A secure design of WoT services for smart cities

Saad El Jaouhari

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THESE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPERIEURE MINES-TELECOM ATLANTIQUE
BRETAGNE PAYS DE LA LOIRE - IMT ATLANTIQUE
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par

« Saad EL JAOUHARI »

« A secure design of WoT services for smart cities »

Thèse présentée et soutenue à Cesson Sévigné, le « 13 décembre 2018 »
Unité de recherche : IRISA/D2/OCIF
Thèse N° : 2018IMTA0120

Rapporteurs avant soutenance :

Maryline Laurent	Professeure, IMT Telecom Sud Paris
Joachim Posegga	Professeur, Université de Passau

Composition du Jury :

Présidente :	Valérie Issarny	Directrice de Recherche, INRIA Paris
Examinateurs :	Farid Naït-Abdeslam	Professeur, Université Paris Descartes
	Tayeb Lemlouma	Maître de Conférences HDR, Université Rennes 1
	Maryline Laurent	Professeure, IMT Telecom Sud Paris
	Joachim Posegga	Professeur, Université de Passau
Directeur de thèse :	Jean-Marie Bonnin	Professeur, IMT Atlantique
Encadrant de thèse :	Ahmed Bouabdallah	Maître de conférences, IMT Atlantique

Invité
Emmanuel Cordonnier — Directeur e-Health, BCom

I would like to dedicate this thesis to my family who supported me all along this thesis.

Acknowledgements

First, I would like to express my gratitude and thanks to my thesis director, Professor Jean-Marie Bonnin and my supervisor, Dr. Ahmed Bouabdallah for their guidances all along this thesis. They did not save any effort to put their skills, experiences, and expertise in order to have the best results. The quality of the contributions and the thesis would have not been possible without the valuable discussions we had.

I wish also to thank my colleagues from the OCIF team in particular, and all the researchers in the SRCD department, for all the debates, discussions, remarks and guidances especially during the OCIF seminars. And a special thanks to my office colleagues: Francois, Xavier, Renzo, Samy, Qipeng, ..., for all the good time spent together and the rich discussions that we had.

Furthermore, I would like to express my sincere gratitudes to the jury members. Their insightful questions and encouraging remarks on my work made my thesis defense a lifelong memory as well as a learning process. It is my utmost honor to have all these experts reviewing my work.

I would like to sincere thanks and love my precious parents Malika and Ahmed, my brother Youssef, my sister Manal, my wife Zineb, and my best friends Youssef, Anas, Mouad, Walis and Alaeddine for their support in the good and the bad, no matter what decisions I have made.

Last but not the least, I would like to thank all those who helped me during this thesis and that I had the privilege to encounter.

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Abstract

The richness and the versatility of WebRTC, a new peer-to-peer, real-time and browser-based communication technology, allowed the imagination of new and innovative services. In this thesis, we analyzed the capabilities required to allow a participant in a WebRTC session to access the smart Things belonging to his own environment as well as those of any other participant in the same session. The access to such environment, which we call "Smart Space (SS)", can be either passive, for example by monitoring the contextual information provided by the sensors, or active by requesting the execution of commands by the actuators, or a mixture of both. This approach deserves attention because it allows to solve in an original way various issues such as allowing experts to remotely exercise and provide their expertises.

From a technical point of view the issue is not trivial because it requires a smooth and mastered articulation between two different technologies: WebRTC and the Internet of Things (IoT) / Web of Things (WoT). Hence, the first part of the problem studied in this thesis, consists in analyzing the possibilities of extending WebRTC capabilities with the WoT. So as to provide a secure and privacy-respectful access to the various smart objects located in the immediate environment of a participant to any other end-user involved in the same ongoing WebRTC session. Moreover, positioning our approach in the context of communication services operating in smart cities requires the ability to support a multiplicity of SSs, each with its own network and security policy. Hence, in order to allow a participant to access one of his own SSs or one of another participant (through a delegation of access process), it becomes necessary to dynamically identify, select, deploy, and enforce the SS's specific routing and security rules, so as to have an effective, fast and secure access. Therefore, the second part of the problem studied in this Ph.D. consists in defining an efficient management of the routing and security issues regarding the possibility of having multiple SSs distributed over the entire network. Thus, the main contributions of the thesis are the following:

The first one concerns the design, from scratch, of an architecture allowing a junction between WebRTC and the WoT. This architecture is then illustrated through a set of innovative use cases. The latter relies essentially on a gateway connecting the two technologies. Since WebRTC is natively secure, its analysis allowed us to propose a set of mechanisms to secure the link between the gateway and the WebRTC client together with the access control to the SS. The implementation of an experimental prototype validates the feasibility of our approach. This prototype is used to concretely explore the initial use cases in the health field as well as other applications associated with the smart home. The need for the health related use cases was confirmed later through

interviews with doctors. Finally, a privacy analysis of our system was proposed. An additional step was taken in order to provide a business model for one of the telemedicine use cases.

The second contribution proposes a new smart home architecture encompassing several services, among them the healthcare and the energy management. The first service is based on the previously presented contribution, and its main idea is to allow interactions between the user inside the smart home and a remote medical staff which can be a doctor for instance. Hence, bringing telemedicine services to the smart home such as the tele-consultation and the remote monitoring. The second one is proposed in order to monitor and manage the energy consumption using smart meters. The overall work targets the introduction of a real smart home, based in Aalborg University labs.

Finally, the third contribution introduces an SDN controller in order to manage the various SSs that can be involved in a WebRTC session. The main idea consists in allowing an end-user to own more than one SS while keeping their management simple and effective. The principle of our approach consists in centralizing the decisions concerning the management of the various SSs. Due to the fact that routing concerns are intimately intertwined with those of security, the SDN clearly appears as a promising tool to solve these issues. Therefore, an original architecture allowing an end user to dynamically manage the access to his different SSs is developed. A prototype illustrating our approach is implemented together with a set of experiments to evaluate the performance and security of the system. This approach is finally illustrated in the e-Health domain by demonstrating the possibility of managing an health infrastructure such as a Hospital.

Résumé

WebRTC est une technologie récente de communication qui permet d'établir des échanges multimédia conversationnels directement entre navigateurs. Sa richesse et sa versatilité laissent présager des opportunités inédites en termes de services de communication innovants. Nous nous intéressons dans cette thèse à des locuteurs dans un "*Smart Space*" (SS) défini comme un environnement centré-utilisateur instrumenté par un ensemble de capteurs et d'actionneurs connectés. Nous analysons les capacités nécessaires pour permettre à un participant d'une session WebRTC d'impliquer dans cette même session, les flux induits par les objets connectés appartenant au SS d'un utilisateur quelconque de la session. L'accès à un tel environnement, peut être soit passif via l'observation d'informations contextuelles fournies par les capteurs, soit actif par l'exécution d'une commande d'un actionneur, soit une combinaison des deux. Cette approche recèle un gisement de nombreux nouveaux usages. Nous limitons notre analyse à ceux concernant l'exercice distant d'une expertise et d'un savoir-faire.

