteurs ont proposé un système domotique télécommandé à faible coût et convivial présenté à l'aide de la carte Arduino, du module Bluetooth, du smartphone, du capteur à ultrasons et du capteur d'humidité. Une application pour smartphone est utilisée dans le système proposé qui permet aux utilisateurs de contrôler jusqu'à 18 appareils, y compris des appareils électroménagers et des capteurs à l'aide de la technologie Bluetooth, le système proposé est un système domotique à usage général. Qui peut facilement être mis en œuvre dans une maison existante, le système suggéré à plus de fonctionnalités que les systèmes domotiques conventionnels tels qu'un capteur à ultrasons est utilisé pour la détection du niveau d'eau et un capteur d'humidité du sol sont utilisés pour un système d'irrigation automatique des plantes. Les auteurs décrivent également l'architecture matérielle et logicielle d'un système, ses travaux futurs et sa portée.

3.4.2 Travaux exploitant l'IoT dans le smart house via le cloud

Dans [56], les auteurs ont proposé un prototype d'une maison intelligente basée sur des systèmes embarqués et internet des objets et des technologies de cloud computing afin de proposer une vie plus intelligente et plus confortable pour les utilisateurs finaux, pour la communication locale entre la passerelle et les différents nœuds dont ils ont utilisé les technologies de communications économiques low-cost et low-power, ZigBee basé sur la norme 802.15.4 et les technologies Bluetooth sont basées sur la norme IEEE 802.15.1, ce prototype implémenté par une carte Raspberry Pi connectée au coordinateur XBee utilisé comme passerelle IoT basée sur Zigbee et deux nœuds. La chambre et la cuisine qui contiennent chacune un Arduino connecté au module routeur XBee. Chaque nœud intelligent dispose d'un ensemble des équipements et des capteurs. La maison intelligente contrôlée par une application de bureau Web basée sur le langage PHP. La plate-forme Cloud utilisée pour enregistrer les actions quotidiennes effectuées par l'utilisateur.

Dans [106], les auteurs ont proposé une approche d'une maison intelligente en intégrant l'Internet des objets (IoT) aux services Web et au cloud computing, cette approche se concentre sur l'intégration de l'intelligence dans les capteurs et les actionneurs utilisant la plate-forme Arduino; mettre en réseau des objets intelligents en utilisant la technologie Zigbee; faciliter les interactions avec les objets intelligentes en utilisant les services Cloud; améliorer l'efficacité de l'échange de données en utilisant le format de données JSON. Cette approche a été mise en œuvre en utilisant la carte Arduino comme carte microcontrôleur pour programmer divers types de capteurs / actionneurs et technologies de communication tels que RFID et ZigBee et basée sur le cloud computing qui fournit des ressources de stockage et de calcul pour implémenter des applications Web.

Dans [42], les auteurs ont proposé la structure de la maison intelligente basée sur le Cloud Computing, qui aide à réduire la charge de travail locale et les utilisateurs obtiennent directement les informations en temps réel via un navigateur Web. De plus, ils construisent la plateforme expérimentale pour valider la structure de la maison intelligente basée sur le Cloud Computing, les résultats expérimentaux montrent que la structure proposée d'une maison intelligente est plus pratique, flexible, à haut rendement et à faible coût. L'architecture proposée implémentée par Raspberry PI et un service Web domestique intelligent basé sur le Cloud Com-

puting pour les utilisateurs sur la plate-forme PaaS fournie par Google. Les données distribuées sont stockées grâce au service d'interface JDO basé sur Google App Engine, et le convertisseur de protocole réalise la transition entre le ZigBee, le Bluetooth et le port série.

Dans [5], les auteurs ont proposé une surveillance et une automatisation de maisons solaires basées sur la plate-forme EmonCMS qui utilise un serveur cloud pour collecter des données à partir de nœuds de capteurs en utilisant le principe IoT. Les données collectées peuvent être affichées, archivées ou traitées et utilisées pour contrôler les appareils dans la maison. Le NodeMCU combiné à l'ESP2866 a été utilisé comme unité de traitement principale qui collecte les données des capteurs, les traite puis les télécharge sur le serveur cloud EmonCMS. Le NodeMCU peut également lire les données et les commandes du même serveur et contrôler les dispositifs de commutation. Il s'agit d'un système complet de surveillance et d'automatisation de la maison intelligente basé sur la technologie IoT.

3.4.3 Travaux sur les architecture des smart houses

Dans [106], les auteurs ont proposé une architecture pour l'intégration de la maison intelligente IoT avec un service web utilisant JSON pour l'échange de données entre les composants du système et la technologie Zigbee pour la mise en réseau. Dans [65], les auteurs ont proposé un système omniprésent de contrôle et de surveillance de la maison à l'aide d'un smartphone.

Dans [113], les auteurs ont proposé une architecture multi-agents qui fournit des caractéristiques autonomes pour l'internet des objets (IoT). Dans [53], les auteurs ont développé un cadre pour la maison intelligente en utilisant un protocole d'application contraint qui fournit une méthode pour contrôler les capteurs et les actionneurs à distance.

Dans [106] [65] [53] comme pour l'architecture utilisée, sa principale (et importante) limitation concerne la structure rigide en niveaux pour la conception du système : elle n'est pas modulaire (ajout ou suppression difficile à faire) et constitue donc un système solide. De plus, il n'y a pas de coopération entre les composants du système. Les objets sont cependant hétérogènes puisqu'il y a différents appareils, les objets ne sont donc pas autonomes.

Dans [65], le système est centralisé et le contrôle est souple, mais le système en général ne l'est pas. Dans [53], nous savons que les agents ont besoin d'un processeur puissant et d'une grande mémoire (RAM), mais la plupart des choses en manquent. Le système multi-agents ne

peut donc pas être intégré dans l'objet. Dans la section suivante, nous voulons présenter une nouvelle conception de l'IdO basée sur la maison intelligente utilisant un système multi-agent.

3.4.4 Travaux sur les architecture des objets

Dans [20], les auteurs ont proposé une structure à quatre couches pour l'IoT; ces couches sont la couche perception, la couche transport, la couche réseau et la couche application. La fonction fondamentale de la couche de détection est de collecter des informations et de fournir une transmission de données au cas où la technologie d'homogénéité utilisée dans le réseau. La couche réseau fournit un environnement combiné pour tous les types de communication. La couche transport fournit une implémentation de QoS (règles de fiabilité et de sécurité). La couche application fournit des opérations à l'utilisateur final et des formes de logiciels de communication. Le logiciel d'application et les fonctions de services sont fournis dans la couche d'application.

Dans [31], une structure à sept couches est proposée; cette structure se compose d'une couche d'environnement qui comprend les objets à suivre qui ne sont généralement pas statiques et d'une couche matérielle qui comprend les composants matériels et les capteurs pour collecter les données environnementales. De plus, la couche information, réseau, communication et service effectue également certaines opérations : stockage et orchestration des services, prise en charge et gestion des applications et couche application.

Dans [125], un IoT cognitif est proposé à travers une topologie de réseau; l'idée est de concevoir une technologie liée au processus de cognition. L'architecture comprend une couche de perception, une couche d'interconnexion de réseau, une couche de fusion d'informations et une couche de service intelligent.

Dans [59], une structure IoT de cinq couches est proposée. La couche de perception se compose d'objets physiques et de dispositifs capteurs. La couche réseau transfère les informations des dispositifs capteurs vers le système de traitement des informations. La couche middleware est responsable de la gestion des services, est liée à la base de données, effectue le traitement de l'information et le calcul omniprésent et prend une décision automatique en fonction des résultats. La couche d'application fournit une gestion globale de l'application basée sur les informations d'objet traitées dans la couche Middleware. Enfin, Business Layer est responsable de

la gestion de l'ensemble du système IoT, y compris les applications et les services (voir le tableau 3.1).

Dans [20] [31] [125] [59], les auteurs s'accordent sur la couche de perception, la couche réseau et la couche application. Cependant, nous avons trouvé une grande différence dans la couche de traitement. Dans [20] [31] [125], comme pour la structure utilisée, sa limitation principale (et importante) concerne la structure rigide en niveaux pour la conception du système. Dans [20] [31], la structure de l'objet n'a pas la propriété d'intelligence, de coopération ou d'organisation.

TABLE 3.1 – Architecture des appareils IoT

	Physical layer	Network	Processing layer	Application Layer
[20]	Perception layer	Network layer Transport layer	/	Application Layer
[31]	Environment layer Hardware layer	Network layer communication layer	Services layer	Application support and management layer Application layer
[125]	Perception Layer	Network Interconnection layer Information Fusion Layer	Intelligent Service Layer	/
[59]	Perception Layer	Network Layer	Middleware Layer	Application Layer

3.4.5 Travaux sur les robots basés internet des objets

Dans [66], Les auteurs ont proposé un robot basé sur la technologie IoT pour améliorer l'efficacité de la récolte de tomates cerise et réduire son taux de casse, ils conçoivent un robot de récolte basé sur la reconnaissance d'images et le contrôle modulaire. Après l'acquisition d'images avec la technologie IOT, le traitement binaire et l'expansion et le traitement de corrosion de l'image originale peuvent augmenter efficacement le taux de reconnaissance des fruits. De plus, l'utilisation de la technologie de commande floue traite l'erreur de réponse du manipulateur.

Les performances du robot de récolte de tomates cerise ont été testées à travers une expérience de récolte. Les résultats expérimentaux ont montré que l'efficacité de la récolte s'améliore considérablement et que le degré de broyage de la tomate cerise diminue considérablement après l'utilisation du robot de récolte de la tomate cerise.

Dans [54], les auteurs mettent en évidence l'utilisation de l'IoT pour le développement d'un système destiné à la localisation de robots mobiles utilisant des réseaux de neurones convolutifs (CNN) dans le processus d'extraction des fonctionnalités des images, selon le concept de transfert d'apprentissage. Le mécanisme utilise la méthode de cartographie topologique pour s'orienter dans l'environnement d'exploration considéré. L'efficacité de l'approche est démontrée par des paramètres tels que la précision, le score F1 et le temps de traitement. Le système IoT confère la centralisation du traitement, réduisant les coûts et permettant la réutilisation de la puissance de calcul inactive du robot. Combiné à cet avantage, CNN atteint toujours une précision de 100% et un score F1, ce qui s'avère être une technique efficace pour l'activité requise. Avec cela, l'approche proposée se révèle efficace pour une utilisation dans la tâche de localisation de robots mobiles.

Dans [74], les auteurs présentent une alternative à faible coût pour un robot explorateur mobile qui utilise du matériel et des logiciels ouverts, dont la conception est destinée à incorporer des caméras, des capteurs et d'autres périphériques pour pouvoir inspecter son environnement en plus d'être contrôlé à distance via Internet technologies des choses. Pour le premier prototype, une caméra et un capteur de température / air-humidité sont ajoutés au robot mobile. Grâce à une interface utilisateur graphique (application web), la vidéo est visualisée avec les mesures environnementales de l'endroit où le robot mobile fonctionne et permet de le contrôler via Internet.

3.5 Conclusion

Dans ce chapitre, nous avons donné un aperçu sur l'évolution des constructions des maisons jusqu'à la naissance du terme maison intelligente (Smart home), ensuite nous avons défini ce qui est une maison intelligente, En plus, nous avons présenté un état de l'art sur les approches utilisant la technologie de l'Internet des objets dans un smart house.

Dans le prochain chapitre nous allons présenter notre propre modélisation relative à la nouvelle approche utilisant la technologie de l'Internet des objets dans un smart house.

Chapitre 4

Modélisation de la nouvelle approche

4.1 Introduction

L'internet des objets étend la technologie Internet actuelle en connectant différents types d'objets (objets ou appareils) entre eux et leur permet de communiquer intelligemment. ce concept est conçu pour connecter des millions d'objets, mais les objets avec ce nombre nécessitent de grands espaces de stockage et génèrent un trafic important, ce qui pourrait créer de nombreux problèmes au réseau. De plus, bien que les objets soient connectés les uns aux autres, ils ne sont pas nécessairement capable de communiquer entre eux. Leur capacité à communiquer entre eux dépend de la similitude du service auquel ils sont affectés. Ces lacunes sont dues au fait que les objets ne peuvent pas raisonner dans leur environnement et prendre ensuite des décisions et des actions intelligentes pour atteindre leurs objectifs.

Bien que l'ensemble du système connecté de l'loT manifeste la capacité de prendre des décisions et d'interagir intelligemment entre les objets, il le fait sur la base des règles implicites sans prendre en considération des modifications involontaires de l'environnement.

Tout le système peut quelque peu être représenté comme un système intelligent mais pas les objets. Ce sont les enjeux majeurs de l'loT en plus des enjeux de sécurité, de gouvernance et de normalisation. Nous rapploons queles lacunes de l'internet des objets sont le suivant :

- L'objet comme une seule entité dans l'loT n'est pas intelligent, mais tout le système est quelque peu intelligent.
- La prise de décision et les actions ultérieures dépendent du service.

- Les objets de l’IoT sont connectés les uns aux autres, mais ils ne sont pas nécessairement capables de communiquer entre eux.
- Les objets ne peuvent communiquer avec d’autres objets que lorsqu’elles sont conçues pour exécuter les mêmes services système.
- L’IoT souffre d’autres problèmes concernant la sécurité, la gouvernance et la normalisation.

La vision du concept IoT est d’établir un système à grande échelle en connectant tout à l’Internet et en les faisant communiquer entre eux de machine à machine (M2M), L’IoT tente d’aider les humains à décider de meilleures solutions de gestion des données pour leurs problèmes spécifiques au domaine en traitant les informations pertinentes à partir des objets interconnectées. Cependant, en raison des lacunes ci-dessus qui empêchent cette vision de devenir une réalité, nous proposons une nouvelle approche basée sur des agents logiciels pour atténuer les problèmes de l’absence de raisonnement et d’intelligence dans les objets des systèmes IoT, En ayant de telles fonctions améliorées, notre nouvelle approche tente d’augmenter l’IoT en incorporant des agents logiciels intelligents dans les objets pour améliorer les services qui pourraient être dérivés des interactions des agents logiciels dans ces objets.

4.2 Architecture générale du système

Dans le domaine de l’IoT, les dispositifs dits IoT fournissent normalement de telles capacités de calcul et sont appelés edge nœuds, edge périphériques ou périphériques intelligents; même si, selon la granularité du scénario particulier, les périphériques peuvent avoir une capacité de calcul plus ou moins élevée. Le fait d’avoir une capacités de calcul dans le périphérique lui-même offre plusieurs avantages : le premier est que nous pouvons prétraiter les données obtenues dans l’edge périphérique et communiquer ces informations aux périphériques externes uniquement lorsque cela est pertinent ou nécessaire, économisant de nombreuses ressources du périphérique et améliorant la latence; le second est que l’appareil peut prendre sa propre décision, s’il est programmé pour lui, particulièrement utile lorsqu’il n’y a pas de connexion réseau.

Dans cette section, nous allons présenter l’architecture générale "IoT approach using multi-

agent system for ambient intelligence" [132] proposée de notre système. En effet, le système proposée est basé sur un système multi-agents (situé et mobile), en gardant l'avantage offert par ce paradigme et afin de maintenir le problème de communication entre les objets (IoT). Nous avons doté les objets de notre système avec des agents intelligents pour améliorer les services qui pourraient être dérivés des interactions des agents logiciels dans ces objets.

L'architecture générale de notre système est une architecture collaborative pour l'IoT, qui est composée de plusieurs edge et fog nœuds et de cloud qui collaborent dans un écosystème orienté services et agents tout au long de la coopération des agents logiciels et du CEP.

Nous avons trois couches dans l'architecture, ces couches se concentreront conceptuellement sur la détection, la passerelle et la couche de service et seront matérialisées dans les edges nœuds, de fog et de cloud correspondants. La figure 4.1 montre une architecture illustrative dans laquelle un nœud de cloud, deux fog nœuds et cinq edge nœuds ont été inclus. Cependant, l'architecture pourrait avoir des nœuds supplémentaires de tous les types (certains fog et edge nœuds supplémentaires sont également représentés en petit). Dans tous les cas, on s'attend à ce que l'architecture ait plus de edge nœuds que de fog nœuds et donc plus d'edge nœuds que de cloud nœuds.

4.2.1 Description des couches et des composants

Comme mentionné précédemment, dans notre architecture, nous avons des edge nœuds, fog nœuds et des nœuds de cloud, comme expliqué dans les paragraphes suivants.