D'un point de vue technique, il s'agit d'articuler de façon contrôlée les deux technologies différentes que sont WebRTC et l'Internet des objets (IoT)/Web des objets (WoT). Nous procédons dans un premier temps à une extension de WebRTC par WoT pour fournir à tout utilisateur d'une session WebRTC, un accès aux objets connectés du SS de tout autre participant à la session, en mettant l'accent sur la sécurisation de cet accès ainsi que sur sa conformité aux exigences de respect de la vie privée de l'utilisateur concerné. Le positionnement de notre approche dans le contexte des services de communication opérant dans les villes connectées, impose la prise en compte de SSs multiples et variés induisant chacun ses propres politiques de routage et de sécurité. Pour répondre à nos objectifs, il devient nécessaire au cours d'une session WebRTC, d'identifier, sélectionner, déployer et appliquer les règles de routage et de sécurité de façon à garantir un accès rapide et sécurisé aux différents SSs concernés. La seconde partie de notre travail consiste précisément à proposer une solution à cette problématique induite par des SSs distribués sur tout le réseau. Notre travail se décline en trois contributions principales.

La première concerne la conception d'une architecture opérant la jonction entre WebRTC et WoT. Sa définition s'appuie sur l'analyse préalable de plusieurs cas d'usage servant de fil conducteur. Cette architecture repose principalement sur une passerelle reliant ces deux environnements. De la sécurisation native de WebRTC, nous dérivons un certain nombre de mécanismes permettant de sécuriser le lien entre la passerelle et le client WebRTC ainsi que le contrôle d'accès au SS. Nous montrons la faisabilité de notre approche par le développement d'un prototype qui est mis à contribution pour

explorer les cas d'usage initiaux. Nous analysons pour finir la conformité de notre système au RGPD (Règlement Général sur la Protection des Données). La pertinence des cas d'usage médicaux a été mise à l'épreuve d'un regard médical neutre grâce à des entretiens pratiqués avec des médecins et un modèle économique permettant une valorisation commerciale de la télé-consultation médicale, a été proposé.

La seconde contribution étudie l'intégration à une maison connectée, de plusieurs types de services basés sur WoT. Nous illustrons notre approche en couplant expérimentalement et de façon sécurisée, les services de santé étudié précédemment au service de gestion de l'énergie consommée dans l'habitat. Les travaux présentés dans ce chapitre ont contribué au développement d'une maison connectée localisée dans un laboratoire de l'Université d'Aalborg qui préfigure la prochaine génération des maisons connectées.

La troisième contribution s'intéresse à la gestion des multiples SSs impliqués dans une session WebRTC. Le principe de l'approche proposée est classique et consiste à centraliser les décisions. Du fait de l'étroite imbrication entre les problématiques de routage et de sécurité, un contrôleur SDN apparaît comme une base de solution intéressante. Nous développons dans ce sens une architecture originale permettant à un utilisateur de gérer dynamiquement et de manière simple et efficace un accès sécurisé à ses différents SSs. Un prototype illustrant notre approche a été mis en œuvre et testé afin d'évaluer la performance et la sécurité du système. Nous illustrons finalement notre approche dans le domaine de la santé en démontrant son apport pour gérer une infrastructure de grande taille telle qu'un hôpital.

Contents

List of Figures

List of Tables

Acronyms

AAL	Ambient Assisted Living.
ABAC	Attribute-based access control.
AMI	Advanced Metering Infrastructure.
API	Application Programmable Interface.
ASIP	The Shared Healthcare Information Systems Agency.
BLE	Bluetooth Low Energy.
CoAP	Constrained Application Protocol.
DDoS	Distributed Denial of Service.
DoS	Denial of Service.
DTLS	Datagram Transport Layer Security.
ECG	Electro Cardiogram.
EHR	Electronic Health Records.
EMS	Energy Management System.
ERC	Emergency Relief Center.
GDPR	General Data Protection Regulation.
HAN	Home Area Network.
HEMS	Home Energy Management System.
HTTP	Hypertext Transfer Protocol.
HTTPS	HyperText Transfer Protocol Secure.
ICE	Interactive Connectivity Establishment.
ICT	Information Communication Technologies.
IdP	Identity Provider.
IETF	Internet Engineering Task Force.
IFTTT	If This Then That.
IoT	Internet of Things.
ISP	Internet service provider.
LAN	Local Area Network.
OFS	OpenFlow Switch.
ORTC	Object Real-Time Communication.
PAP	Policy Administration Point.
PDP	Policy Decision Point.
PEP	Policy Enforcement Point.
PIA	Privacy Impact Assessment.
PIP	Policy Information Point.
PKI	Public Key Infrastructure.
PoC	Proof of Concepts.
PRP	Policy Retrieval Point.
RBAC	Role-Based Access Control.
REST	Representational State Transfer.
RTP	Real-time Transport Protocol.

SCTP	Stream Control Transmission Protocol.
SDN	Software Defined Networking.
STUN	Session Traversal Utilities for NAT.
TCP	Transmission Control Protocol.
TURN	Traversal Using Relay NAT.
UDP	User Datagram Protocol.
UHC	Universal Health Coverage.
W3C	World Wide Web Consortium.
WebRTC	Web Real-Time Communication.
WHO	World Health Organization.
WoT	Web of Things.
WP29	Working Party 29.

Chapter 1 | Introduction

Contents

The aim of this chapter is to provide an overall view of the organization of the manuscript and the flow of ideas that led us to our major contributions. Thus, the chapter provides, first, a general introduction to the thesis with a simple discussion of the challenges. Then, presents a technical context with a brief explanation of the main technologies and notions, in order to understand the technical aspects behind the main challenges investigated during this thesis. These challenges are then solved through three main contributions. Finally, a roadmap of the whole dissertation is provided.

1.1 General introduction

IHS forecasts a massive growth of the IoT market, and in the number of the connected devices deployed all over the globe [5]. It estimates that there are currently more than 17.6 billion devices, and it will grow to 30.7 billion in 2020 and 75.4 billion in 2025. Moreover, and with this current advancement of the IoT, it is estimated that each user owns an average of roughly 3.3 connected device, according to [6]. These connected devices are characterized by being autonomous, and augmented with sensing/actuating, processing, storing, and networking capabilities. They are capable of sensing several types of information such as the weather condition, the traffic jam, the noise level in the surrounding area, the vital signs of the users, the luminosity level in the room, etc. Thus, the possibility of collecting contextual information can be helpful for the service providers in order to either improve their currently proposed services or to even propose new ones.

A particular interest was given to the case of the smart home, where obviously in the next generation ones, the IoT sensors and actuators will be deployed all over the home.

These sensors are part of the so called "Smart Space" (SS) [7, 8], which, briefly, represents a perfectly managed local network of smart objects accessible only through a gateway. Hence, the context awareness, particularly related to the owner of the home, can be helpful to provide a better quality of experience, and a better life quality.

An important aspect, in this case, is the one related to the healthcare services, especially for the dependent persons such as the elderlies, and particularly the in-home related ones. Undoubtedly, we are in an aging population, where, according to the statistics from the World Population Prospects in 2017 [9], elderlies aged 60 and over represents 13% of the world population, around 962 million, and it estimates that globally, population aged 60 or over is growing faster than all younger age groups. Thus, new solutions need to be proposed, in order to deal with the rising needs of the medical care and assistance, in particular for this specific category of dependent persons. Since, they are the most vulnerable to diseases and the ones that have more medical conditions that need to be taken care of.

On the other side, with the apparition of the new peer-to-peer and real-time communication technology called WebRTC, a whole new level of applications emerged. With the possibility to send and receive arbitrary data in real-time together with the audio/video flow, new ways for enhancing the multimedia flow with the contextual information of the user become possible. In this case, each of the communicating users, or WebRTC endpoint, has access to a set of smart objects, and is considered as a gateway to the smart space.

Consequently, this particular type of interactions require solving additional issues related to these surrounding smart objects, or as mentioned before the management of the smart space. Some of these issues are mainly related to:

- The collection of the data, usually from a heterogeneous environment of smart objects, communicating using different communication protocols, and may be present in different IoT architectures which are not usually interoperable.
- The discovery of these objects and their registration.
- The security and the privacy, since these sensors are collecting information directly related to the user, hence, directly impacting the privacy of the user.

Thus, all these challenges need to be solved in order to be able to provide reliable and trusted services to the users.

These issues become more convoluted when dealing with several smart objects, located in different smart spaces. Where, for instance, considering each room of a smart hospital as a smart space, with several medical sensors and actuators. Managing separately each smart space, can be an arduous task for the administrator, in particular regarding the security part. In a unified smart infrastructure, with several smart spaces, the security policies and the security mechanisms, need to be perfectly managed in order

to avoid any collision between the security rules and to avoid any security breach.

Thus, this thesis proposes solving some of these issues, in an innovative way, through three contributions, which will be highlighted in the next sections.

1.2 Technical context

In this thesis, the contributions are based mainly on two technologies, WebRTC and the WoT, and applied to a particular domain, which is the e-Health. Thus, this section introduces them briefly in order to be able to understand the context of our propositions. More details are provided in chapter 2.

First, WebRTC is a new real time communication technology, which provides the ability to exchange media directly between peers just with theirs browsers. It also means that there is no need to download additional plugins or softwares to establish a communication. Currently, many browsers such as Google Chrome or Mozilla Firefox natively support WebRTC. Beside the powerful communication capabilities provided by WebRTC, security was considered from the beginning in order to guarantee authentication, confidentiality and integrity in the system.

Secondly, the Web of Things (WoT), which can be seen as a specialization of the IoT. It mainly abstracts all the complexity of the connectivity part of the IoT, and grants more importance to the applications and services. The strength of the WoT is that it provides a standard and interoperable application layer based on Web. Thus, allowing more flexibility to the developers to develop their WoT/IoT based application and services. The main idea is that all the smart objects can communicate using a Web language (i.e. CoAP) through a REST API. This API can be either directly exposed by the smart object itself or via an intermediary (e.g. a gateway).

Finally, the e-health which is a health domain that takes advantages of the ICT in order to provide complementary services to the traditional ones. One of the main objective of the e-Health is to provide remote medical services to the patient, in particular in the medical deserts.

1.3 Motivations and challenges

The new vision of the smart city aims at providing better services to the citizens, by taking advantage of the current advancements of the information and communications technologies (ICT), the public resources, and the data which can be collected from all over the city using the IoT/WoT connected devices. Furthermore, the city can also be seen as a structure of services (i.e. Health, Energy, Traffic, Transportation, Public services, etc.), where each service have its own challenges and needs to be deeply analyzed.

In this thesis, a particular attention is given to the security challenges by designing and implementing secure services. From our perspective, the main building block toward

providing such services is the design of a secure smart space. Mainly, the targeted service will take advantage of the ability of the smart space to manage multiple IoT connected devices. For instance, a smart home can be considered as a smart space. In order to provide a smart energy management service (firstly in the smart home and then on a larger scale of a smart city), it can take advantage of the IoT data collected securely from the smart space in order to interact with the smart grid and with the energy utility, with the aim of optimizing the energy consumption and cost. Thus, a first step to introduce the challenges is to define the notion of the smart space.

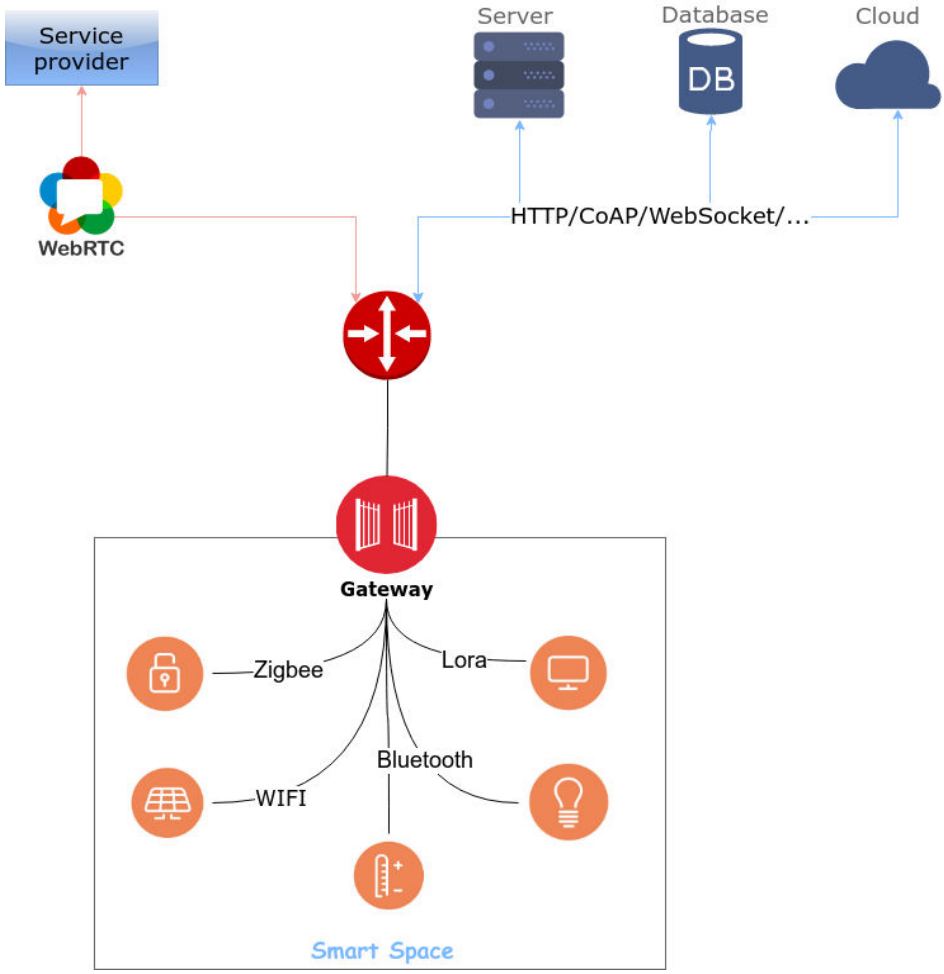


Figure 1.1 – Definition of a smart space (SS)

As shown in Fig. 1.1, a smart space (SS) is defined as a user-centric set of heterogeneous smart objects (SO)s, that communicate using different communication protocols, such as Bluetooth, Zigbee, WIFI, and so on, and which are accessible through a single gateway. In this thesis, only the client-server IoT/WoT topology is considered, other works may consider managing also mesh topologies, edge computing and fog computing. The gateway is directly connected to a network equipment, which can be a router, a box, or an OpenFlow switch as we will see later, and which main role is to collect the data from the different SOs, eventually actuating them, and to perform an access control regarding the different requests. Moreover, the interactions with this SS are secured, by first authenticating the users requesting the access to the SS's resources.

Then by controlling the access, so that only the legitimate users are allowed to access their corresponding resources. Finally, by encrypting the interaction with the SS in order to guarantee the confidentiality and the integrity of the data.

The data collected from the SS can be delivered using several types of protocols depending on the targeted service.

The first type of services are the ones that do not require a real-time transfer of the data. It uses mainly the traditional web protocols to transport the data flows in a secure way, such as as HTTPS, WebSocket Secure, or CoAP-DTLS. This kind of services can be either directed to an entity (for instance the web browser of an administrator), toward servers and clouds for processing and management or toward databases for storage. These services are based on the traditional WoT architecture, which will be introduced later in section 2.2 of the chapter 2.