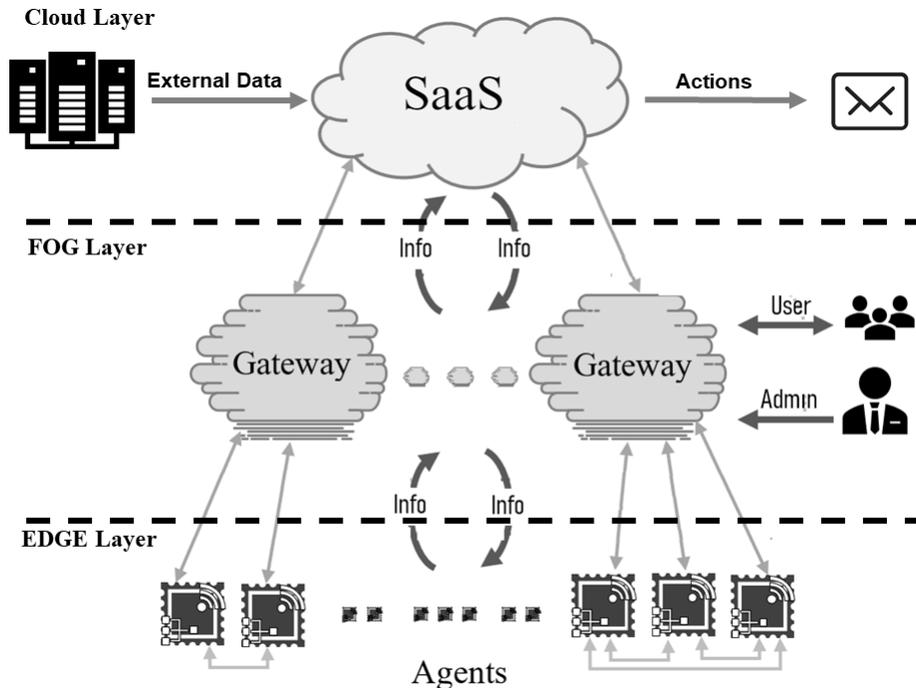


FIGURE 4.1 – Architecture en couches du système

4.2.1.1 Edge nodes

Il faut se rappeler que les IoT devices, représentés comme des edge nœuds, ont généralement des ressources de calcul limitées, c'est pourquoi (1) les logiciels qu'ils incorporent doivent consommer peu de ressources en exécution et (2) les communications via Internet doivent être autant limitées que possible pour économiser les ressources. Avec cet objectif, nous avons déployé un agent logiciel dans chaque edge nœud, car les agents consomment peu de ressources et disposent d'une autonomie et d'une capacité d'intelligence et de décision suffisantes pour filtrer les informations réellement nécessaires à transmettre et celles qui ne seront utilisées qu'en interne dans le nœud. L'intelligence de l'agent est programmée à partir d'une série de règles de comportement; de cette manière, le comportement du dispositif pourra s'adapter à l'environnement prenant des décisions en fonction des informations obtenues de ses capteurs et des informations obtenues des autres edge nœuds connectés au réseau et du fog nœud. Les agents pourront communiquer entre eux et traiter et analyser les informations détectées et reçues des autres acteurs et prendre les décisions correspondantes sans avoir besoin de contacter l'utilisa-

teur.

4.2.1.2 Fog nodes

Ce sont les appareils qui serviront de passerelles entre les appareils IoT en edge et (a) l'utilisateur et (b) les nœuds dans le cloud. D'une part, toutes les communications des agents logiciels avec l'utilisateur final se feront via un Broker dans la passerelle : les edge devices peuvent envoyer des messages à un topic dans le Broker auquel l'utilisateur est abonné. D'autre part, les communications entre l'edge et le cloud se feront également via le Broker de manière bidirectionnelle. Un tel Broker sera connecté à un moteur CEP dans l'edge nœud pour traiter les données reçues et détecter les informations pertinentes pour l'autre partie. Autrement dit, (1) le cloud envoie des informations au fog, le moteur CEP traite ces informations et si un modèle est satisfait, il soumet les informations pertinentes au edge; (2) l'edge envoie des informations au fog, le moteur CEP traite ces informations et si un modèle est respecté, il soumet les informations pertinentes au cloud. Bien sûr, un modèle peut être rencontré à partir des informations reçues à la fois de l'edge et du cloud. Les utilisateurs et le cloud auront différents topic auxquels s'abonner.

4.2.1.3 Cloud Node

Le nœud cloud a une communication bidirectionnelle avec les fog nœuds. D'une part, le nœud cloud s'abonne aux topic de message des fog nœuds, ce qui lui permet d'obtenir des informations de tous ces nœuds et de prendre des décisions de plus haut niveau. Ces informations sont acquises avec d'autres sources de données hétérogènes et traitées via une SOA 2.0 compatible CEP. Le moteur CEP peut détecter des situations d'intérêt soit pour des tiers, soit pour des fog nœuds et les soumettre au Broker de messages correspondants auxquels ces acteurs pourraient s'abonner. Il est important de se rappeler qu'il pourrait y avoir plus de nœuds dans le cloud qui pourraient en même temps communiquer avec d'autres nœuds dans le cloud, mais pour plus de simplicité, nous n'allons en représenter qu'un.

Pour mettre en place le système, tout d'abord, les edges dispositifs seront programmés avec les agents logiciels et les règles correspondantes avec les actions qu'ils peuvent effectuer en fonction des informations reçues des capteurs, des autres agents, de la passerelle et de l'utili-

sateur. Ces règles comprendront non seulement des actions telles que la mise en marche du climatiseur, mais également les informations susceptibles d'être diffusées aux agents restants ainsi que les informations à soumettre à l'utilisateur ou au Broker de la passerelle à traiter dans le moteur CEP. Deuxièmement, nous déploierons et configurerons le Broker de messages et le moteur CEP dans le fog nœud, qui recevront les données de l'edge, d'autres fog nœuds et de cloud, comme expliqué précédemment. A cet effet, les acteurs seront abonnés aux topics correspondants. De plus, nous déploierons les modèles conçus dans le moteur CEP. Enfin, nous déploierons également un Broker de messages et SOA 2.0, avec un moteur CEP intégré, dans le cloud. Nous nous abonnerons également au Broker de messages aux sources de données d'intérêt, ainsi qu'aux fogs nœuds correspondant au scénario en question, et nous déploierons les patterns conçus. Nous déploierons également d'autres actions souhaitées lorsqu'un modèle est respecté (soumission de notification, offre de service, etc.).

4.3 Architecture détaillée de système

Dans cette section nous allons présenter les détails de notre conception.

4.3.1 IoT Gateway

Le dispositif IoT Gateway est un moyen de normaliser les informations provenant de l'environnement ambiant pour la perception des agents. Cet étalonnage est nécessaire en raison de l'hétérogénéité des protocoles des différents fabricants. Ainsi, le MAS (système multi-agent) perçoit l'environnement ambiant via la passerelle et agit sur celui-ci sans se soucier du format des données perçues. Cette gateway est composée de deux couches [129].

Java Agent DEvelopment Framework : JADE est un Framework de développement de systèmes multi-agents, open-source et basé sur le langage Java. Ce dernier assure une communication transparente par l'échange de messages dans le langage normalisé FIPA-ACL, la plateforme JADE a été ajoutée pour faciliter la communication entre les agents;

MQTT Broker : Nous avons doté l'IoT Gateway d'un MQTT Broker pour faciliter le processus de communication entre l'utilisateur et les objets tout en réduisant le trafic du réseau et d'économiser l'énergie du dispositif intermédiaire (IoT Gateway).

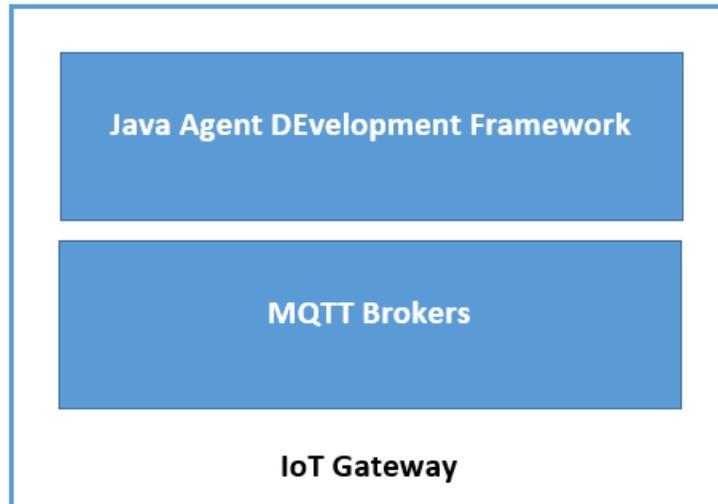


FIGURE 4.2 – Architecture proposée IoT Gateway

4.3.2 Agent de sécurité

L'architecture de cet agent illustré dans la figure 4.3 est composée de cinq éléments qui sont [132] :

Environnement (Environment) : il est constitué par l'environnement réel (maison, appartement, lieu de travail), il contient les composants de l'utilisateur et les composants voisins (l'autre composant situé au même endroit).

Capteurs (Sensors) : c'est un appareil utilisé pour transformer le statut d'une grandeur physique observée (environnement) en une quantité utilisable (mesures). Tels qu'une tension électrique, une hauteur de mercure, une intensité ou la déviation d'une aiguille. Il y a souvent une confusion entre capteur et transducteur : un capteur est au moins composé d'un transducteur.

Base historique des états (Historical DB) : c'est un outil pour stocker l'historique d'informations fourni par les capteurs (l'état de l'environnement) et les états des autres composants du système.

Base de règles (Rules DB) : elle rassemble la connaissance de l'agent. elle contient des règles pour aider l'agent à prendre des décisions. De telles décisions changent son état en fonction du état actuel de l'environnement et des états d'autres agents. La base de règles est mise à jour en fonction des nouveaux statuts et besoins de l'environnement. Les règles représentent la connaissance générale de la sécurité et les constatations factuelles.

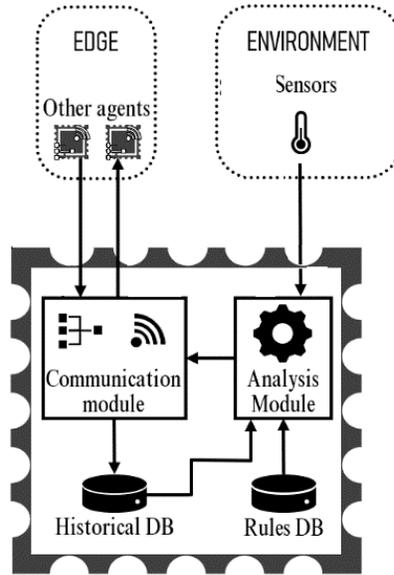


FIGURE 4.3 – Architecture concrète de l'agent de sécurité

4.3.2.1 Diagramme d'état

Le diagramme d'état de la figure 4.4 représente les états de la sécurité de l'agent et les conditions permettant de passer d'un état à d'un autre. L'enchaînement entre les différents états est décrit de la manière suivante :

L'agent commence à percevoir l'environnement et attend les messages des autres agents. S'il y a un message, l'agent passe à l'état d'analyse des informations. Il compare les règles de sécurité avec les informations fournies par les agents et les données de détection. S'il existe une règle vraie (la règle est réalisable), l'agent envoie des ordres aux autres agents afin de contrôler le problème. L'agent informe l'utilisateur et le service qui peut gérer cette situation.

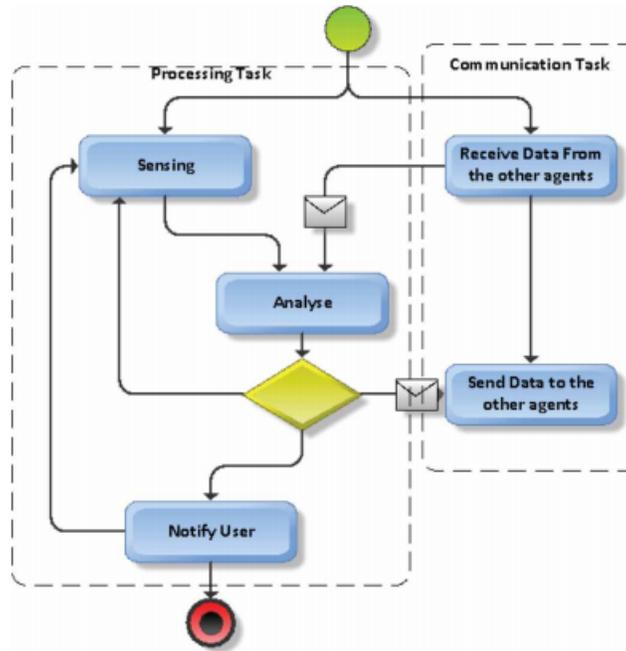


FIGURE 4.4 – Diagramme d'état de l'agent de sécurité

4.3.3 Dispositif IoT

Comme expliqué précédemment, l'architecture IoT actuelle est trop rigide (voir Figure 1.1 dans le chapitre 1) et ses objets ne sont pas autonomes. De plus, il n'est pas cognitif pour la construction de dispositifs IoT intelligents et autonomes.

Pour le dispositif IoT, nous avons proposé une extension "Ambiance Intelligence Approach Using IoT and Multi-Agent System" [129] en quatre couches. En particulier, nous avons ajouté une couche d'agent dans l'architecture, telle que présentée dans la figure 4.5. Nous avons intégré cette couche pour garantir les caractéristiques d'autonomie et d'intelligence de l'architecture IoT de cette manière, les objets conservent les caractéristiques d'autonomie et d'intelligence de l'agent.

Comme on peut le voir à la figure 4.5, nous avons inclus la couche d'agent entre la couche application et la couche réseau. La raison en est que l'agent proposé est un logiciel et qu'il doit utiliser la couche Network and Data Communication (NDC) pour : échanger des informations avec les appareils se trouvant à proximité; les informations reçues d'autres dispositifs et les informations captées par les capteurs sont les données sur lesquelles l'agent base ses décisions.

L'agent utilise la couche physique (couche réseau et couche de perception) en tant que capteurs pour détecter l'environnement et traiter ces informations réelles. De cette façon, le comportement de l'agent change en fonction de ces informations. En conséquence, cela modifie un ensemble de décisions utilisées dans la couche logicielle en fonction du but qui lui est destiné.

Le dispositif IoT c'est l'élément central de cette section. Nous y avons ajouté la couche d'agent afin de doter le périphérique d'une sorte d'auto-intelligence, De cette manière, l'appareil est capable d'adapter et de mettre à jour le statut de l'environnement, grâce aux informations obtenues par les capteurs. L'agent traite et analyse ces informations. En conséquence, il prend la décision appropriée sans contacter l'utilisateur [129];

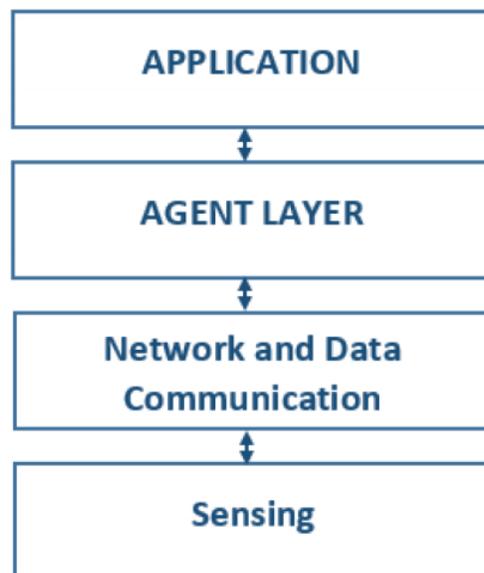


FIGURE 4.5 – Architecture IoT Device

4.3.4 Agent de santé

L'agent de santé est intégré dans un composant physique, il peut contrôler l'état de santé de l'utilisateur de manière autonome et continue à travers les informations fournies par son smartphone, fit-bit montre, les autres agents...

Le smartphone fournit des informations telles que le nombre de pas que l'utilisateur fait, sa vitesse, les endroits qu'il a visités et les chemins qu'il a empruntés. En programmant l'alarme, le pourcentage d'éclairage et l'utilisation de l'appareil, nous pouvons connaître l'heure de som-

meil de l'utilisateur, l'heure de réveil et la durée du sommeil, Nous pouvons confirmer ces informations à travers les informations que nous recevons d'autres agents telles que le temps d'éteindre l'éclairage et le mouvement de l'utilisateur à la maison, qui sont des informations importantes pour connaître la nature de sa santé.

Grâce à la liste d'achats qui nous est fournie en utilisant une carte de crédit, nous pouvons connaître la qualité et la nature des aliments qu'il mange.

Grâce à la communication de l'utilisateur avec les autres à travers les médias sociaux, par téléphone, à la maison et à travers l'histoire de la recherche sur Internet, nous pouvons extraire son état psychologique à travers l'analyse des informations précédentes et le développement de ses vélos vocaux pendant la conversation. L'agent analyse toutes les informations précédentes et à travers lui il peut extraire la situation approximative de la santé de l'utilisateur, l'agent fournit des conseils sur une base régulière et continue à l'utilisateur afin de maintenir sa santé.

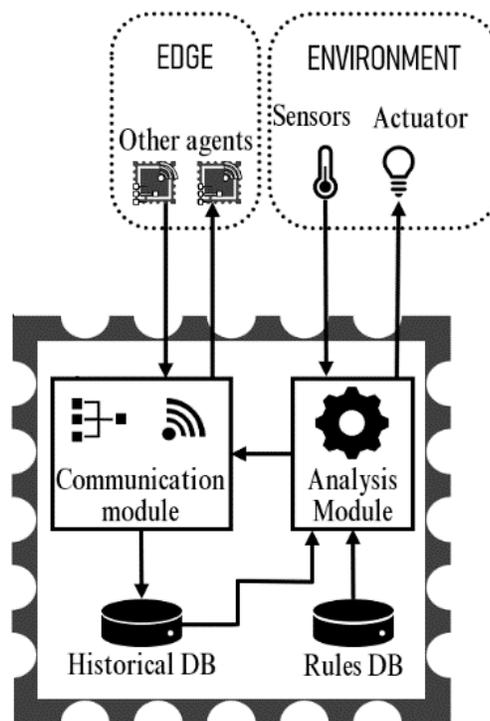


FIGURE 4.6 – Architecture concrète de l'agent «Health Care»

— **Environnement** : qui est perçu et contrôlé par l'agent de soins de santé qui est l'utilisa-

teur du système. L'ensemble des perceptions est l'état actuel de la santé et des informations physiques.

- **Historique des états** : c'est une base de données qui contient d'anciennes informations sur la santé de l'utilisateur. Ses informations sont utilisées pour prédire ce qui se passerait et soyez prudent avant de tomber dans un état critique.
- **Base de Règles** : est composée d'un ensemble de règles qui représentent des observations factuelles et des connaissances générales sur le domaine médical.
- **Module d'action** : permet aux agents de santé de mener un raisonnement logique et de tirer des conclusions à partir l'historique des états et d'une base de règles.

4.3.5 La modélisation de l'IoT Robot

L'Internet des objets[6][97] est un réseau d'objets principalement supportés par des dispositifs électroniques et des composants électroniques tels que des capteurs et des cartes électroniques. Ces objets peuvent être des périphériques physiques et virtuels, des capteurs ou des actionneurs. L'environnement intelligent nécessite beaucoup de capteurs et à différents endroits. Le coût de ces capteurs est cher. Pour réduire le coût de ces capteurs, nous avons proposé un robot mobile comportant une gamme de capteurs différents qui vont détecter l'environnement et envoyer des informations et des données aux objets de

l'environnement[118][98]. À l'aide de ce robot, On n'aura pas besoin de mettre de capteurs partout puisque le Robot qui doté de capteurs se déplace aux endroits désirés. La communication entre le robot et les objets IoT dans l'environnement se fait via le protocole MQTT[82]

Les objets IoT peuvent contrôler le robot directement via les commandes reçues par le robot.

Nous avons implémenté ce robot avec une carte électronique perfectionnée Raspberry Pi B dotée de deux capteurs et d'une carte Arduino UNO pour la gestion des mouvements de robot [130]. Il est également équipé de capteurs pour empêcher le robot d'entrer en collision avec des barrières et des obstacles. Ces capteurs aident également le robot à surmonter les obstacles de manière autonome et aident à tracer un contour partiel de l'environnement environnant lorsque le schéma est un cercle de 4 mètres de diamètre et la commande du robot effectuée via un smartphone. Les commandes envoyées via le réseau local au robot si le smartphone et le

robot sont connectés dans la même passerelle. Si le robot est distant, il reçoit les commandes via le réseau public Internet.

Nous avons fourni au robot une caméra orientable pour visualiser l'environnement. Il est contrôlé à distance par les commandes envoyées de l'extérieurs. Le serveur de caméra (Motion) envoie les images capturées à l'utilisateur en temps réel et directement. Motion est un système de surveillance assez complet. Il est extrêmement personnalisable : détection de mouvement, enregistrement image par image, enregistrement vidéo, laps de temps.

4.3.5.1 Description de IoT Robot

IoT Robot est composé principalement de 03 modules dont chacun contient des composants différents [130].

Le module de déplacement : est le module responsable du mouvement du robot et de son déplacement. Ce module transforme l'énergie électrique en énergie cinétique à travers les moteurs du robot.

Le module de détection : est le module responsable de la détection du support externe. Cette unité convertit les valeurs physiques de l'environnement en valeurs numériques pouvant être stockées et analysées. Ce module relie le monde réel au monde numérique.

Le module de communication : est le module responsable de l'interaction du robot avec l'utilisateur en recevant des commandes, des rapports et des images sur l'environnement.

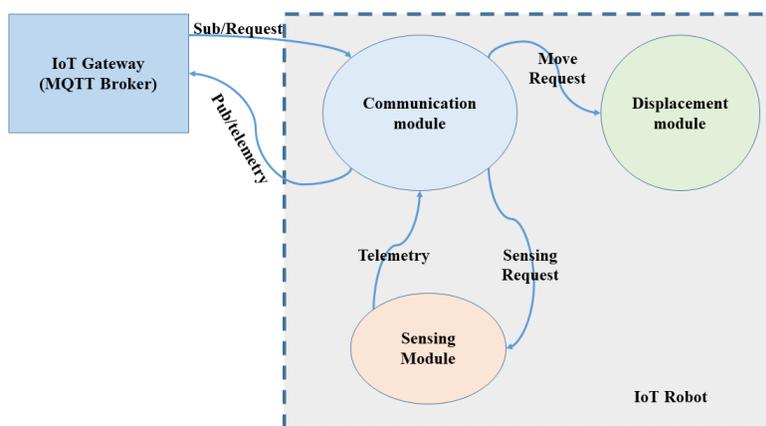


FIGURE 4.7 – Modules d'IoT Robot

4.3.5.2 Diagramme d'état

Ce diagramme montre comment les pièces du robot interagissent les unes avec les autres pour atteindre leurs objectifs de manière idéale.

- Le module de communication reçoit des commandes de l'utilisateur, ces éléments doivent être l'état initial.
- Le module de communication dirigera la commande en fonction de son type. S'il s'agit d'un ordre de commande de déplacement ou de détection.
- S'il s'agit d'un mouvement, le module de déplacement l'exécutera directement.
- Le module de détection détecte l'environnement à travers les capteurs, prend des photos de l'environnement et les envoie directement et en temps réel à l'utilisateur.

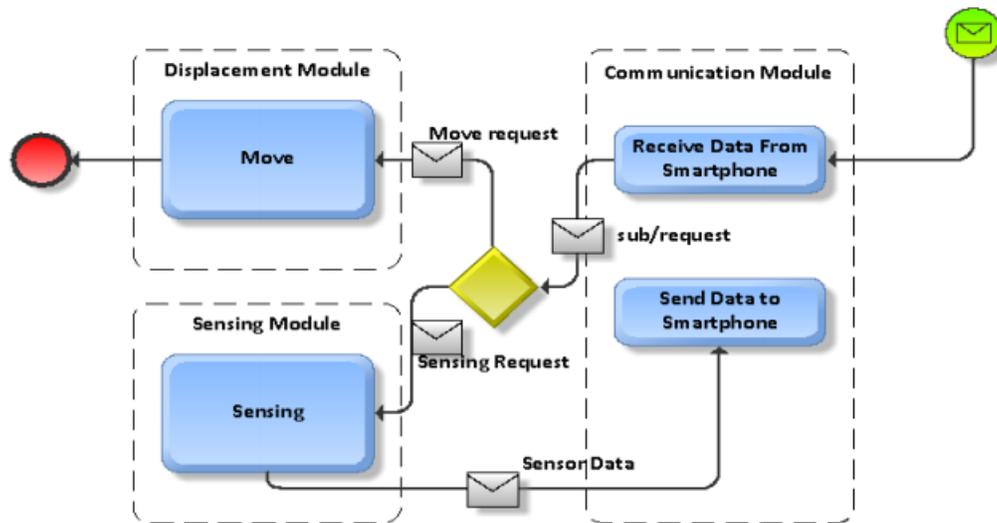


FIGURE 4.8 – Diagramme d'état de IoT Robot

4.3.6 Simulateur de maison intelligente

Une maison intelligente consiste à connecter les différents appareils et systèmes de la maison afin qu'ils puissent être contrôlés de n'importe où et provoquer l'interaction souhaitée entre eux. Ces appareils sont des objets connectés via un réseau qui a une fenêtre sur Internet. Ce réseau d'objets s'appelle l'Internet des objets (IoT). L'IoT peut s'appliquer à plusieurs domaines : les villes intelligentes (villes totalement ou partiellement connectées à Internet leur permettant d'optimiser leurs capacités comme la gestion du trafic et le traitement de l'eau), la santé, les

wearables (toutes les technologies portables telles que les montres connectées et les localisateurs), transport, mais aussi les lieux de travail, la production et les maisons [131].

4.3.6.1 Conception de l'API Simulateur

Notre simulateur est conçu en deux parties; une partie invisible à l'utilisateur (c'est-à-dire ne peut pas voir le contenu) contient les deux couches (API du simulateur, interface graphique du simulateur), et la deuxième partie visible à travers le code qui est classe publique.

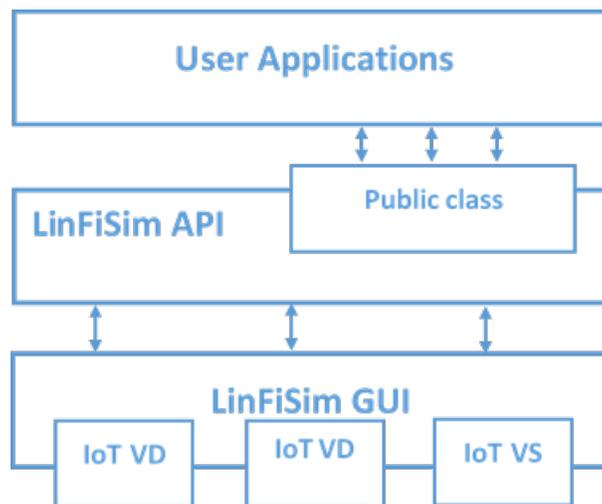


FIGURE 4.9 – Architecture de l'API du simulateur

4.3.6.1.1 Composantes de l'API Dans cette section, nous expliquons chaque composante de notre API, son contenu et son mode de fonctionnement.

4.3.6.1.1.1 Interface utilisateur graphique (GUI) du simulateur : l'interface utilisateur graphique (GUI) du simulateur est l'environnement graphique que l'utilisateur peut voir à l'écran et que l'interface contient.

Périphérique virtuel IoT : le périphérique virtuel Internet des objets (IoT VD) est un objet virtuel qui simule un périphérique réel tel que (TV, climatiseur et lumière ...), chaque IoT VD a trois statuts possibles : ON, OFF, SUSPENDU, dans l'environnement du simulateur.

Pour tester notre API, Nous nous sommes concentrés sur les appareils les plus importants utile dans chaque maison, le tableau suivant illustre les appareils, leurs états et ses caractéristiques :

TABLE 4.1 – Périphérique virtuel IoT

IoT VD	États	Caractéristique
lampes	ON, OFF, SUSPENDU	/
Réfrigérateur	ON, OFF, SUSPENDU	/
cuisinière	ON, OFF, SUSPENDU	/
télévision	ON, OFF, SUSPENDU	/
climatisation	ON, OFF, SUSPENDU	température et humidité

Capteur virtuel IoT : Le capteur virtuel Internet des objets (IoT VS) est un capteur qui simule un capteur réel tel que (température, humidité, gaz, présence), chaque IoT VS a deux méthodes importantes, la méthode Set est un moyen de modifier le contenu (valeur) de capteurs, la méthode Get est un moyen de récupérer le contenu (valeur) des capteurs.

4.3.6.1.1.2 interface de programmation d'application (API) : API de simulateur, c'est un ensemble de classes Java capable de gérer par l'utilisateur de l'API (interface de programmation d'application) à partir des appels des méthodes publiques trouvées dans la classe publique.

Classe publique : Cette classe est la fenêtre utilisateur du simulateur, sans obliger l'utilisateur à connaître le code source de l'interface de programmation d'application (API), elle possède un ensemble de méthodes et d'attributs qui peuvent être utilisés par l'utilisateur pour gérer l'environnement de simulation.

4.3.6.1.1.3 User Application Une application que l'utilisateur programme à l'aide de l'interface. Dans lequel la programmation se fait en utilisant des briques de fonctionnalités fournies par l'API. Cette construction d'assemblage nécessite que le programmeur sache comment interagir avec le simulateur, qui dépend de son interface utilisateur de programmation. Une ap-

plication utilisateur peut être un ensemble de couches, par exemple en ajoutant une couche réseau ou un service Web.

4.4 Communications des composants de notre architecture

Dans cette section, nous donnerons plus de détails sur l'architecture interne de chaque couche et expliquerons toutes les communications possibles qui peuvent se produire entre ses composants. N'oubliez pas que l'edge ne communiquera pas directement avec le cloud mais via le fog nœud. Toutes ces communications sont effectuées via un Broker de messagerie léger via le protocole MQTT.

4.4.1 Communication des edge nœuds

Chaque edge nœud sera piloté par un agent. L'agent accédera aux informations captées par le capteur en question et / ou activera / désactivera l'actionneur en question. Il disposera d'un ensemble de règles et d'un accès aux données historiques pour détecter les informations d'intérêt à soumettre à d'autres agents et / ou au fog nœud. De plus, il communiquera avec ce dernier grâce au module de communication. Toutes les communications sont prises en charge par la plate-forme JADE, que nous avons utilisée pour implémenter notre logiciel orienté agent.

Par conséquent, plusieurs communications peuvent être effectuées à partir d'un edge device, comme illustré à la figure 4.10 :

1. Tout d'abord, (1a) les capteurs dans le dispositif détecteront les informations de l'environnement et (1b) le module d'analyse soumettra les informations détectées au module de communication dans l'edge dispositif.
2. Deuxièmement, lorsque cela est nécessaire, selon les règles programmées dans les agents, nous transmettrons les informations détectées aux edge devices intéressés restants dans le même réseau.
3. De la même manière, l'appareil recevra des informations d'autres appareils du réseau.
4. En outre, en fonction des règles programmées dans l'agent et des décisions prises, l'edge dispositif (4a) peut soumettre des informations au courtier de messages dans la passe-

relle et / ou (4b) activer un actionneur.

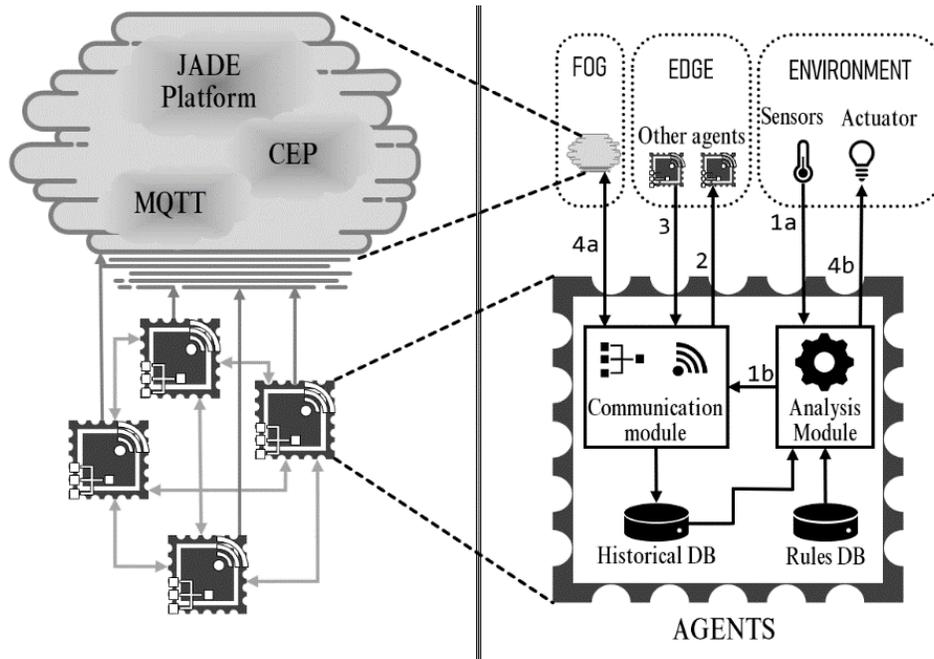


FIGURE 4.10 – Edges nœuds et de leur communication avec le fog

4.4.2 Perspicacité et Communications sur les edge nœuds

Dans les edge nœuds, nous allons avoir, d'une part, la plate-forme JADE qui prend en charge les agents répartis entre les edge nœuds; et d'autre part, un moteur CEP. Puisque tous les messages reçus dans le fog sont formatés dans notre propre architecture, ils seront déjà dans un format homogène, ainsi nous éviterons le besoin d'un SOA 2.0 et d'un ESB et n'aurons besoin que du moteur CEP. Nous aurons également un courtier de messages pour offrir des revenus et des topics de message de sortie au fog nœud. Les événements pertinents détectés par les edge nœuds seront envoyés à la rubrique d'entrée; dans ce même sujet, les événements d'intérêt envoyés à partir du cloud ou d'autres fog nœuds seront reçus. Toutes les données reçues dans cette rubrique seront traitées par le moteur CEP qui téléchargera les nouvelles situations d'intérêt détectées dans le cloud, d'autres edge et fog nœuds, comme programmé dans le modèle via une nouvelle rubrique de sortie. De plus, l'utilisateur peut envoyer de nouvelles conditions aux edge nœuds via une rubrique et recevoir des informations d'intérêt par les mêmes moyens. Par conséquent, les communications de l'edge nœud, la passerelle.