The second one requires instantaneous and real-time transfer of the data. We introduce an original solution to this issue by using the data channels provided by WebRTC. It opens the door to a whole range of new services coupling the WoT flows with the multimedia flows of WebRTC. Such a coupling takes its full sense in the user-centric view: the WoT flows may represent contextual information associated to the smart space of the user while the multimedia ones concern his conversational services. An interesting example can be the remote consultation service in e-Health, which needs both real-time data from the IoT sensors and a multimedia interaction between the doctor and the patient. It is mainly destined for an entity (a doctor for instance), but not limited to. The challenges consist in designing an architecture which allows to deploy and operate such services in a secure an privacy-respectful way. These challenges are solved in the architecture proposed in our first contribution, which will be introduced later in chapter 3.

Whereas, in real cases, both services can be present in the same infrastructure, for instance in the smart home. The infrastructure needs to be able to manage and provide both services depending on the need. Hence, a second contribution is proposed by designing a smart home architecture in chapter 4, where the energy part is managed using the first type of services, by sending the data to the Energy Management System (EMS) and also to the energy utility using the traditional channels, and the second type of services is used to manage the health and the wellbeing of the users inside the smart home, and to provide in-home healthcare services remotely.

Furthermore, these two previous contributions focus mainly on the management of a single smart space. In order to go toward the notion of the smart city, where several SSs distributed all over the network exist, we need to be able to manage all of them in an efficient way, especially regarding the security. Thus, this management of multiple smart spaces represents our third contribution in chapter 5, and it is based mainly on the Software Defined Networking (SDN) technology. A special focus was given to the

security management and the access control to these smart spaces.

1.4 Contributions

1.4.1 Coupling WebRTC and the Web of Things

First, we propose coupling the real-time communication capabilities of WebRTC and the Web of Things, in an innovative way, through a newly designed architecture. The main purpose of this contribution is to provide real-time communication services, enhanced with contextual data, from the environments of the communicating users. These contextual data can be collected for instance from the IoT sensors surrounding the user. As it is known, these sensors can gather an incredible amount of data for a wide range of applications, such as for weather, health, energy, traffic, home security, etc. The strength of this architecture is then illustrated in the domain of e-Health, through a set of use cases providing remote medical services to the patients, and in particular the disabled and elderly ones, in their homes. Moreover, the proposed architecture is enhanced with a security layer and a privacy analysis, since the users are very concerned about the security and the privacy of their data, and in particular the sensitive data such as the health related ones.

1.4.2 Toward next generation smart homes

In this contribution, we propose an implementation of the next generation smart homes, where various data coming from various IoT sensors (medical, wellbeing, energy, etc.) and home equipment (smart fridges, smart TV, etc.) need to be managed, visualized, and secured. This contribution will take into account only the energy and the health data in the smart home. However, future works are planed in order to deal with any kind of data in the smart home, in addition to the smart interactions with the home equipments such as with AI and machine learning. These data are securely collected using a centralized WoT gateway, located inside the smart home. For the e-Health part, a set of possible use cases are provided. It integrates the architecture provided in the first contribution (mentioned earlier as the second type of services). The main idea, is to be able to link the next generation smart homes with external medical entities in order to provide, first, quick intervention in case of detection of an abnormality, and also to be able to provide basic medical services, such as remote consultations with a doctor, for a particular health issue. This vision can be very promising, in particular in the rural areas, where they have difficulties to access the medical services. As for the energy part, the aim is to be able to collect the energy consumption inside the smart home, which can be either heat, water, gas, or electricity, and to be able to apply advanced algorithms in order to perfectly predict and manage the energy consumption. The proposition is based on the traditional WoT architecture (mentioned earlier as the first type of services), and uses smart meters data, together with the data collected from the smart plugs and linked to the previously mentioned gateway.

1.4.3 Secure access to multiple smart spaces using SDN

This contribution proposes a solution to manage multiple smart spaces, which can be duplicated and dispatched all over the network. Undoubtedly, managing just a single smart space presents several challenges, in particular the ones related to the network management and to the security. Since, first the exchanges with the smart space need to be protected from external threats, and secondly the smart space needs to be secured against the unauthorized access, both at the network layer through the firewalls, and at the application layer using an access control mechanism. Obviously, these issues become more convoluted when dealing with multiple smart spaces. In the vision of a smart city, managing the security of each smart space (i.e. smart home) separately, may be an arduous task for the administrator and may create security breaches, which can be for instance a conflicts between the security policies of two smart spaces. Thus, this contribution proposes to solve this issue by introducing an architecture based on a Software Defined Networking (SDN) controller for providing a secure and scalable access control to the different SSs of a smart infrastructure in a centralized way. It operates at both the network layer by dynamically configuring the network equipments attached to the smart space, and at the application layer, by providing a fine granular access control. The hospital infrastructure is chosen to illustrate the concept.

1.5 Thesis roadmap

This dissertation is composed of seven chapters, and an appendix part. It is organized as follows:

Chapter 1. Introduces the context of the thesis, the motivations behind, and presents the main challenges and issues that we solved with our contributions.

Chapter 2. Provides the general background regarding the two main technologies of this thesis, which are WebRTC and the Web of Things, together with an introduction of the e-Health domain, which will be the main application domain of all our contributions.

Chapter 3. Proposes the first contribution of this thesis, which is an architecture coupling WebRTC and the Web of Things. This chapter starts with the proposed architecture, followed by the main use case/scenarios, illustrating the interest in using this architecture. Together with the security layer and the current state of the implementation. And finally, a privacy analysis regarding one particular use case, "*Tele-consultations*", is conducted in order to identify the privacy breaches and to measure the potential risks.

Chapter 4. Provides the second contribution, regarding the application of the previously mentioned architectures in a real smart home. The chapter starts with the introduction of the notion of the Home Area Network (HAN), which encompasses several domains which are: entertainment, healthcare and wellbeing; energy management; security, safety and privacy;

and the interaction with the external providers and entities. A special focus is given to the energy and healthcare management. Then, a section dedicated to the architecture of the smart home, with the main communication technologies. Next, the main contributions toward the notion of the smart home are provided. And finally a performance analysis is conducted.

Chapter 5. Provides the third and final contribution, regarding the security management of several smart spaces using SDN. After an introduction, the main challenges and issues have been identified, in order to justify the use of SDN. Then, a theoretical architecture, with the different components is provided. Followed by the main proposition, together with the technical details regarding the implementation and the different access delegation methods. Next, the application of this solution to the e-Health domain is illustrated. And finally an analysis is conducted in order to evaluate this proposition.

Chapter 6. As a part of EIT digital doctoral school programs [10], which provides a training to acquire a mindset for Innovation and Entrepreneurship (I&E), this chapter was introduced. The main aims of this program, hence, this chapter as a result, is to link the academic field to the market, and to avoid keeping the result under the research shelves. For this purpose, a project called "SIRONA" is proposed. It is based on the remote consultation use case of the first contribution. Hence, a presentation of the added value of this project, and an analysis of the business model together with the market analysis are provided. A privacy, and data protection evaluation is conducted in order to push the final product to the market. Such requirement is mandatory in this kind of projects.

Chapter 7. Provides a conclusion of this thesis, and presents some future perspectives and directions in order to improve these contributions.

Chapter 8. The appendix part includes mainly the achievements, a security analysis of these two technologies used in this thesis, and the French resume of the thesis:

Appendix A. Presents the list of publications.

Appendix B. Analyzes the security in WebRTC, starting from the browser side to the security of the peer-to-peer exchanges, and passing by the security related to the service providers and the one related to the signaling process.