Ainsi, le comportement est le suivant :

1. Le fog nœud recevra les informations suivantes :
 - Le fog nœud recevra les informations pertinentes soumises par les agents en edge via un topic de message.
 - En outre, le fog nœud recevra des informations pertinentes à traiter à partir du cloud. Ces informations seront reçues via le même topic de message.
 - Enfin, le fog nœud peut recevoir des informations pertinentes à traiter à partir d'autres fog nœuds , qui seront également reçues via le topic du message.
2. Lorsque les informations parviennent au Broker , elles sont envoyées au moteur CEP.
3. Les informations étant traitées en temps réel dans le moteur CEP, diverses situations d'intérêt peuvent être détectées. Ces situations d'intérêt peuvent être envoyées à trois sorties.
 - Situation pertinente d'intérêt pour les edge nœuds.
 - Situation pertinente d'intérêt pour les nœuds cloud.
 - Situation pertinente intéressant les autres fog nœuds.
4. Enfin, il y aura une communication directe avec l'utilisateur, qui ne passe pas par le moteur CEP :
 - Les agents de l'edge soumettront les informations pertinentes à l'utilisateur, selon les règles programmées en eux; ces informations sont soumises via un topic de message.
 - L'utilisateur sera abonné à ce topic et recevra donc ces informations par son intermédiaire.
 - De même, l'utilisateur soumettra de nouvelles conditions, règles ou actions à l'edge via le topic du message.
 - Et les edges nœuds recevront ces informations via leur abonnement au topic.

4.5 Modèle de coopération utilisé

Nous avons mis en œuvre notre architecture pour créer un périphérique de sécurité IoT. Ceci, enfin, contrôle la sécurité d'un environnement contre les événements accidentels et accidentels. Son but est d'agir sur l'existence d'un criminel dans l'environnement. Et il peut contrô-

ler les risques d'accident tels que les incendies et les fuites de gaz. La mise en œuvre est illustrée à la figure 4.11 et expliquée ci-dessous.

Nous mettons en œuvre notre architecture proposée avec la carte Raspberry pi 2 (voir la partie gauche de la figure 4.11). Cette carte nous a permis de combiner les trois couches suivantes en une seule et même structure (comme illustré à droite de la figure 4.11) :

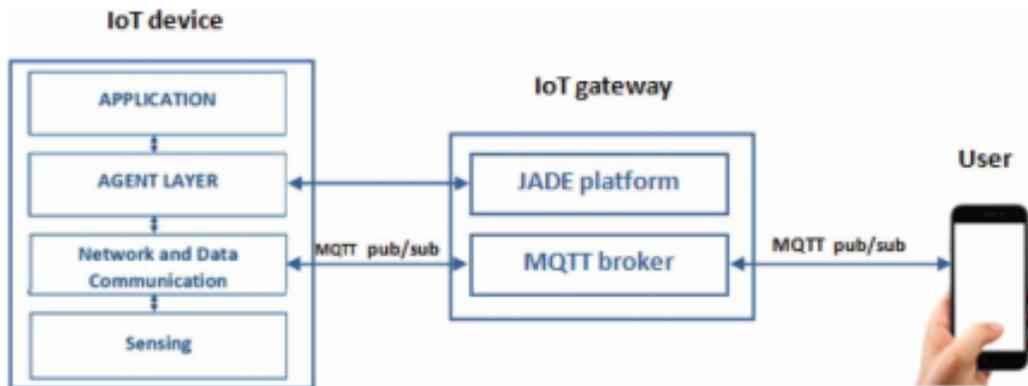


FIGURE 4.11 – Architecture proposée pour le fonctionnement global

La couche d'application (The Application Layer) : c'est une application d'un environnement de sécurité réel (maison, appartement, lieu de travail) ;

La couche d'agent (The Agent Layer) : il s'agit d'un agent intégré dans l'instrument IoT, il gère les informations reçues par les capteurs, les analyse et prend des décisions.

Couche de communication réseau et données (Network and Data Communication Layer) : le périphérique IoT communique avec les autres périphériques via cette couche ;

Couche de détection (Sensing Layer) : elle a été mise en œuvre sur le capteur de présence et le capteur d'humidité et de température.

4.5.1 Communication entre les objets

Afin de visualiser la communication inter objets , nous avons utiliser deux diagrammes de séquence de deux scénarios.

4.5.1.1 Scénario de détection des voisinages :

pour un système dynamique et flexible, il est possible d'ajouter de nouveaux objets sans aucun problème. La figure 4.12 illustre un diagramme de séquence décrivant comment les objets détectent leurs voisins.

Dans le diagramme, l'objet du climatiseur diffuse le message "bonjour" au démarrage à tous les objets de l'environnement. Chaque objet reçoit le message, enregistre les informations du climatiseur en tant que nouvel objet et répond avec un autre message contenant son information. Avec cette méthode, le système a découvert les nouveaux objets de l'environnement.

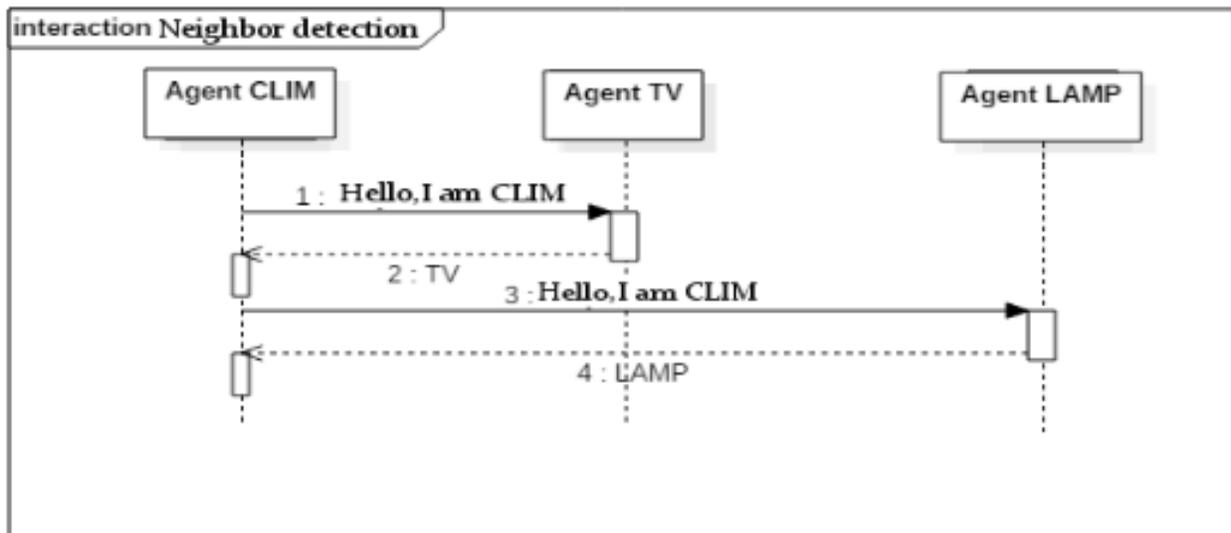


FIGURE 4.12 – Diagramme de séquence de détection des voisins

4.5.1.2 Scénario de détection de défaut :

pour créer un système complètement autonome et éviter le problème de la centralisation. Car dans ce cas, si le composant central tombe en panne, tout le système y tombe en panne, pour cela on a proposé un système décentralisé et les objets échangent les informations entre eux et le système détecte les défaillances.

Chaque objet stocke des informations provenant d'objets voisins. Les objets diffusent toutes les 30 secondes leur état et enregistrent à l'aide de leurs voisins. Si un objet dépasse le délai de

30 secondes sans l'état de diffusion et que l'objet ne déclare pas que le système est éteint, il est considéré comme un échec.

Le système pris en charge demande au serveur de dépannage de réparer celui qui a échoué ou a appelé l'agence de dépannage pour réparer dans ce cas. Dans l'environnement, qui est responsable de l'information du serveur? Le système est centralisé n'est pas un composant responsable pour informer le serveur.

Résoudre ce problème en important la technique CSMA / CD [75] utilisée dans les réseaux pour éviter les collisions. Dans notre cas, l'utilise pour sélectionner un objet qui joue le rôle d'informer le serveur de dépannage.

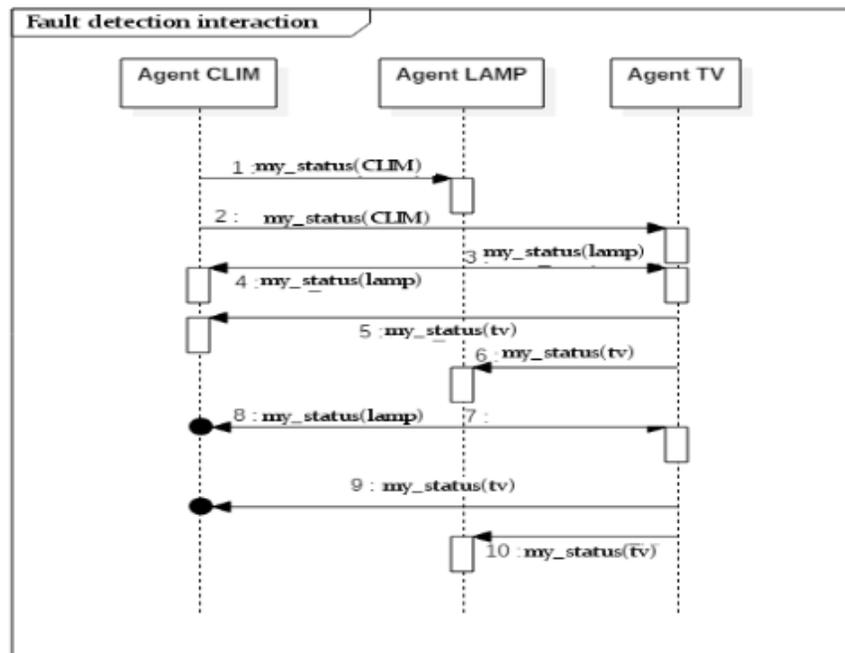


FIGURE 4.13 – diagramme de séquence de détection des désactives et échoue

La technique est utilisée après la détection de l'échec. Chaque objet de notre système choisit un entier aléatoire et diffus ce nombre à tous les objets système. L'objet qui a le nombre maximum prend le rôle d'informer le serveur de dépannage.

4.6 Conclusion

Dans ce chapitre nous avons présenté nos contributions, nous avons commencé par l'architecture générale de notre approche cloud computing basé IoT pour le smart house, nous avons détaillé chaque éléments de l'architecture.

Nous avons détaillé les différents couches du dispositif IoT en ce concentrent sur la couche ajoutée qui est "Layer Agent", Afin de minimiser le nombre de capteurs dans l'environnement, nous avons montré l'IoT Robot que nous avons expliqué chaque module et le détaillé avec des diagrammes.

Nous avons mis le point sur l'API que nous avons bati, cette API qui reflète un simulateur de de maison intelligente.

En fin et afin de visualiser la communication inter objets ,nous utiliser deux diagrammes de sequence de deux scenarios.

Chapitre 5

Résultat et mise en oeuvre

5.1 Introduction

Dans le processus du développement, l'implémentation d'un logiciel vient après un enchaînement de plusieurs étapes et son but principal est de réaliser un produit capable de résoudre les problèmes posés en utilisant des outils et des algorithmes.

Dans le but de réaliser et valider les idées proposées dans le chapitre 4, le présent chapitre montre les outils et les configurations utilisées afin de développer ce système.

Dans ce chapitre nous allons présenter les outils et les plateformes utilisés pour développer notre approche. Ensuite, nous présentons quelques interfaces qui montrent les résultats obtenus. Puis, nous présentons une discussion sur les résultats pour chaque évaluation.

5.2 Outils et plateformes utilisés

Dans cette section, nous allons présenter les plateformes utilisées afin de réaliser notre système. Les implémentations ont été réalisées en utilisant un ordinateur de processeur Core I3 avec une RAM de 8 Go sous Windows 7, mais on peut implémenter ce système sur n'importe quelle machine grâce à la machine virtuelle du Java.

Pour développer les agents, nous avons utilisés la plateforme JADE, grâce aux avantages de cette dernière qui nous offre un langage de communication entre les agents en utilisant le FIPA-ACL. Afin de simuler les idées proposées, nous avons créé une plateforme pour simuler les res-

sources virtuelles d'un smart-home ainsi que les tâches réalisées dans notre système.

5.3 Plateforme JADE

Afin d'assurer le bon fonctionnement de notre système et à cause de la nature de notre architecture qui est basée sur les agents, nous avons utilisé la plateforme de développement des systèmes multi-agent JADE, qui comporte un ensemble d'outils et d'API qui permettent la construction et la mise en service des agents sur un contexte bien spécifique.

5.3.1 Présentation générale

JADE est une plateforme implémentée avec le langage JAVA. Ce Framework est destiné aux développeurs qui s'intéressent à créer des systèmes multi-agents, et qui assure un langage de communication prédéfini et représenté dans les standards FIPA-ACL. Parmi les avantages offerts par la plateforme JADE; l'interopérabilité sans limite pour les applications quelles prennent en charge, et l'indépendance de cette plateforme du système d'exploitation ou du matériel sur laquelle elle est implémentée [28]. JADE est composée de trois principaux modules qui dépendent directement des normes FIPA :

- **DF** : pour Directory Facilitator en anglais, qui sert à la fourniture d'un service de «page jaune» à toute la plateforme JADE.
- **ACC** : pour Agent Communication Channel en anglais, qui représente l'outil de gestion de communication entre agents, pour la plateforme JADE.
- **AMS** : pour Agent Management System en anglais, qui représente l'outil de gestion (authentification, supervision, accès ... etc.) des agents de la plateforme JADE.

5.3.2 Architecture du logiciel

JADE est une plateforme basée sur les standards FIPA. En outre, le modèle de base de l'architecture de la plateforme JADE proposée par FIPA est intitulé AMRM (Agent Management Reference Model en anglais). Chaque module, qui compose l'architecture de cette plateforme, est présenté sous forme de service, ce qui permet aux agents de bénéficier d'une plateforme

orientée service, afin de faciliter la communication et la collaboration entre eux [109].

Afin d'assurer un fonctionnement efficace des agents sur la plateforme JADE, cette dernière utilise :

- **AID** : pour Agent Identifier en anglais, afin de distinguer et d'identifier chaque agent.
- **DF** : joue le rôle d'un annuaire qui enregistre les compétences de chaque agent. Les pages jaunes fournis par ce service, sont destinées à mettre en relation les différents agents fonctionnels sur la plateforme JADE, et cela pour qu'un agent puisse consulter et interroger ce service, afin d'obtenir des informations sur les compétences des autres agents et par conséquent, d'assurer une bonne collaboration entre eux.
- **AMS** : joue le rôle d'un annuaire pour l'enregistrement des adresses de transport des différents agents de la plateforme. Le but est de fournir un service de «pages blanches» afin de mettre en correspondance les agents avec l'AID, pour faciliter leur contrôle et supervision.

5.4 Simulateur smart house

Le simulateur est une API de simulation de maison intelligente basée sur le langage Java et une interface graphique très riche créée par le langage JavaFX, cette interface fournit au programmeur et au chercheur un environnement pour simuler une maison intelligente. Il lui donne également le contrôle des appareils électroménagers virtuels.

Dans notre environnement, nous avons six IoT VD, chaque appareil a deux méthodes principales :

1. **Méthode d'affectation des états** : il s'agit d'une méthode pour modifier l'état du périphérique (*ON*, *OFF*, *SUSPENDU*) pendant le fonctionnement de l'environnement sans relancer l'exécution de l'API; Le statut à passer comme paramètre dans la méthode Set (états), pour appeler la méthode set dans une autre classe, procédez comme suit :

Name of Main Class (.) LinFiSim instance (.) Set_Method(status)

```
MainClass.L.Set_LAMP_1_Status(Status.start); //Example
```

Le climatiseur virtuel a deux autres méthodes définies : la température définie et l'humidité définie qui définissent les valeurs pour stabiliser la température et l'humidité de l'atmosphère et de l'environnement. La température et l'humidité à passer en paramètre dans la méthode Set (*int*).

L'API a une méthode Status_show (*booléenne*) qui affiche un tableau dans l'interface de simulateur pour expliquer l'état de chaque périphérique si la valeur du paramètre est vraie, le tableau est affiché sinon le tableau ne s'affiche pas.

- 2. La méthode de récupération la valeur de l'état :** il s'agit d'une méthode pour obtenir l'état du périphérique (*ON, OFF, SUSPENDU*) pendant le fonctionnement de l'environnement sans relancer l'exécution de l'API; la méthode get retourne l'état du périphérique, le type de valeur retournée est status, pour appeler la méthode get dans une autre classe, définie comme suit :

Status variable=Name of Main Class (.) LinFiSim instance (.) Get_Method()

```
Status s=MainClass.L.Get_LAMP_1_Status();//Example
```

Le climatiseur virtuel a deux autres méthodes Get : Obtenir la température et l'humidité qui obtient les valeurs de la température et de l'humidité de l'atmosphère et de l'environnement. Le type de valeur de retour est un entier.

Ce tableau définit les méthodes de chaque appareil :

TABLE 5.1 – Méthodes d'IoT VDs

IoT VD	Set Method	Get Method
Lamp 1	Set_LAMP_1_Status(status)	Get_LAMP_1_Status()
Lamp 2	Set_LAMP_2_Status(status)	Get_LAMP_2_Status()
Refrigerator	Set_Refrigerator_Status(status)	Get_Refrigerator_Status()
Stove	Set_Stove_Status(status)	Get_Stove_Status()
Television	Set_TV_Status(status)	Get_TV_Status()
Air conditioner	Set_AirConditioner_Status(status) Set_AirConditioner_Temperature (int) Set_AirConditioner_humidity(int)	Get_AirConditioner_Status() Get_AirConditioner_Temperature () Get_AirConditioner_humidity ()

Dans notre environnement, nous avons également quatre IoT VS, chaque capteur possède deux méthodes principales :

TABLE 5.2 – Méthodes de l'IoT VS

IoT VS	Set Method	Get Method
Temperature Sensor 1	Set_S1_Temperature(int)	Get_S1_Temperature()
Temperature Sensor 2	Set_S2_Temperature(int)	Get_S2_Temperature()
Gas Sensor	Set_Gas(boolean)	Get_Gas()
Presence Sensor	Set_Presence(boolean)	Get_Presence()

5.4.1 Présentation des interfaces de simulateur

Dans cette section, nous expliquons et clarifions certaines méthodes d'appel de l'interface de programmation d'application.

5.4.1.1 Lancement de l'interface graphique du simulateur

Cette classe (MainLinfiSim) explique comment lancer l'interface graphique du simulateur (Figure 5.1), nous instancions un objet (L) du simulateur de classe (voir ligne 5) pour l'interface

graphique du simulateur de spectacle, nous avons déclaré une méthode main () et nous faisons appel à la méthode publique Start_LinfiSim(); (voir ligne 11) pour exécuter le code.

```
1. import java.io.IOException;
2. import linfofim.Linfofim;
3.
4. public class MainLinfofim {
5.     public static Linfofim L=new Linfofim();//declaration of Linfofim object in the
        main class
6.     /**
7.      * @param args the command line arguments
8.      * @throws java.io.IOException
9.      */
10.    public static void main(String[] args) throws IOException {
11.        L.Start_Linfofim();//Starting the simulation window Linfofim GUI
12.    }
13.
14. }
```

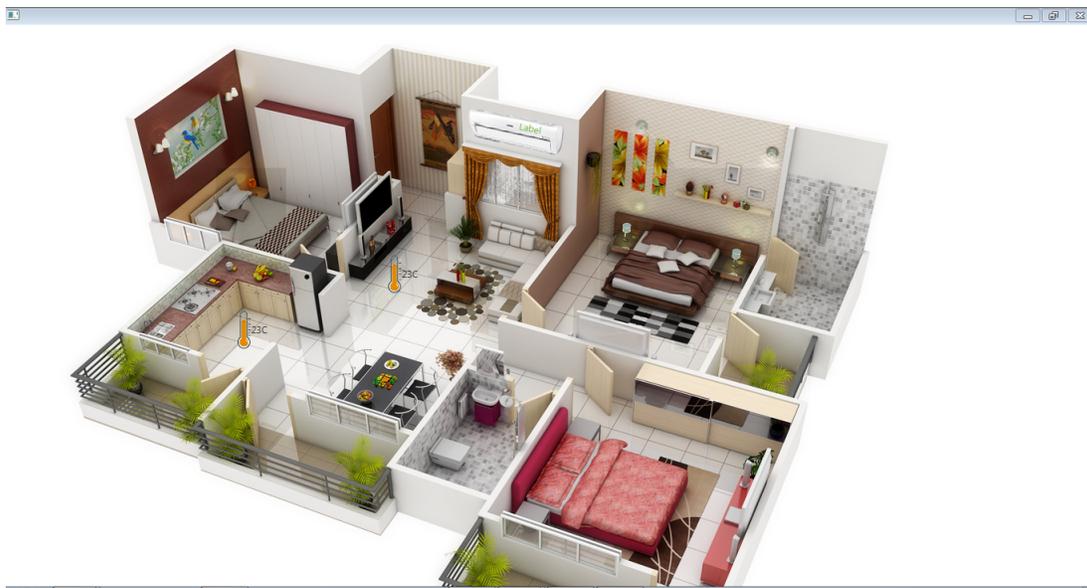


FIGURE 5.1 – LinFiSim GUI

5.4.1.2 Manipulation des états IoT VD

Cette classe (Control) explique comment utiliser l'API du simulateur, nous appelons l'objet (L) déclaré dans le MainLinfiSim classe. Et pour changer le statut de l'IoT VD, nous appelons la méthode Set_Name de VD_Status (status) et passons le statut en paramètre : par exemple (voir ligne 14), nous changeons le statut de la lampe 1 en état (Status.broken).

Pour obtenir l'état de l'IoT VD, nous avons un appel à la méthode Get_Name de VD_Status (), cette méthode retourne une valeur qui définit le statut de l'IoT VD cette valeur de type "Status" : par exemple (voir ligne 21).

```
6. import java.io.IOException;
7. import linfisim.Status;
8.
9. public class Control {
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         MainLinfiSim.L.Set_LAMP_1_Status(Status.broken);
14.         MainLinfiSim.L.Set_LAMP_2_Status(Status.start);
15.         MainLinfiSim.L.Set_TV_Status(Status.start);
16.         MainLinfiSim.L.Set_Refrigerator_Status(Status.start);
17.         MainLinfiSim.L.Set_AirConditioner_Status(Status.start);
18.         MainLinfiSim.L.Set_Stove_Status(Status.start);
19.
20.         System.out.println(MainLinfiSim.L.Get_LAMP_1_Status());
21.         System.out.println(MainLinfiSim.L.Get_LAMP_2_Status());
22.
23.     }
24. }
```



FIGURE 5.2 – IoT VD status.

5.4.1.3 Manipulation de l'IoT VS

Ce code explique comment changer la valeur des capteurs, par exemple dans les lignes 15-16, nous définissons la valeur de la température ambiante. Et dans la ligne 17, nous définissons la valeur de la température de la climatisation.

```

7. import java.io.IOException;
8. import linfoSim.Status;
9.
10. public class Control {
11.
12.     public static void main(String[] args) throws IOException {
13.
14.         MainLinfoSim.L.Set_AirConditioner_Status(Status.start);
15.         MainLinfoSim.L.Set_S1_Temperature(5);

```

```
16.         MainLinfiSim.L.Set_S2_Temperature(14);
17.         MainLinfiSim.L.Set_AIRcondition_Temperature(14);      ;
18.     }
19. }
```

5.4.1.4 Afficher le tableau des statuts

L'instruction à la ligne 14 sert à afficher la table d'état dans l'interface principale du simulateur.

```
7. import java.io.IOException;
8. import linfisim.Status;
9.
10. public class Control {
11.
12.     public static void main(String[] args) throws IOException {
13.
14.         MainLinfiSim.L.Show_Status(true);      ;
15.
16.     }
17. }
```



FIGURE 5.3 – API du simulateur avec la table d'état.

5.5 IoT Robot

Dans cette section, nous présentons les pièces électroniques les plus importantes du robot. la pièce maitresse du Robot est le circuit électronique qui est lié à un moteur pour contrôler les roues du robot (avant et arrière).

Nous avons réalisé le câblage entre les différentes unités du circuite avec Arduino et Raspberry Pi en les attachant avec un câble série pour le transfert d'informations.

Pour la communication Raspberry Pi et Arduino, les commandes reçues par le capteur WiFi sur le Raspberry Pi en utilisant le protocole MQTT le Raspberry Pi font une redirection des commandes vers l'Arduino via le port série (RX, TX).

Câblage du L293D avec moteur DC : les languettes 2, 7, 8 du L293D sont connectées aux broches 2,9 et 10 de l'Arduino, et le moteur CC avec les languettes 3,6 pour contrôler la direction du moteur (avant et arrière).

Câblage des servomoteurs : la broche du servo variateur est connectée à la broche Arduino 8 qui est responsable de la rotation de la caméra, ainsi que la deuxième broche du servo variateur avec la broche Arduino 7 qui est responsable de l'orientation des roues.

Câblage complet avec l'Arduino : Nous faisons le câblage complet avec l'Arduino, nous

connectons l'anode des LED rouges avec la broche 4 et leur cathode avec (GND) et aussi l'anode des LED blanches avec la broche 3 et leur cathode avec (GND). Nous pouvons contrôler la mise sous et hors tension des LED blanches, mais les LED rouges s'allument lorsque le robot s'arrête.

Nous avons doté le robot avec une caméra de surveillance orientable 5.4. Il est contrôlé à distance par des commandes envoyées pas depuis smartphone. Le serveur de caméra (mouvement) envoie les images capturées à l'utilisateur en temps réel et directement. Motion est un système de surveillance assez complet. Il est extrêmement personnalisable : détection de mouvement, enregistrement image par image, enregistrement vidéo, time-laps.



FIGURE 5.4 – Robot IoT

5.6 IoT device

Nous avons implémenté notre architecture pour créer un dispositif IoT de sécurité. Ceci, enfin, contrôle la sécurité d'un environnement contre les événements accidentels. Son objectif est d'agir sur l'existence d'un criminel dans l'environnement. Et il peut contrôler les risques d'accidents tels que les incendies et les fuites de gaz. La mise en oeuvre est illustrée à la figure 4.11 et expliquée ci-dessous.

Nous implémentons notre architecture proposée avec les cartes Raspberry pi 2 (voir le côté gauche sur la figure 4.11), où cette carte nous a permis de combiner les trois couches suivantes en une seule structure (comme indiqué sur le côté droit de la figure 4.11) :

- **la couche application** : il s'agit d'une application d'un véritable environnement de sécurité (maison, appartement, bureau, lieu de travail) ;
- **La couche d'agent** : c'est un agent embarqué (voir figure 4.3) dans le dispositif IoT, il gère les informations reçues par les capteurs, analyse ces informations et prend des décisions ;
- **Couche de communication réseau et données** : le périphérique IoT communique avec les autres périphériques via cette couche ;
- **Sensing Layer** : il a été implémenté sur le capteur de présence et l'humidité et capteur de température. La figure 4.3 montre l'architecture particulière de l'agent de sécurité ;
- **Environnement** : il est constitué par l'environnement réel (maison, appartement, lieu de travail), il contient les composants d'un utilisateur et voisins (l'autre composant situé au même endroit).
- **capteurs** : il s'agit d'un appareil qui permet de transformer l'état d'une grandeur physique observée (environnement) en une grandeur utilisable (mesures). Tels qu'une tension électrique, une hauteur de mercure, une intensité ou la déviation d'une aiguille. Il y a souvent une confusion entre le capteur et le transducteur : un capteur est au moins composé d'un transducteur ;
- **Base historique des États** : c'est un outil pour stocker l'historique des informations fournies par les capteurs (l'état de l'environnement) et les états des autres composants du système ;
- **Base de règles** : il rassemble les connaissances de l'agent. Il contient des règles pour aider l'agent à prendre des décisions. De telles décisions modifient son statut sur l'état actuel de l'environnement et les statuts d'autres agents. La base de règles est mise à jour en fonction des nouveaux états et besoins de l'environnement. Les règles qui représentent les connaissances générales sur la sécurité et les conclusions factuelles.

Nous avons implémenté notre architecture en utilisant Raspberry PI Model B et le système d'exploitation Raspbian. Avec eux, nous avons créé l'appareil IoT qui lance l'agent de sécurité intégré à l'appareil pour assurer la sécurité de l'environnement contre les événements accidentels et fortuits associés à un module sans fil pour la communication de l'appareil avec les autres appareils et avec Internet.

Nous avons implémenté notre agent de sécurité via le JADE (Java Agent Development Fra-

mework). Il simplifie la mise en œuvre de systèmes multi-agents grâce à un middleware conforme aux spécifications de la Foundation for Intelligent Physical Agents (FIPA) et à travers un ensemble d'outils graphiques qui prennent en charge les phases de débogage et de déploiement.

Nous utilisons deux types de capteurs pour fournir la couche de détection : un capteur de mouvement PIR (HCSR501 PIR) pour détecter la présence d'une personne (intrus) dans l'environnement et un capteur de gaz pour détecter les fuites de gaz à l'emplacement. Nous avons connecté des capteurs électroniques au Raspberry Pi Model B. De plus, nous les avons contrôlés avec Java en utilisant une entrée / sortie à usage général (GPIO).

5.6.1 Composantes électroniques de IoT device

Cette section représente l'ensemble des pièces électroniques responsables de la transformation (capteurs), de l'analyse, du traitement (Raspberry PI) et du transfert (module de communication) de l'état d'une grandeur physique observée en quantité utilisable.

5.6.1.1 PIR Motion Sensor (HCSR501 PIR)

PIR sensors (Figure 5.5) nous permettent de détecter le mouvement et ils sont toujours utilisés pour détecter si un humain est entré / sorti de la plage des capteurs. Ils sont petits, peu coûteux, de faible puissance, faciles à utiliser et ne s'usent pas. Pour cette raison, ils se trouvent généralement dans les appareils et gadgets utilisés dans les maisons ou les entreprises. Ils sont souvent appelés capteurs PIR, «infrarouge passif», «pyroélectrique» ou «mouvement PIR».

5.6.1.2 Capteur d'humidité et de température (DHT-11)

Figure 5.5 montre le module de capteur d'humidité et de température et est basé sur le capteur DHT-11, qui est un capteur de température et d'humidité numérique très populaire et à très bas prix. Il utilise un capteur d'humidité capacitif et une thermistance pour mesurer la température de l'air ambiant. Le résultat est disponible via un signal numérique sur la broche de données (aucune broche d'entrée analogique n'est nécessaire). Il peut mesurer l'humidité de 20 à 80% des lectures d'humidité avec une précision de 5% et des lectures de température de 0 à 50 ° C avec une précision de ± 2 ° C.

5.6.1.3 Raspberry Pi Model B

Figure 5.5 montre le Raspberry Pi Model B qui est une petite version d'un ordinateur exécutant le système d'exploitation Linux sur carte SD pour une application informatique intégrée. En raison de ses caractéristiques, la couche d'agent de déploiement dans l'appareil et le fonctionnement sont la capacité de stockage et de traitement pour améliorer les performances dans lesquelles cette carte est équipée :

- Broadcom BCM2836 processor, four 900 MHz ARMv7 cores;
- 1 GB of DDR2 memory;
- Ethernet / RJ45 port, 10.100 BaseT is connect Raspberry Pi to a network;
- Four USB 2.0 ports;
- HDMI video output for connecting a display;
- Taken for camera;
- 40 pins GPIO connectors (all running at 3.3 V);
- Memory card port : 1 micro SD port is used as a storage medium;
- 1 Micro USB for power supply;
- Analogue audio connectors : 1 3.5mm jack output;

5.6.2 Câblage et circuit électronique

Après avoir préparé les composants électroniques et la carte Raspberry PI , nous devons connecter les pièces nécessaires pour rendre le système opérationnel (voir figure 5.5).