Appendix C. Is related to the main security mechanisms proposed for the Web of Things, starting from the security of the smart object [11], and surveying the security mechanisms regarding the identity management, data confidentiality and integrity, authorization, and access control [12].

Appendix D. Provides the French resume of the thesis.

Chapter 2 | Context of the work

Contents

This chapter introduces the concepts, the architectures and the protocols of the main technologies used in this thesis, together with the notion of e-Health, since it will be the main application domain of all our contributions. The first technology concerns a real-time and a peer-to-peer communication one called WebRTC (Web Real-Time Communication). And the second one concerns the Web of Things.

2.1 WebRTC

WebRTC [13], is a recent communication technology allowing the convergence between the telecommunication and the web worlds. It allows the establishment of a remote and peer-to-peer communication between the users just with their browsers. And provides the ability to send and/or receive different types of data in real-time, through WebRTC data channels, in parallel with multimedia flows. Currently, WebRTC is natively supported by most of the well known browsers to mention: Firefox, Chrome and Opera, in both their desktop and Android versions, Microsoft Edge (through a notion similar to WebRTC called ORTC (Object Real-Time Communication)[14], and also recently

with iOS 11 (Safari).

WebRTC is maintained and developed by both W3C (World Wide Web Consortium) and IETF (Internet Engineering Task Force). The IETF is responsible of the definition of the architecture and the associated protocols, while, the W3C is responsible of the specification and the development of the APIs that can be used by the developers of the web applications.

The technology is mainly used for audio/video real-time communication, visio-conferencing and the transfer of various types of data in a peer-to-peer fashion. The novelties and the advantages that WebRTC bring, with respect to the existing means of communication, are numerous, just to mention a few:

- Easy to use: since all that is needed is a compatible browser (Chrome, Mozilla, Opera, etc.). Moreover, it does not require any particular and additional software installation (neither plugins nor third party applications).
- Benefits from the advantages of the peer-to-peer communication, such as:
  - Low latency.
  - Good transmission quality (echo cancellation mechanisms, etc.).
  - Low network cost due to best-effort communications.
  - No intermediary, hence, less security threats.
- An open source solution.
- The security was taken into account from the beginning, by using several native mechanisms to ensure authentication, confidentiality and integrity.

Next, the WebRTC architecture will be presented. In the rest of the document, we will focus only on the real-time communication between two users and on the real-time exchange of data. The visio-conferencing system is out of the scope of the thesis.

2.1.1 Architecture

In order to establish a communication between two users, first, each user needs to have access to a compatible browser, either from a smartphone, a laptop or from a tablet. And secondly, at least one signaling server is required in order to initialize the communication, as we will explain later in the next section. Hence, depending on the administrative domain of the users, two types of topologies can be distinguished: Topology 1) if the two users are in the same administrative domain, only a single signaling server is required. Topology 2) if they are in two different ones, two signaling servers are required [15].

Topology 1:

In this case, both users are in the same administrative domain, hence, they can use the same signaling server in order to establish the communication, as shown in Fig. 2.1. The architecture is composed of the two users, with their respective compatible browsers, and the web server, which is responsible for providing the web application to the browsers and for the signaling. The main steps for establishing the communication are:

1. Both browsers download a JavaScript (JS) application (containing the logic of the application) from the Web Server (the signaling server), which allows the establishment of the communication.

2. The establishment of the session is based on an Offer/Answer procedure that allows the caller to contact the called party. The Offer request is transmitted in a first secure bidirectional WebSocket/HTTPS communication channel, connecting the caller's browser to the web server. The latter relays the Offer request to the called party in a second channel of the same type. And the same goes for the Answer of the called party. WebRTC does not impose a particular signaling protocol. Thus, the previous exchanges can be done using SIP (Session Initiation Protocol) or Jingle for instance.

3. Once the session is established, a secure, bidirectional, peer-to-peer and real-time communication channel is created enabling the exchange of multimedia and data between the communicating peers.

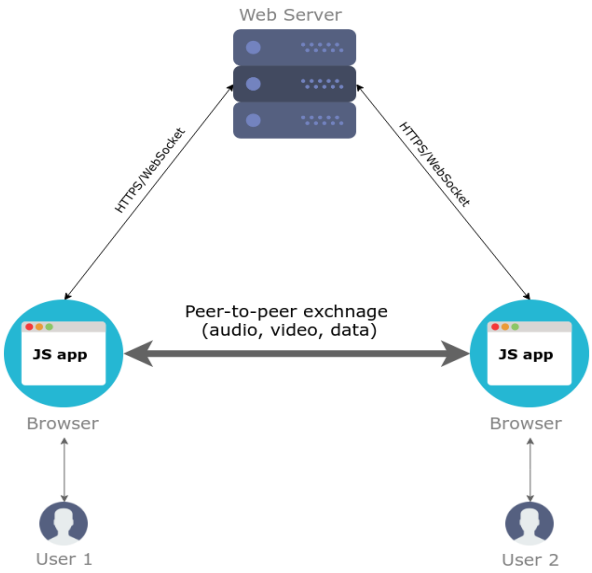


Figure 2.1 – WebRTC triangle topology

More information on the second topology can be found in [15].

2.1.2 Structure of a WebRTC session

This process generally allows the coordination of the communication between the participants, which includes: the initialization, the control and the termination of the

communication. The initialization of the communication allows the negotiation of the multimedia characteristics of the future communication and the transport addresses of the multimedia and/or data flow, following an offer/answer model. And also the allocation of the resources associated to this path. The control of the communication checks the stability and the smoothness of the ongoing communication. And finally, the termination of the communication in order to free the previously allocated resources for future use.

Moreover, it up to the developers to choose the most convenient signaling mechanism, which can be either proprietary or already existing (SIP, XMPP, Jingle, etc.). However, the chosen protocol should support the offer/answer model, which is necessary in order the exchange the signaling information though SDP (Session Description Protocol) [16] messages, as it will be explained next.

2.1.3 Offer/Answer process

The WebRTC communication between the browsers, requires first the negotiations of some session's related parameters, such as the audio and/or video codecs, the IP addresses, the authorizations, etc. As previously mentioned, these parameters are exchanged during the initialization phase of the session, as a part of the offer/answer process. The SDP, in this context, is the IETF standardized format for this negotiation [16].

The SDP is currently used in several well-known protocols such as SIP, HTTP, SMTP, etc. Thus, this format is used by WebRTC in order to exchange and specify the session parameters such as the origin the ID of the session, the media type (video, audio, etc.), the ports number, the transport protocol (RTP/UDP/IP, etc.), the supported codecs by the browser (G.711 audio, OPUS audio, VP8 video, H.264 video, etc.), and so on and so forth.

2.1.4 WebRTC's related protocols

The communication using WebRTC requires the collaboration of several types of protocols, to mention transport, communication and security protocols. The essential and the most important ones are RTP (Real-time Transport Protocol) [17], DTLS (Datagram Transport Layer Security)[18] [19] and SCTP (Stream Control Transmission Protocol) [20].

2.1.5 NAT traversal, STUN, TURN and ICE

A user cannot be directly connected to Internet without being behind a NAT or a firewall. Hence, in order to be able to connect any user to a WebRTC session, extra mechanisms and functions need to be introduced. Mainly, WebRTC should first be able to know the IP addresses of the communicating peers, to be able to choose the best path between them, using the gathered IP addresses, in order to establish the connexion, and

finally be able to provide alternative solution in case of failure of the previous approach.