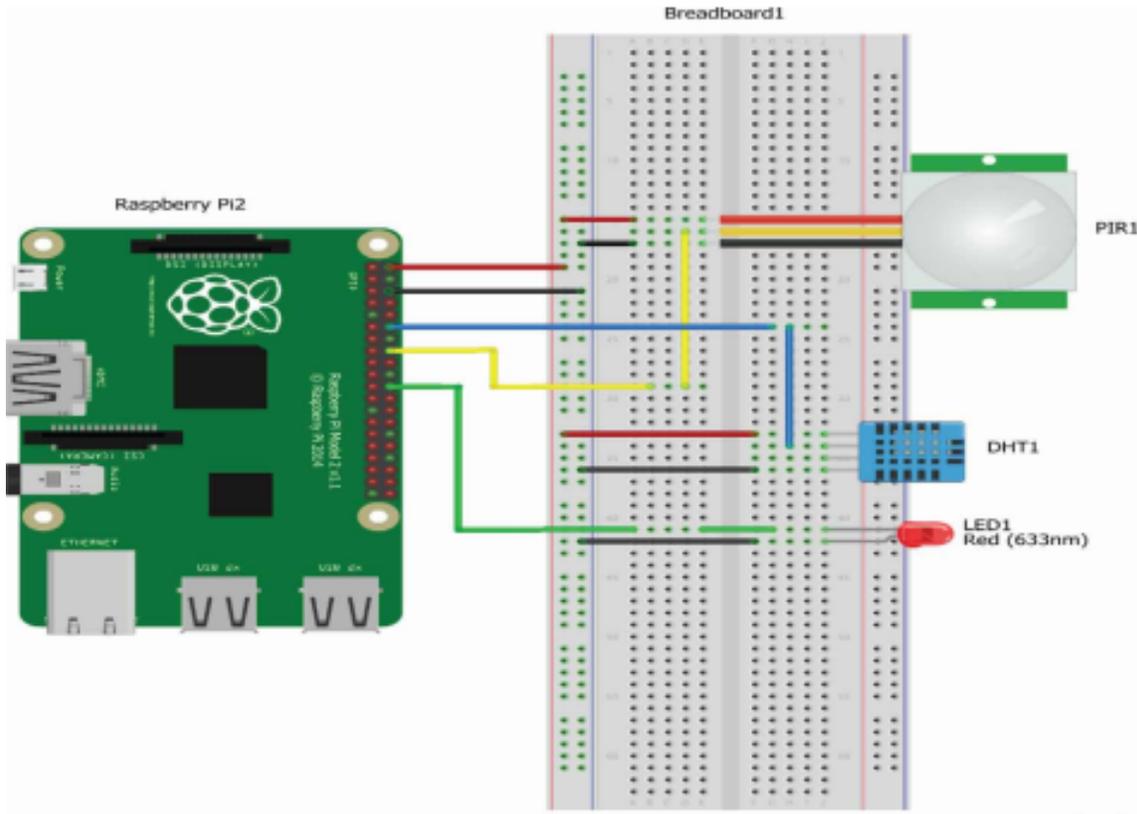


FIGURE 5.5 – Circuit électronique de IoT device

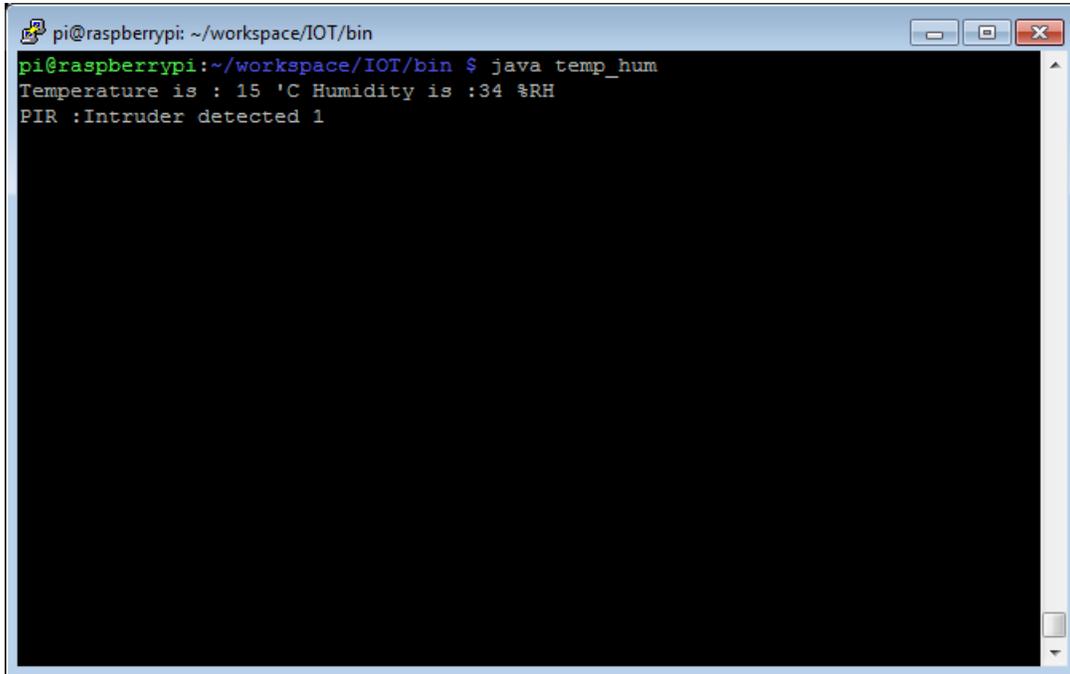
5.6.3 Fonctionnalités de IoT device

Pour tester notre application , nous avons collecté les informations (température l'humidité) dans la période 19-20 décembre 2017 dans la ville de Biskra, Algérie.

Le système travaille de manière continue est independante, détecte la température et l'humidité de l'environnement, puis stocke les données dans sa propre base de données. Il avertit également l'utilisateur en cas de changements importants de température et d'humidité.

En plus il protège également l'environnement contre les intrus en l'absence de l'utilisateur lorsque le système avertit l'utilisateur.

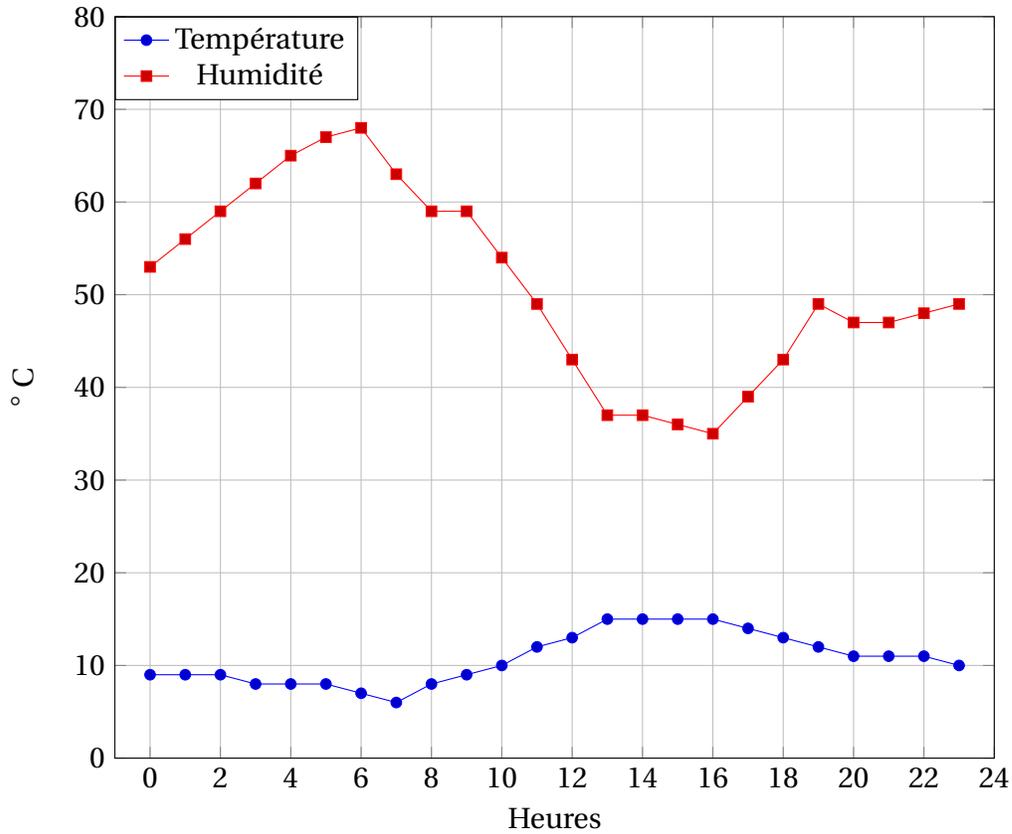
La figure 5.6 montre la température et l'humidité détectées par le systeme ainsi que la detection d'un intrus dans le smart house.



```
pi@raspberrypi: ~/workspace/IOT/bin
pi@raspberrypi:~/workspace/IOT/bin $ java temp_hum
Temperature is : 15 'C Humidity is :34 %RH
PIR :Intruder detected 1
```

FIGURE 5.6 – Agent sécurité IoT détecte la présence d’une personne à la maison.

La courbe suivante résume les changements de température et d’humidité dans l’atmosphère capturés en 24 heures, la température est comprise entre 6 et 15 degrés tandis que l’humidité est comprise entre 35 et 68 degrés.



La fenêtre suivante explique l'échange de messages entre les agents, l'agent "IoT security" n'envoie pas de message aux autres agents car il ne détecte aucun danger dans l'environnement.

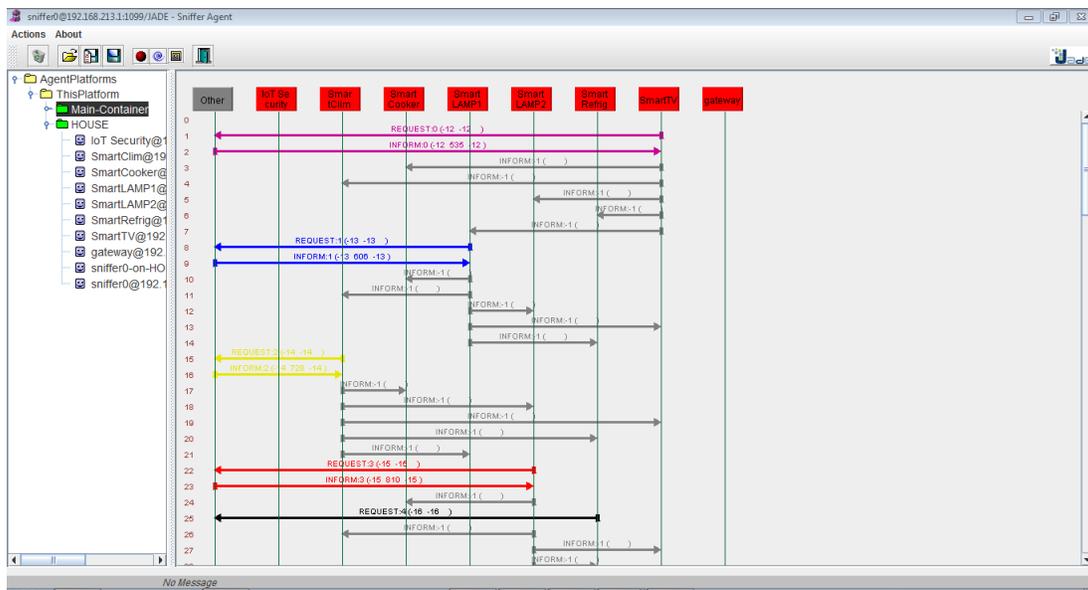


FIGURE 5.7 – Communication entre agents

La fenêtre suivante explique la réaction de l'agent "sécurité IoT" après la détection d'une personne dans l'environnement et le mode de surveillance est à l'état actif.

L'agent «sécurité IoT» a diffusé un message de «demande de réaction» et chaque agent reçoit ce message qui répondra par l'action appropriée à la demande.

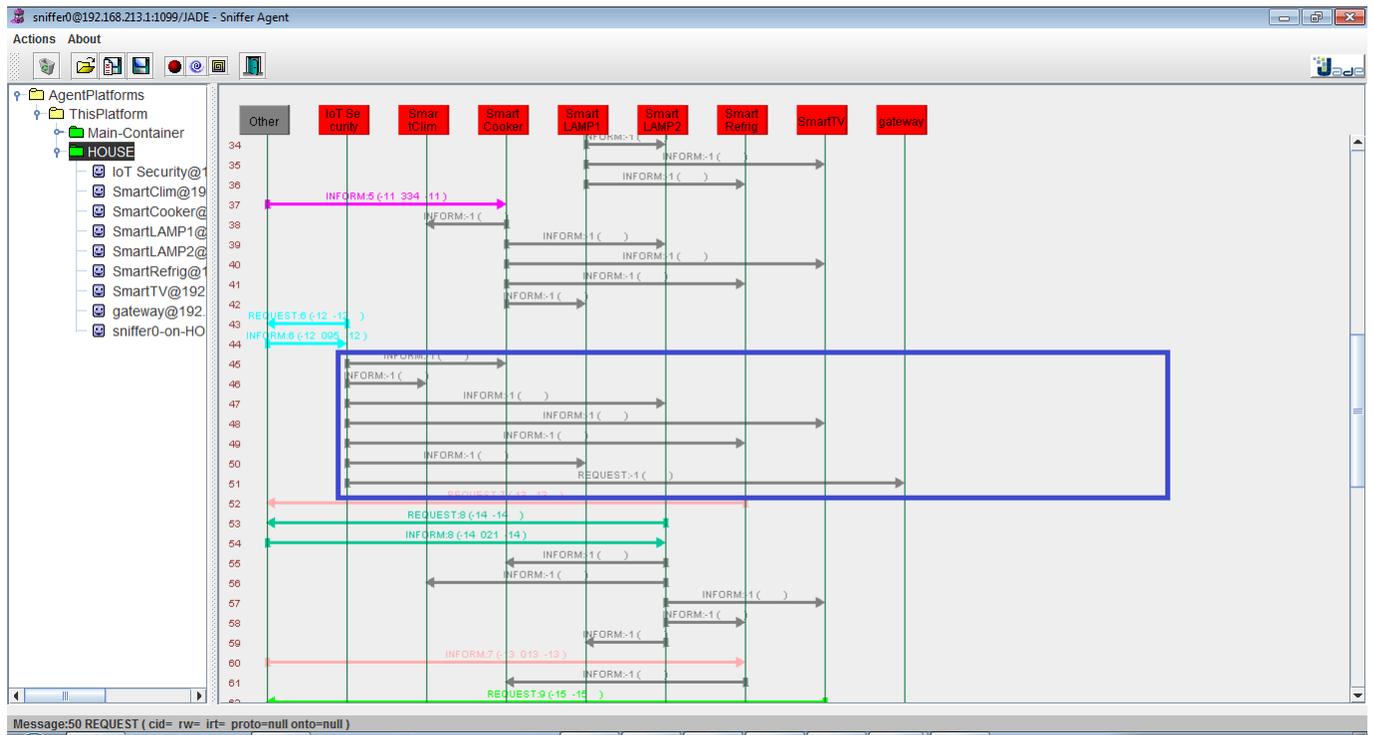


FIGURE 5.8 – Réaction de l'agent sécurité IoT

L'agent alerte également l'utilisateur avec un message envoyé au smartphone. La figure suivante montre le message que l'utilisateur reçoit immédiatement après que l'appareil détecte un intrus dans la maison.

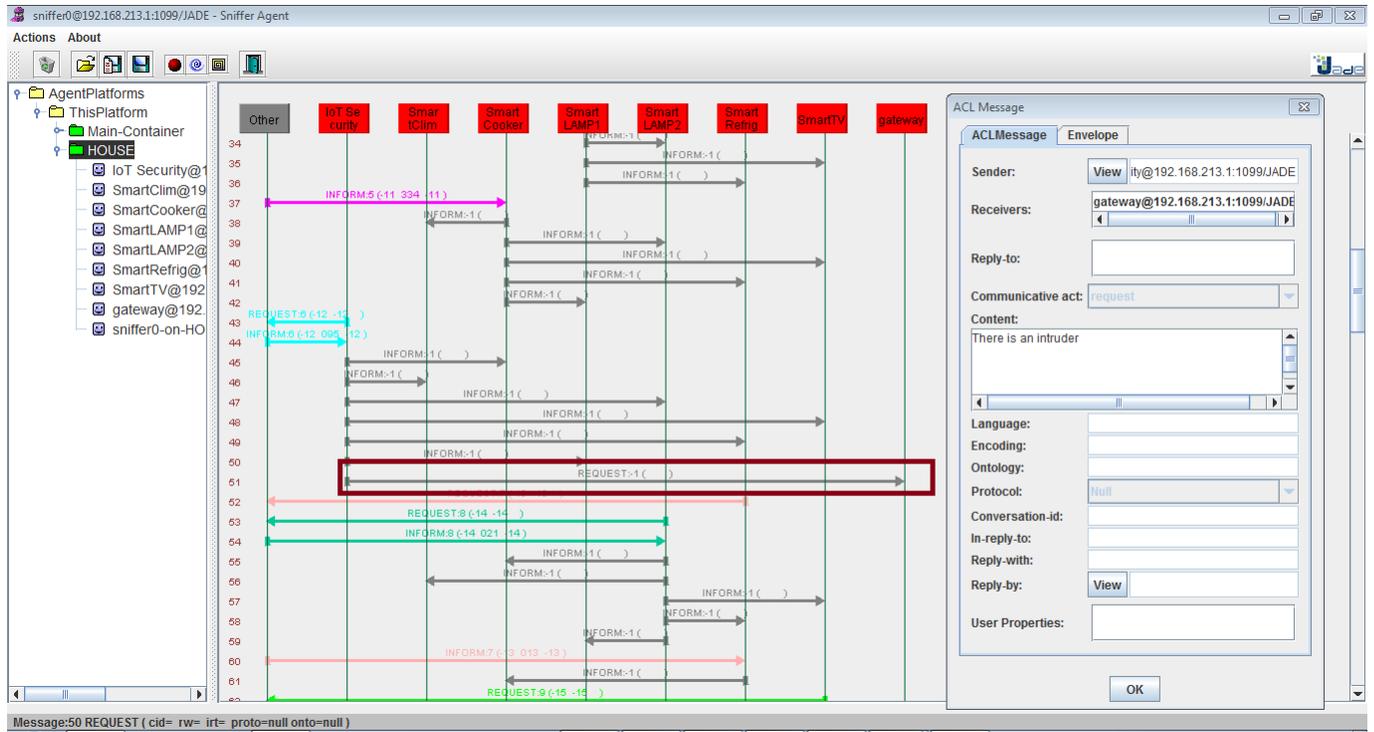


FIGURE 5.9 – Agent de sécurité IoT envoie un message à l'utilisateur.