Hence, WebRTC requires three IETF standards in order to manage the NAT traversal in these situation: 1) Session Traversal Utilities for NAT (STUN) [21], Traversal Using Relay NAT (TURN) [22], and Interactive Connectivity Establishment (ICE) [23].

2.1.6 WebRTC Architecture and APIs

The relevant parts of the WebRTC architecture are shown in Fig. 2.2, and explained in detail in [24]. It contains mainly:

- Multimedia (voice and video) engines in order to encode and decode the audio/video flows between the communicating peers. Different codecs and algorithms are implemented in order to be able to connect the two peers (since each peer support a set of codecs, but most of the time not all of them).
- The transport protocols as explained previously, in order to deliver the multimedia data: SRTP; P2P using ICE, STUN and TURN, together with the multiplexing mechanisms.
- The APIs for the web developers, which contain the main methods and functions for the creation of a WebRTC application.
- The APIs for the browser developers, which allow them to have a more granular and fine access to the features of the browser such as for the manipulation of the input/output for the media, the secure and peer-to-peer transport of data, the choice of the appropriate codecs, etc.

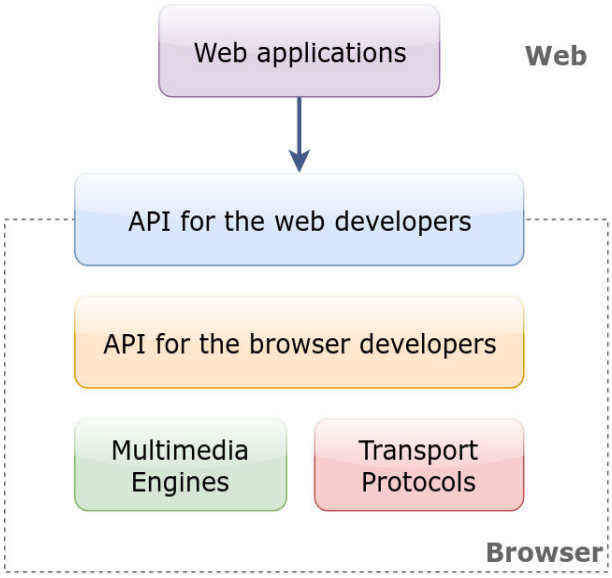


Figure 2.2 – Simplified WebRTC architecture

In this thesis, we are more interested in the API related to the web developers, or the Web API, in order to implement our own WebRTC application. The establishment of

a WebRTC session requires the use of three main categories of functions, or in other words, three main APIs:

1. Functions allowing the access to the input/output of the media streams (camera, microphone, etc.).
2. Functions related to the establishment of the multimedia session usually with a peer whose network characteristics are not necessary known (remote peer from another network).
3. Functions related to the transfer of different types of data during an ongoing multimedia session (images, files, etc.).

Thus, three APIs were defined by the W3C in order to manage the different functions, which are respectively: MediaStream, RTCPeerConnection and RTCDataChannel.

2.1.6.1 MediaStream

This API is responsible for accessing the input/output multimedia devices of the user. It presents the audio/video flow as a MediaStream object, and proposes a set of methods/functions in order to manipulate them. Each MediaStream can contain several tracks, a track being a source of media. The media flow is transported in a secure way using the SRTP protocol, and the key management of the encryption keys associated with SRTP is insured by DTLS. This API offers also the possibility of multiplexing several flows in a single SRTP connection.

The main function allowing the access of the multimedia devices is "getUserMedia()". Every time this function is used, an explicit authorization is requested by the browser.

2.1.6.2 RTCPeerConnection

This is the main API of WebRTC. Its main purpose is to establish the connection between the two communicating peers. This API is called using the "RTCPeerConnection()" function, and offers mainly, but not limited to, these features:

- Secure exchange of signaling messages (SDP) through a channel WebSocket/HTTPS.
- Implements the offer/answer method.
- Management of the codecs: selection, compression/decompression, etc.
- Management of NAT traversal with the ICE, STUN and TURN techniques.
- Bandwidth management.
- Echo cancellation and noise management.

2.1.6.3 RTCDataChannel

This API allows the browser to exchange various types of data in real-time bidirectionally. These data are transported using SCTP encapsulated in DTLS, this encapsulation combined with the ICE method guarantees: the confidentiality of the data; the authentication of the source of the data; the integrity of the data, and the NAT traversal.

The SCTP protocol supports the transport of different types of flows (media or/and data) in duplex, and provides two modes of transport: a reliable mode, with the same concept as TCP, and an non-reliable mode, with the same concept as UDP. The channel created by this API is characterized by its low latency and its security. The function in charge of the creation of theses channels (Data Channels) is called "createdatachannel()". The first call of this function allows the creation of an SCTP association and a Data Channel. Later, every call of this function will imply only the creation of a DataChannel which will be then multiplexed in the SCTP association already existing.

2.1.7 The security in WebRTC

Communication protocols, in particular the ones dedicated to the real time exchanges, are required to guarantee the security of the system and their users. Especially regarding the authentication, the integrity and the confidentiality.

During the development of WebRTC, the IETF and the W3C identified several security challenges, in particular:

- A well know issue in the web, and WebRTC is not an exception, is that the JavaScript application can be downloaded from any website, and most of the time without the consent of the user.
- The user must be able to control the access to his/her input-output devices (in particular the multimedia related ones such as the camera and the microphone).
- The user should not receive an important signaling traffic as long as he/she did not accept the incoming call.
- The confidentiality and the authentication must be guaranteed during all the communication.
- The private information of the user such as the identity, geographic location, and so on, must not be revealed without his agreement, (and also to provide the possibility of making anonymous calls).

Thus, the security of WebRTC articulates around five fundamental points [25][26]:

- The security depends on the level of trust granted/accorded to the browser and to the supplier of the WebRTC services.
- The JavaScript application must be executed inside the Sandbox of the browser.

- The whole communication must be encrypted in order to protect it from the external threats.
- Getting the explicit consent of the users is a requirement, and it becomes even an obligation with the new GDPR law.
- The authentication, the identity verification and the origin checks are also required.

Hence, following this logic, the security in WebRTC can be divided into four parts representing the main aspects to be secured in order to guarantee the security of the overall system:

1. The browser.
2. The web server providing the WebRTC application, or the service provider.
3. The signaling.
4. The security of the peer-to-peer exchanges.

The details of the security analysis of each part of the WebRTC system are provided in the appendix B.

2.2 Web of Things

2.2.1 IoT in a nutshell

In the last few years, the notion of the Internet of Things (IoT) emerged, and became one of the most promising technologies, with a wide range of applications, in almost all the fields. While the term is now more and more broadly used, there is no common definition or understanding of what the IoT is really today. However, and up to our understanding, the IoT refers to a network of interconnected people and objects, these objects can be anything surrounding us, thus, the term "Thing". This network of Things can revolutionize every area in our live, toward the notion of the smart city.

These Things have usually an embedded systems, and characterized, by theirs capabilities to communicate, and by their constraints such as their limited processing and storage capabilities, their limited battery storage, and the short range of their communication. Moreover, these physical objects or Things, and also known as "Smart objects" (SO) (which will be used all along the thesis), have the capability to interact either directly with humans, in a Machine-to-Human system, or to communicate with other devices, such as in a Machine-to-Machine system. Hence, opening an opportunity for a large number of novel applications in all the domains of our live, starting from a simple sensed data from a cheap sensor in the market, to a complex system in the next generation industry 4.0.

The application fields for the IoT technologies are as numerous as they are diverse. The most prominent areas of application include, the healthcare, where more and more

well-being and medical devices are introduced into the market. These medical devices can be used to collect vital signs and to monitor patients either in the hospitals or in their homes. The smart industry, which is going toward the "Industry 4.0", where the development of intelligent production systems and connected production sites is often discussed, and where the IoT plays the leading role. The smart home or building, where intelligent thermostats, lighting and security systems are receiving a lot of attention, while the smart energy systems uses the smart plugs and the smart meters in order to collect heat, electricity, gas and water data and consumption. And last but not least, the Intelligent Transportation Systems (ITS) solutions which includes for instance the smart and autonomous vehicles ables to sense the surrounding environment and adapt to it, in order to provide a better and safer experience to the users.

As mentioned earlier, an important feature of the smart objects is their ability to communicate with each others and with the humans. For this purpose, many wireless communication technologies and protocols emerged. Two main communication categories can be distinguished in the IoT. The short range technologies such as 6LoWPAN, Zigbee, Bluetooth, Z-Wave, NFC, RFID, and the long range ones such as SigFox, LoRa, and Cellular (considered also as Low Power Wide Area Networks (LPWAN)). In this thesis, and as a proof of concept, only the short range ones were implemented. The proposed concepts remains the same even with the long range communication protocols. However, further analysis need to be undertaken, in particular regarding the security and the privacy related issues.

Several architectures are currently proposed for the IoT, just to mention oneM2M[27], IoT-A[28], AIOTI [29], etc., but currently, only one architecture is fully implemented and open to the public which is oneM2M. However, one of the drawbacks of the oneM2M architecture is that it focuses only on the M2M interaction and there is few involvements of the end user. Additionally, none of those architectures can interconnect with the other, hence, creating silos of users and interoperability issues.

However, the novelty of the IoT is not in the functional capability of a smart object, since today many embedded systems are already connected to the Internet, but in the expected number of billions or even trillions of smart objects, which will eventually creates novel technical and societal challenges which did not exist before. IHS forecasts a massive growth of the IoT market, and in the number of deployed devices all over the globe [5]. It estimates that there are currently more than 17.6 billion devices, and it will grow to 30.7 billion in 2020 and 75.4 billion in 2025. Some of these challenges are: Security and Privacy; Dynamic discovery and Management, Interoperability and Silos of users, which can be at the architectural level as mentioned earlier, or at the communication protocol level, since most of them are not interoperable; etc.

In order to deal with these particular challenges, and since the IoT deals mainly with the connectivity aspects of the smart objects, and does not accord big importance to the service and application layer, a new paradigm called "Web of Things (WoT)" appeared. It provides an application layer that simplify the creation of the IoT applications, and

aims to enable real-world objects to be part of the World Wide Web. This objective can be achieved on top of connecting all these objects together at a global scale in IoT.

2.2.2 WoT Definition

The Web of Things (WoT) can be seen as a specialization of the Internet of Things (IoT), where, in one hand, all the complexity of the connectivity part of the smart objects is abstracted, and on the other hand, it provides a standard application layer based on Web standards to simplify the creation of IoT applications. In the IoT, the one-application one-communication protocol overwhelms, which creates silos of users and which restricts the harnessing of the full potential of the IoT. One of the main interest of using the WoT instead of the IoT is the different advantages that it offers. Just to mention, the simplicity of development, the loose coupling, since HTTP is loosely coupled by design, and the openness of the standards. The idea is that all the smart objects can communicate using a Web language (i.e. HTTP) by exposing a REST API. This API can be either present in the smart object itself or in an intermediary that can act on behalf of the smart object to expose the Web API [30]. This has become possible with the improvement of the embedded systems fortunately.

Hence, three main solutions can be distinguished in order to enable the constrained and the non-constrained devices to access the Internet and to provide their services:

Solution 1: by directly implementing the WoT API on the smart object itself. Technically, by running a tiny Web server on the smart object. Hence, the smart object can communicate via HTTP, and provides an access to its resources through a REST API. Nowadays, we can run tiny servers inside the constrained devices such as lighttpd [1], Nginx [2] and MiniWeb [3]. However, this solution is not meant for devices which are not battery-powered, since running a server and/or using WIFI or Ethernet on the device requires extra energy.

Solution 2: is mainly for the constrained devices that are battery-powered and that rely on low-energy protocols to communicate. Instead of having the WoT API on the device, it will be hosted on an intermediary that can expose the device's functionalities (such as a WIFI router). These intermediaries are called *gateways* or *application gateways*, and they are usually not constrained. On one side, they can talk with smart objects using a non-Web protocols such as Bluetooth, Zigbee, etc., and on the other side, they expose the smart object's WoT API via a REST API. In addition to just converting from one protocol to another, these gateways can also add a security layer, store data temporary, expose the semantic description of the smart objects, and so on and so forth.

[1] https://www.lighttpd.net/
[2] https://www.nginx.com/
[3] http://miniweb.sourceforge.net

Solution 3: is by using the cloud. This solution is an extension of the gateway solution, where instead of putting the WoT API on a simple intermediary, the gateway is a remote cloud server. This solution is used in the case where managing a large quantity of devices and data, and where a more powerful and bigger database is needed.

2.2.3 Differences between IoT and WoT

In order to justify the need of the WoT, and to clarify the differences between the notion of the IoT and the notion of the WoT [30], a list of arguments is provided below:

- No more One Application - One protocol: in the current IoT, for each smart object, usually a dedicated application that can talk exclusively with this smart object (or with a small set of smart objects) is needed, and usually using a specific communication protocol. The inconvenient of this approach is that in one hand, the user will need to deal with a different application for each smart object, and on the other hand it creates silos of users and smart objects. However, with the idea of the WoT, a single Web application can control all of these smart objects, since each smart object is exposing a WoT API.

- Easier to program: in the IoT many solutions and protocols are proposed to connect the various smart objects. Hence, learning the specification of each protocol is an arduous task, especially for amateurs that want to connect various objects (case of the smart home). However, with the WoT, building a Web application to connect various devices becomes easier and faster since most of the developers have already knowledge about Web protocols. Moreover, if all the devices could offer a WoT API, developers can use the same programming model to interact with any of them, which can be very interesting in the case of smart home, smart buildings and smart cities.

- Open and extensible standards: in the IoT communications protocols and standards are continuously proposed by different entities. However, not all the protocols are publicly available for the community. Moreover, at any time breaking changes can be introduced to the protocol which require at least a firmware update if not also changing the hardware. In the other hand, the open and free Web Standards have significantly evolved. Hence, there is nearly no risk that they would change overnight.

- Fast and easy to deploy, maintain, and integrate: in the IoT, if a single application layer is needed, significant effort and investment is required to write custom converters for each new device or application that needs to be integrated in the system. Moreover, adding the previous limitation, where the protocol is changed or upgraded, adds more complexity. In the WoT, there is no need to maintain or upgrade the Web.

- Loose coupling between elements: in the IoT, a very tight coupling between the device and the corresponding application in the network is present. Moreover, the different interactions that the device can have are statically planed in advance.

Hence, does not leave much space for ad hoc, unplanned interactions and re-purposing of services into new use cases. For the WoT, HTTP is loosely coupled by design. One of the objectives of the WoT is to be able to add any device into the system and to be able to talk with it without any upgrade.