5.7 Conclusion

Ce chapitre a été consacré à l'implémentation de notre solution IoT. Au début nous avons présenté les outils et plateformes utilisés.

Nous avons commencé par l'implémentation sur simulateur smart house où nous avons expliqué et clarifié certaines méthodes. Nous avons montré comment utiliser l'API de simulateur et comment manipuler les capteurs IoT.

Dans la seconde implémentation, nous avons présenté l'IoT Robot, ses composants électroniques et son fonctionnement.

Enfin, nous avons terminé ce chapitre par l'implémentation de l'IoT device dont nous avons présenté les composants électronique et aussi le fonctionnement logique de l'IoT device.

Conclusion générale et perspectives

Conclusion

La nouvelle informatique se caractérise par des nouveaux domaines qui représentent la tendance de la technique comme le cloud computing et l'internet des objets et le big data prenant une grande partie de cette nouvelle informatique. Le Big Data représente un changement significatif dans les technologies de l'information. Techniquement, nous vivons un véritable phénomène de rupture. En effet, au-delà de quelques dizaines de téraoctets, les technologies traditionnelles sont inadéquates, elles ne permettent plus d'analyser la volumétrie élevée et disparate des données (le 3V du Big Data), et la question qui se pose alors est de savoir comment et où se traiter cette quantité de donnée, au niveau de l'objet ou cloud, chacune a des avantages et les inconvénients et des limites.

Dans cette thèse, nous avons décrit en détail nos contributions :

Notre première contribution consiste en la proposition d'une nouvelle approche basée agents pour smart house. Nous avons utilisé la plate-forme Jade dans laquelle nous avons lancé nos agents. Cette plate-forme est utilisée dans les différentes implémentations pour héberger des agents, nous avons utilisé le protocole MQTT également pour transmettre les données entre les objets (edges), Fog et le cloud.

Dans la deuxième contribution, nous avons présenté une architecture IoT autonome, c'est une nouvelle architecture pour l'IoT en utilisant le paradigme agent. Nous avons profité des fonctionnalités de l'agent pour enrichir l'appareil de qualités d'autonomie, de coordination et de coopération. Nous avons implémenté cette approche en utilisant une Raspberry Pi avec intégration des capteurs et en utilisant le protocole MQTT pour les communications.

Dans la troisième contribution, nous avons proposé un Robot IoT basé sur la deuxième

contribution. Ce robot est doté d'une caméra rotative et des capteurs intégrés. Son rôle principal est de percevoir l'environnement à la demande des objets, autrement, il offre un service de sensing (sensing as service).

Pour la quatrième et la dernière contribution, nous avons créé une API qui simule une maison intelligente, cette maison intelligente consiste à connecter les différents appareils et systèmes de la maison afin qu'ils puissent être contrôlés de n'importe où et provoquer l'interaction souhaitée entre eux. Ces appareils sont des objets connectés via un réseau qui a une fenêtre sur Internet.

Toutes ces contributions ont été implémentées en utilisant la plateforme JADE pour manipuler les agents et les langages JAVA et JAVAFX comme outils de programmation.

Pour concrétiser nos solutions, nous avons utilisé les cartes électroniques telles que : Raspberry PI, Arduino et des capteurs.

Perspectives

Dans le futur et comme perspectives de ce travail, nous considérerons les extensions suivantes :

- Faire un couplage avec le modèle de Big-Data;
- exploiter la technologie Cloud Robotics pour les smart cities;
- Orienter nos prochains travaux vers la sécurité des maisons intelligentes;
- Développer une stratégie basée sur nos contributions pour la sécurité dans une smart city;
- faire une projection de nos résultats pour les véhicules autonomes;

Annexe A

Liste des publications

A.1 Revues Internationales

Meftah. Zouai, Okba. Kazar, Belgacem. Haba, Hamza. Saouli, Hind. Benfenati ***IoT Approach Using Multi-Agent System for Ambient Intelligence***, *International Journal of Software Engineering and Its Applications*, vol11_no9_2017, (pp. 15-32).

Meftah.Zouai,Okba.Kazar,Guadalupe.Ortiz,Belgacem.Haba,Kabach. Nadia,krishnamyrhty ***Ambiance Intelligence Approach Using IoT and Multi-Agent System***, *International Journal of Distributed Systems and Technologies (IJDST)* ,10(1),(pp. 15-32),2019.

Meftah.Zouai,Okba.Kazar,Guadalupe.Ortiz,Belgacem.Haba,Kabach. Nadi, ***New approach using an IoT robot to oversightthe smart home environment***, *Internet of Things* ,10(1),(pp. 1-6),2019.

A.2 Conférences Internationales

Meftah. Zouai, Okba. Kazar, Belgacem. Haba, Hamza. Saouli, ***Smart house simulation based multi-agent system and internet of things***, *International Conference on Mathematics and Information Technology (ICMIT)*, Université de adrar,2017.

Meftah. Zouai, Okba. Kazar, Belgacem. Haba, Dounya. Kassimi, ***Simulation of an ambient environment using multi-agent systems***, *1er Journées Doctoriales Informatique, Théorique et Appliquée (JDITA'2018)*, Université de Biskra,2018.

Meftah. Zouai, Okba. Kazar, Belgacem. Haba,Guadalupe. Ortiz, Nadia. Kabachi, ***Using an***

IoT mobile robot to sensing the environment for smart home, *International Conference on Advanced Intelligent Systems for Sustainable Development (IA2SD), Université Abdelmalek Essaâdi, Tangier, Morocco, 2018.*

Abdelhak. Merizig, Hamza. Saouli, Meftah. Zouai, Okba. Kazar , ***An intelligent approach for enhancing the agricultural production in arid areas using IOT technology***, *International Conference on Advanced Intelligent Systems for Sustainable Development (IA2SD), Université Abdelmalek Essaâdi, Tangier, Morocco, 2018.*

Bibliographie

- [1] M Abomhara and G Køien. Cyber security and the internet of things. 2015.
- [2] Rachit Agarwal, David Gomez Fernandez, Tarek Elsaleh, Amelie Gyrard, Jorge Lanza, Luis Sanchez, Nikolaos Georgantas, and Valerie Issarny. Unified iot ontology to enable interoperability and federation of testbeds. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 70–75. IEEE, 2016.
- [3] Shashank Agrawal and Dario Vieira. A survey on internet of things. *Abakós*, 1(2) :78–95, 2013.
- [4] Hayder AA Al-Kashoash and Andrew H Kemp. Comparison of 6lowpan and lpwan for the internet of things. *Australian Journal of Electrical and Electronics Engineering*, 13(4) :268–274, 2016.
- [5] Majid Al-Kuwari, Abdulrhman Ramadan, Yousef Ismael, Laith Al-Sughair, Adel Gastli, and Mohieddine Benammar. Smart-home automation using iot-based sensing and monitoring platform. In *2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018)*, pages 1–6. IEEE, 2018.
- [6] AK Al-Qurabat and Ali Kadhum Idrees. Adaptive data collection protocol for extending lifetime of periodic sensor networks. *Qalaai Zanist Scientific Journal*, 2(2) :83–92, 2017.
- [7] Fadele Ayotunde Alaba, Mazliza Othman, Ibrahim Abaker Targio Hashem, and Faiz Alo-taibi. Internet of things security : A survey. *Journal of Network and Computer Applications*, 88 :10–28, 2017.

-
- [8] Sara Amendola, Rossella Lodato, Sabina Manzari, Cecilia Occhiuzzi, and Gaetano Marrocco. Rfid technology for iot-based personal healthcare in smart spaces. *IEEE Internet of things journal*, 1(2) :144–152, 2014.
- [9] Noah Apthorpe, Dillon Reisman, and Nick Feamster. A smart home is no castle : Privacy vulnerabilities of encrypted iot traffic. *arXiv preprint arXiv :1705.06805*, 2017.
- [10] Julio Arauz and Tony Fynn-Cudjoe. Actuator quality in the internet of things. In *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, pages 34–42. IEEE, 2013.
- [11] Muhammad Asadullah and Khalil Ullah. Smart home automation system using bluetooth technology. In *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, pages 1–6. IEEE, 2017.
- [12] K Ashokkumar, Baron Sam, R Arshadprabhu, et al. Cloud based intelligent transport system. *Procedia Computer Science*, 50 :58–63, 2015.
- [13] Luigi Atzori, Antonio Iera, Giacomo Morabito, and Michele Nitti. The social internet of things (siot)–when social networks meet the internet of things : Concept, architecture and network characterization. *Computer networks*, 56(16) :3594–3608, 2012.
- [14] Nazmiye Balta-Ozkan, Rosemary Davidson, Martha Bicket, and Lorraine Whitmarsh. The development of smart homes market in the uk. *Energy*, 60 :361–372, 2013.
- [15] Oladayo Bello and Sherali Zeadally. Intelligent device-to-device communication in the internet of things. *IEEE Systems Journal*, 10(3) :1172–1182, 2014.
- [16] Sven Bendel, Thomas Springer, Daniel Schuster, Alexander Schill, Ralf Ackermann, and Michael Ameling. A service infrastructure for the internet of things based on xmpp. In *2013 IEEE international conference on pervasive computing and communications workshops (PERCOM Workshops)*, pages 385–388. IEEE, 2013.
- [17] Jayant D Bokefode, Swapnaja A Ubale, and Roopali M Gaikwad. Retrieving real time data through iot devices and storing securely on cloud using li-fi. In *2018 3rd International Conference for Convergence in Technology (I2CT)*, pages 1–5. IEEE, 2018.

- [18] Carsten Bormann, Angelo P Castellani, and Zach Shelby. Coap : An application protocol for billions of tiny internet nodes. *IEEE Internet Computing*, 16(2) :62–67, 2012.
- [19] Alessio Botta, Walter De Donato, Valerio Persico, and Antonio Pescapé. Integration of cloud computing and internet of things : a survey. *Future generation computer systems*, 56 :684–700, 2016.
- [20] Z Bozdogan and R Kara. Layered model architecture for internet of things. *Journal of Engineering Research and Applied Science*, 4(1) :260–264, 2015.
- [21] Clark Bradley, Ralph Hollinshead, Scott Kraus, Jason Lefler, and Roshan Taheri. Data modeling considerations in hadoop and hive. *Technical paper*, SAS, 2013.
- [22] Rajkumar Buyya, Christian Vecchiola, and S Thamarai Selvi. *Mastering cloud computing : foundations and applications programming*. Newnes, 2013.
- [23] Luca Catarinucci, Danilo De Donno, Luca Mainetti, Luca Palano, Luigi Patrono, Maria Laura Stefanizzi, and Luciano Tarricone. An iot-aware architecture for smart healthcare systems. *IEEE internet of things journal*, 2(6) :515–526, 2015.
- [24] Kuor-Hsin Chang. Bluetooth : a viable solution for iot?[industry perspectives]. *IEEE Wireless Communications*, 21(6) :6–7, 2014.
- [25] Dustin Chen and Qibing Pei. Electronic muscles and skins : a review of soft sensors and actuators. *Chemical reviews*, 117(17) :11239–11268, 2017.
- [26] Min Chen, Jun Yang, Xuan Zhu, Xiaofei Wang, Mengchen Liu, and Jeungeun Song. Smart home 2.0 : Innovative smart home system powered by botanical iot and emotion detection. *Mobile Networks and Applications*, 22(6) :1159–1169, 2017.
- [27] Hsu-Chen Cheng and Wen-Wei Liao. Establishing an lifelong learning environment using iot and learning analytics. In *2012 14th International Conference on Advanced Communication Technology (ICACT)*, pages 1178–1183. IEEE, 2012.
- [28] Krzysztof Chmiel, Maciej Gawinecki, Pawel Kaczmarek, Michal Szymczak, and Marcin Paprzycki. Efficiency of jade agent platform. *Scientific Programming*, 13(2) :159–172, 2005.

- [29] He Chun-Zhi, Xia Yin-shui, and Wang Lun-yao. A universal asynchronous receiver transmitter design. In *2011 International Conference on Electronics, Communications and Control (ICECC)*, pages 691–694. IEEE, 2011.
- [30] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries : A survey. *IEEE Transactions on industrial informatics*, 10(4) :2233–2243, 2014.
- [31] Dina Darwish. Improved layered architecture for internet of things. *Int. J. Comput. Acad. Res.(IJCAR)*, 4 :214–223, 2015.
- [32] Mohammad Aizuddin Daud and Wida Susanty Haji Suhaili. Internet of things (iot) with coap and http protocol : A study on which protocol suits iot in terms of performance. In *International Conference on Computational Intelligence in Information System*, pages 165–174. Springer, 2016.
- [33] Jeff Donald Drake. Zigbee ip/6lowpan router, May 13 2014. US Patent 8,724,490.
- [34] Yonina C Eldar and Gitta Kutyniok. *Compressed sensing : theory and applications*. Cambridge university press, 2012.
- [35] Deborah Estrin, David Culler, Kris Pister, and Gaurav Sukhatme. Connecting the physical world with pervasive networks. *IEEE pervasive computing*, 1(1) :59–69, 2002.
- [36] Rosario Fedele, Massimo Merenda, and Filippo Giammaria. Energy harvesting for iot road monitoring systems. *Instrumentation, Mesure, Metrologie*, 17(4) :605, 2018.
- [37] Tang Hai Yang Feng. Zigbee integrated remoter technology [j]. *Electronic Science and Technology*, 8, 2008.
- [38] Giancarlo Fortino, Claudio Savaglio, Carlos E Palau, Jara Suarez de Puga, Maria Ganzha, Marcin Paprzycki, Miguel Montesinos, Antonio Liotta, and Miguel Llop. Towards multi-layer interoperability of heterogeneous iot platforms : The inter-iot approach. In *Integration, interconnection, and interoperability of IoT systems*, pages 199–232. Springer, 2018.
- [39] Dimitris Geneiatakis, Ioannis Kounelis, Ricardo Neisse, Igor Nai-Fovino, Gary Steri, and Gianmarco Baldini. Security and privacy issues for an iot based smart home. In *2017*

- 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1292–1297. IEEE, 2017.
- [40] Khusvinder Gill, Shuang-Hua Yang, Fang Yao, and Xin Lu. A zigbee-based home automation system. *IEEE Transactions on Consumer Electronics*, 55(2) :422–430, 2009.
- [41] Carles Gomez, Josep Paradells, Carsten Bormann, and Jon Crowcroft. From 6lowpan to 6lo : Expanding the universe of ipv6-supported technologies for the internet of things. *IEEE Communications Magazine*, 55(12) :148–155, 2017.
- [42] Haijun Gu, Yufeng Diao, Wei Liu, and Xueqian Zhang. The design of smart home platform based on cloud computing. In *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, volume 8, pages 3919–3922. IEEE, 2011.
- [43] 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, 2013.
- [44] Harald Haas. Lifi is a paradigm-shifting 5g technology. *Reviews in Physics*, 3 :26–31, 2018.
- [45] Harald Haas, Liang Yin, Yunlu Wang, and Cheng Chen. What is lifi? *Journal of lightwave technology*, 34(6) :1533–1544, 2015.
- [46] Ronald van Ham, Thomas Sugar, Bram Vanderborght, Kevin Hollander, and Dirk Lefeber. Compliant actuator designs. *IEEE Robotics & Automation Magazine*, 3(16) :81–94, 2009.
- [47] Richard Harper. *Inside the smart home*. Springer Science & Business Media, 2006.
- [48] Sarah Hasan and Edwin Yaz. Intelligent transportation systems. Technical report, Springer, 1999.
- [49] David F Hoeschele. *Analog-to-digital and digital-to-analog conversion techniques*, volume 968. Wiley New York, 1994.
- [50] Jiří Homola. On the sensitivity of surface plasmon resonance sensors with spectral interrogation. *Sensors and Actuators B : Chemical*, 41(1-3) :207–211, 1997.

-
- [51] RÉVEILLAC Jean-Michel. *La réalité augmentée : techniques et entités virtuelles*. Lavoisier, 2013.
- [52] Xiaolin Jia, Quanyuan Feng, Taihua Fan, and Quanshui Lei. Rfid technology and its applications in internet of things (iot). In *2012 2nd international conference on consumer electronics, communications and networks (CECNet)*, pages 1282–1285. IEEE, 2012.
- [53] Manveer Joshi and Bikrampal Kaur. Web integrated smart home infrastructure using internet of things. *International Journal of Engineering Research and General Science*, 3(6), 2015.
- [54] Carlos MJM Dourado Junior, Suane PP da Silva, Raul VM da Nóbrega, Antonio CS Barros, Arun K Sangaiah, Pedro P Rebouças Filho, and Victor Hugo C de Albuquerque. A new approach for mobile robot localization based on an online iot system. *Future Generation Computer Systems*, 100 :859–881, 2019.
- [55] Manfred Kaltenbacher. *Numerical simulation of mechatronic sensors and actuators*, volume 2. Springer, 2007.
- [56] Mahdi Kasmi, Faouzi Bahloul, and Haykel Tkitek. Smart home based on internet of things and cloud computing. In *2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pages 82–86. IEEE, 2016.
- [57] Sean Dieter Tebje Kelly, Nagender Kumar Suryadevara, and Subhas Chandra Mukhopadhyay. Towards the implementation of iot for environmental condition monitoring in homes. *IEEE sensors journal*, 13(10) :3846–3853, 2013.
- [58] Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. Future internet : the internet of things architecture, possible applications and key challenges. In *2012 10th international conference on frontiers of information technology*, pages 257–260. IEEE, 2012.
- [59] Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. Future internet : the internet of things architecture, possible applications and key challenges. In *2012*

- 10th international conference on frontiers of information technology*, pages 257–260. IEEE, 2012.
- [60] Seong-Min Kim, Hoan-Suk Choi, and Woo-Seop Rhee. Iot home gateway for auto-configuration and management of mqtt devices. In *2015 IEEE Conference on Wireless Sensors (ICWiSe)*, pages 12–17. IEEE, 2015.
- [61] Ravi Kishore Kodali, Vishal Jain, Suvadeep Bose, and Lakshmi Boppana. Iot based smart security and home automation system. In *2016 international conference on computing, communication and automation (ICCCA)*, pages 1286–1289. IEEE, 2016.
- [62] Linghe Kong, Muhammad Khurram Khan, Fan Wu, Guihai Chen, and Peng Zeng. Millimeter-wave wireless communications for iot-cloud supported autonomous vehicles : Overview, design, and challenges. *IEEE Communications Magazine*, 55(1) :62–68, 2017.
- [63] Valerie Lampkin, Weng Tat Leong, Leonardo Olivera, Sweta Rawat, Nagesh Subrahmanyam, Rong Xiang, Gerald Kallas, Neeraj Krishna, Stefan Fassmann, Martin Keen, et al. *Building smarter planet solutions with mqtt and ibm websphere mq telemetry*. IBM Redbooks, 2012.
- [64] Marco Leo, Federica Battisti, Marco Carli, and Alessandro Neri. A federated architecture approach for internet of things security. In *2014 Euro Med Telco Conference (EMTC)*, pages 1–5. IEEE, 2014.
- [65] Peter Leong and Liming Lu. Multiagent web for the internet of things. In *2014 International Conference on Information Science & Applications (ICISA)*, pages 1–4. IEEE, 2014.
- [66] Biqing Li, Yongfa Ling, Hongyan Zhang, and Shiyong Zheng. The design and realization of cherry tomato harvesting robot based on iot. *International Journal of Online and Biomedical Engineering (iJOE)*, 12(12) :23–26, 2016.
- [67] Francesco Longo, Dario Bruneo, Salvatore Distefano, Giovanni Merlino, and Antonio Puliato. Stack4things : a sensing-and-actuation-as-a-service framework for iot and cloud integration. *Annals of Telecommunications*, 72(1-2) :53–70, 2017.

- [68] H el ene GRIMAL LOU, C ecile LEV EQUE, and BELY H el ene. Enjeux & prospective en midi-pyr en ees : les m etiers du num erique. 2015.
- [69] Chia-Wen Lu, Shu-Cheng Li, and Quincy Wu. Interconnecting zigbee and 6lowpan wireless sensor networks for smart grid applications. In *2011 Fifth International Conference on Sensing Technology*, pages 267–272. IEEE, 2011.
- [70] Rwan Mahmoud, Tasneem Yousuf, Fadi Aloul, and Imran Zualkernan. Internet of things (iot) security : Current status, challenges and prospective measures. In *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 336–341. IEEE, 2015.
- [71] Luca Mainetti, Luigi Patrono, and Antonio Vilei. Evolution of wireless sensor networks towards the internet of things : A survey. In *SoftCOM 2011, 19th international conference on software, telecommunications and computer networks*, pages 1–6. IEEE, 2011.
- [72] Stevan J Marinkovic and Emanuel M Popovici. Nano-power wireless wake-up receiver with serial peripheral interface. *IEEE Journal on selected areas in communications*, 29(8) :1641–1647, 2011.
- [73] High-Order Hidden Bivariate Markov. Technical program.
- [74] Alberto Marroquin, Adalberto Gomez, and Alejandro Paz. Design and implementation of explorer mobile robot controlled remotely using iot technology. In *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, pages 1–7. IEEE, 2017.
- [75] James L McClelland and Timothy T Rogers. The parallel distributed processing approach to semantic cognition. *Nature reviews neuroscience*, 4(4) :310–322, 2003.
- [76] Muhammad Hunain Memon, Wanod Kumar, AzamRafique Memon, Bhawani S Chowdhry, Muhammad Aamir, and Pardeep Kumar. Internet of things (iot) enabled smart animal farm. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 2067–2072. IEEE, 2016.

- [77] Peter V Messina. Integrated system for line-of-sight stabilization and auto-alignment of off-gimbal passive and active electro-optical sensors, September 11 2001. US Patent 6,288,381.
- [78] Vivek P Mhatre, Catherine Rosenberg, Daniel Kofman, Ravi Mazumdar, and Ness Shroff. A minimum cost heterogeneous sensor network with a lifetime constraint. *IEEE Transactions on Mobile Computing*, 4(1) :4–15, 2005.
- [79] Jirapond Muangprathub, Nathaphon Boonnam, Siriwan Kajornkasirat, Narongsak Lekbangpong, Apirat Wanichsombat, and Pichetwut Nillaor. Iot and agriculture data analysis for smart farm. *Computers and electronics in agriculture*, 156 :467–474, 2019.
- [80] Gerrit Mur. Edge elements, their advantages and their disadvantages. *IEEE transactions on magnetics*, 30(5) :3552–3557, 1994.
- [81] Anas M Mzahm, Mohd Sharifuddin Ahmad, and Alicia YC Tang. Agents of things (aot) : An intelligent operational concept of the internet of things (iot). In *2013 13th International Conference on Intelligent Systems Design and Applications*, pages 159–164. IEEE, 2013.
- [82] Nitin Naik. Choice of effective messaging protocols for iot systems : Mqtt, coap, amqp and http. In *2017 IEEE international systems engineering symposium (ISSE)*, pages 1–7. IEEE, 2017.
- [83] Swapnil Naik and Vikas Maral. Cyber security—iot. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 764–767. IEEE, 2017.
- [84] C Nathani, D Sutter, R van Nieuwkoop, S Kraner, M Peter, and R Zandonella. Energiebezogene differenzierung der schweizerischen iot 2008 und revision der energie-iot 2001 und 2005. *Office fédéral de l'énergie (OFEN)*, Berne, 2013.
- [85] Awa Ndoye. *Migration d'OpenERP dans le cloud computing pour une PME*. PhD thesis, Haute école de gestion de Genève, 2012.

- [86] Luís Nóbrega, André Tavares, António Cardoso, and Pedro Gonçalves. Animal monitoring based on iot technologies. In *2018 IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany)*, pages 1–5. IEEE, 2018.
- [87] Vanga Odelu, Ashok Kumar Das, Muhammad Khurram Khan, Kim-Kwang Raymond Choo, and Minho Jo. Expressive cp-abe scheme for mobile devices in iot satisfying constant-size keys and ciphertexts. *IEEE Access*, 5 :3273–3283, 2017.
- [88] Ramon Pallas-Areny and John G Webster. *Sensors and signal conditioning*. John Wiley & Sons, 2012.
- [89] Marc Pastre, Hubert Blanchard, and Maher Kayal. Continuously calibrated magnetic field sensor, June 29 2010. US Patent 7,746,065.
- [90] D Pavithra and Ranjith Balakrishnan. Iot based monitoring and control system for home automation. In *2015 global conference on communication technologies (GCCT)*, pages 169–173. IEEE, 2015.
- [91] Vitaly Petrov, Andrey Samuylov, Vyacheslav Begishev, Dmitri Moltchanov, Sergey Andreev, Konstantin Samouylov, and Yevgeni Koucheryavy. Vehicle-based relay assistance for opportunistic crowdsensing over narrowband iot (nb-iot). *IEEE Internet of Things journal*, 5(5) :3710–3723, 2017.
- [92] Guillaume Plouin. *Cloud Computing-2e éd. : Une rupture décisive pour l'informatique d'entreprise*. Dunod, 2011.
- [93] Guillaume Plouin. *Tout sur le Cloud Personnel : Travaillez, stockez, jouez et échangez... dans le nuage*. Dunod, 2013.
- [94] Pavan Pongle and Gurunath Chavan. A survey : Attacks on rpl and 6lowpan in iot. In *2015 International conference on pervasive computing (ICPC)*, pages 1–6. IEEE, 2015.
- [95] Swaroop Poudel. Internet of things : underlying technologies, interoperability, and threats to privacy and security. *Berkeley Technology Law Journal*, 31(2) :997–1022, 2016.
- [96] Peter Saint-Andre et al. Extensible messaging and presence protocol (xmpp) : Core. 2004.

-
- [97] Imad Saleh. Les enjeux et les défis de l'internet des objets (ido). *Internet des objets*, 1(1) :5, 2017.
- [98] Imad Saleh, Mehdi Ammi, and Samuel Szoniecky. *Challenges of the Internet of Things : Technique, Use, Ethics*. John Wiley & Sons, 2018.
- [99] Farzad Samie, Lars Bauer, and Jörg Henkel. Iot technologies for embedded computing : A survey. In *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS)*, pages 1–10. IEEE, 2016.
- [100] Gérald Santucci. The internet of things : Between the revolution of the internet and the metamorphosis of objects. *Vision and Challenges for Realising the Internet of Things*, pages 11–24, 2010.
- [101] Udit Satija, Barathram Ramkumar, and M Sabarimalai Manikandan. Real-time signal quality-aware ecg telemetry system for iot-based health care monitoring. *IEEE Internet of Things Journal*, 4(3) :815–823, 2017.
- [102] Jun-Oh Seo and Chul-Won Kim. Design and implementation of realtime things control system using mqtt and websocket in iot environment. *The Journal of the Korea institute of electronic communication sciences*, 13(3) :517–524, 2018.
- [103] Sajjad Hussain Shah and Ilyas Yaqoob. A survey : Internet of things (iot) technologies, applications and challenges. In *2016 IEEE Smart Energy Grid Engineering (SEGE)*, pages 381–385. IEEE, 2016.
- [104] Zach Shelby, Klaus Hartke, and Carsten Bormann. The constrained application protocol (coap). 2014.
- [105] John Soldatos, Martin Serrano, and Manfred Hauswirth. Convergence of utility computing with the internet-of-things. In *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 874–879. IEEE, 2012.
- [106] Moataz Soliman, Tobi Abiodun, Tarek Hamouda, Jiehan Zhou, and Chung-Horng Lung. Smart home : Integrating internet of things with web services and cloud computing. In

- 2013 IEEE 5th international conference on cloud computing technology and science, volume 2, pages 317–320. IEEE, 2013.
- [107] Benjamin K Sovacool and Dylan D Furszyfer Del Rio. Smart home technologies in europe : A critical review of concepts, benefits, risks and policies. *Renewable and Sustainable Energy Reviews*, 120 :109663, 2020.
- [108] Stamatis Stamatiadis, Dimitris Taskos, Eleftheria Tsadila, Calliopi Christofides, Christos Tsadilas, and James S Schepers. Comparison of passive and active canopy sensors for the estimation of vine biomass production. *Precision Agriculture*, 11(3) :306–315, 2010.
- [109] Chuan-Jun Su and Chia-Ying Wu. Jade implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring. *Applied Soft Computing*, 11(1) :315–325, 2011.
- [110] Hui Suo, Jiafu Wan, Caifeng Zou, and Jianqi Liu. Security in the internet of things : a review. In *2012 international conference on computer science and electronics engineering*, volume 3, pages 648–651. IEEE, 2012.
- [111] Shu Tang, Dennis R Sheldon, Charles M Eastman, Pardis Pishdad-Bozorgi, and Xinghua Gao. A review of building information modeling (bim) and the internet of things (iot) devices integration : Present status and future trends. *Automation in Construction*, 101 :127–139, 2019.
- [112] Marco Terán, Juan Aranda, Henry Carrillo, Diego Mendez, and Carlos Parra. Iot-based system for indoor location using bluetooth low energy. In *2017 IEEE Colombian Conference on Communications and Computing (COLCOM)*, pages 1–6. IEEE, 2017.
- [113] Suraj Tonage, Sandhya Yemul, Rajendra Jare, and Veena Patki. Iot based home automation system using nodemcu esp8266 module. *International Journal of Advance Research and Development*, 2018.
- [114] Ishaq Unwala and Jiang Lu. Iot protocols : Z-wave and thread. In *International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRSCE)*, volume 3, pages 355–359. 2017.

- [115] Ishaq Unwala, Zafar Taqvi, and Jiang Lu. Iot security : Zwave and thread. In *2018 IEEE Green Technologies Conference (GreenTech)*, pages 176–182. IEEE, 2018.
- [116] Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. Long-range iot technologies : The dawn of lora™. In *Future access enablers of ubiquitous and intelligent infrastructures*, pages 51–58. Springer, 2015.
- [117] Benny Vejlgaard, Mads Lauridsen, Huan Nguyen, István Z Kovács, Preben Mogensen, and Mads Sorensen. Coverage and capacity analysis of sigfox, lora, gprs, and nb-iot. In *2017 IEEE 85th vehicular technology conference (VTC Spring)*, pages 1–5. IEEE, 2017.
- [118] Ovidiu Vermesan, Arne Bröring, Elias Tragos, Martin Serrano, Davide Bacciu, Stefano Chessa, Claudio Gallicchio, Alessio Micheli, Mauro Dragone, Alessandro Saffiotti, et al. Internet of robotic things : converging sensing/actuating, hypoconnectivity, artificial intelligence and iot platforms. 2017.
- [119] Jiafu Wan, Hu Cai, and Keliang Zhou. Industrie 4.0 : enabling technologies. In *Proceedings of 2015 international conference on intelligent computing and internet of things*, pages 135–140. IEEE, 2015.
- [120] Jochen Wirtz, Paul G Patterson, Werner H Kunz, Thorsten Gruber, Vinh Nhat Lu, Stefanie Paluch, and Antje Martins. Brave new world : service robots in the frontline. *Journal of Service Management*, 2018.
- [121] Shintaro Yamamoto, Shinsuke Matsumoto, and Masahide Nakamura. Using cloud technologies for large-scale house data in smart city. In *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, pages 141–148. IEEE, 2012.
- [122] Muneer Bani Yassein, Mohammed Q Shatnawi, Shadi Aljwarneh, and Razan Al-Hatmi. Internet of things : Survey and open issues of mqtt protocol. In *2017 International Conference on Engineering & MIS (ICEMIS)*, pages 1–6. IEEE, 2017.
- [123] Keiichi Yasumoto, Hirozumi Yamaguchi, and Hiroshi Shigeno. Survey of real-time processing technologies of iot data streams. *Journal of Information Processing*, 24(2) :195–202, 2016.

- [124] Tetsuya Yokotani and Yuya Sasaki. Comparison with http and mqtt on required network resources for iot. In *2016 international conference on control, electronics, renewable energy and communications (ICCEREC)*, pages 1–6. IEEE, 2016.
- [125] Mingchuan Zhang, Haixia Zhao, Ruijuan Zheng, Qingtao Wu, and Wangyang Wei. Cognitive internet of things : concepts and application example. *International Journal of Computer Science Issues (IJCSI)*, 9(6) :151, 2012.
- [126] Kai Zhao and Lina Ge. A survey on the internet of things security. In *2013 Ninth international conference on computational intelligence and security*, pages 663–667. IEEE, 2013.
- [127] Lin Zhao and Zhihao Peng. Analysis and design of a context-aware smart home system. In *2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, pages 716–719. IEEE, 2020.
- [128] Liang Zhou, Dan Wu, Jianxin Chen, and Zhenjiang Dong. When computation hugs intelligence : Content-aware data processing for industrial iot. *IEEE Internet of Things Journal*, 5(3) :1657–1666, 2017.
- [129] Meftah Zouai, Okba Kazar, Guadalupe Ortiz Bellot, Belgacem Haba, Nadia Kabachi, and M Krishnamurthy. Ambiance intelligence approach using iot and multi-agent system. *International Journal of Distributed Systems and Technologies (IJDST)*, 10(1) :37–55, 2019.
- [130] Meftah Zouai, Okba Kazar, Belgacem Haba, Guadalupe Ortiz, and Nadia Kabachi. New approach using an iot robot to oversight the smart home environment. 2019.
- [131] Meftah Zouai, Okba Kazar, Belgacem Haba, and Hamza Saouli. Smart house simulation based multi-agent system and internet of things. In *2017 International Conference on Mathematics and Information Technology (ICMIT)*, pages 201–203. IEEE, 2017.
- [132] Meftah Zouai, Okba Kazar, Belgacem Haba, Hamza Saouli, and Hind Benfenati. Iot approach using multi-agent system for ambient intelligence. *International Journal of Software Engineering and Its Applications*, 11(9) :15–32, 2017.

- [133] Evi Zouganeli and Inge Einar Svinnset. Connected objects and the internet of things—a paradigm shift. In *2009 International Conference on Photonics in Switching*, pages 1–4. IEEE, 2009.