- Security & Privacy mechanisms: which is one of the main issues either in the IoT nor the WoT, in order to build a real-world application that can be safe for the users and that protect their data from any threat. The problem in the IoT is that the security mechanisms are sometime build from scratch specially for a specific device or application and sometimes does not even exist. Moreover, even those mechanisms are not always seriously tested and may have some really serious flaws (weak keys, weak passwords, weak key exchange algorithms, static keys or keys that are publicly exchanged and so on and so forth). In the case of the WoT, application layer security on the Web have matured enough to resist to several attacks and considerable progress has been made to provide reliable security mechanisms.

Next, a use case is provided in order to illustrate the interest in using the WoT, in the context of the smart home.

2.2.4 Clarification use case

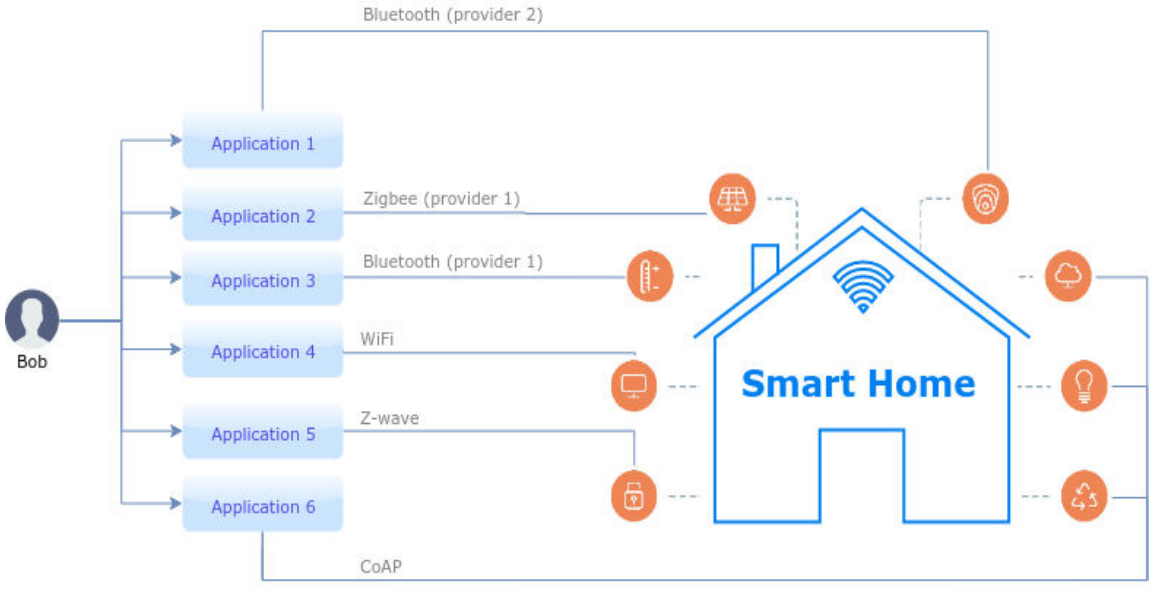


Figure 2.3 – Smart Home with IoT

Through this use case, the added value of the WoT is introduced. Imagining a simple case of a smart home where every smart object (lights, TVs, fridge, thermometers, etc.) belongs to a different manufacturer (Samsung, LG, Apple, etc.). As shown in Figure 2.3, in the IoT case, each smart object is controlled via a dedicated application that was provided by the respective manufacturer. Hence, the first issue is that Alice, the owner of the home, will have to deal with dozens of applications that are not interoperable with each other. The second issue is the integration of all of these devices in a single

application. This is clearly an arduous task, since the developer needs to be able to integrate devices that uses different and usually incompatible protocols.

However, in the case of the WoT, if each smart object offers a simple WoT API, building a centralized Web application becomes as easy as building any Web service. In the case of the smart home, the centralized application, which controls all the smart objects in the home, can be simply a Web mashup, as explained in Figure 2.4. With this solution, the time and the cost needed to build such system will decrease significantly. Moreover, it will also minimize the effort required to maintain the system each time a device or service is added, removed, or updated. Also the Figure 2.4, shows that by using the different intermediaries (Cloud or simple gateways) interconnecting any kind of device becomes easy.

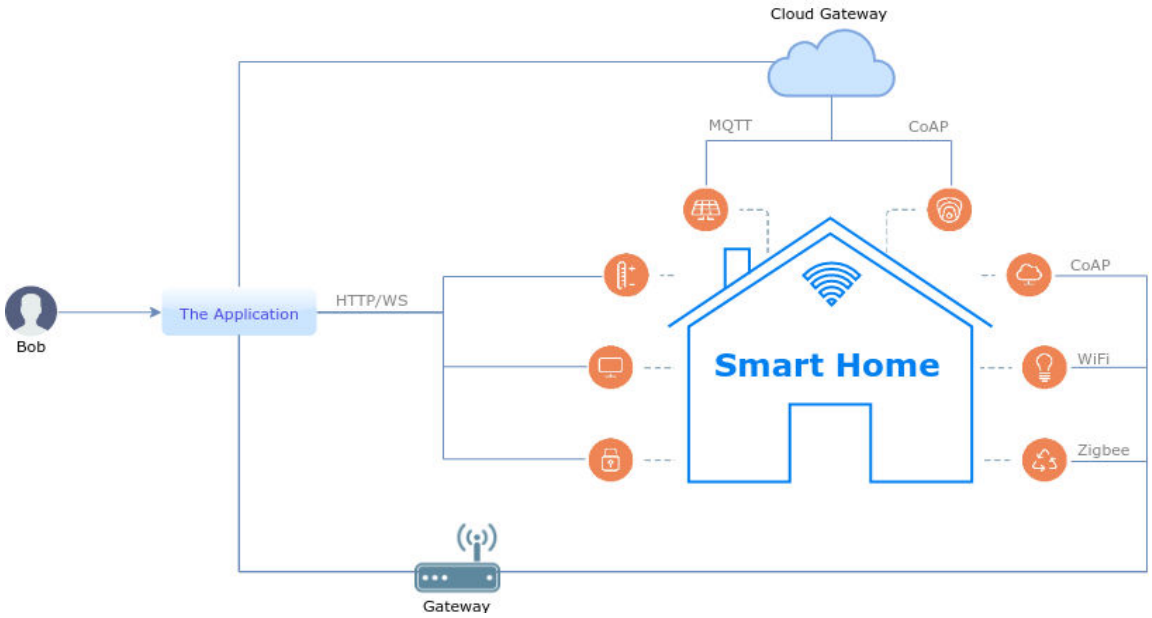


Figure 2.4 – Smart Home with WoT

2.2.5 WoT Interest group

In the W3C, there is an interest group called "Web of Things Interest Group" that discusses the possibility of defining platform independent APIs for application developers, in order to have interoperability between the different IoT platforms, by taking advantage of the Web. They also explore the potentials of using the Web standards (JavaScript as a default language, JSON for data encoding, HTTP, WS etc.) in order to communicate with the smart objects in the IoT platforms and to be able to efficiently control their data. All aspects that are related to the security and the privacy are also in the main interest of this working group, since particular treatment needs to be accorded to these constrained objects [1].

The working group considers the smart objects as a virtual presentation of physical or abstract entities (since a SO can also be people, places, ideas (event), etc.). Each smart object can have one or more virtual presentation (called also Avatars). Avatars

have identities and URIs and accessible via Web technologies [1].

The WoT architecture should respect several requirements such as the flexibility, to be able to map all kinds of physical devices; compatibility, to be able to bridge to the current legacy its solutions and standards; and finally, to be able to provide safety and security functionalities. W3C proposes a WoT architecture, where the **WoT Servient** is the main component, as explained in [1] .

This WoT Servient is a software stack that implements the WoT building blocks, and which has the following structure [4]:

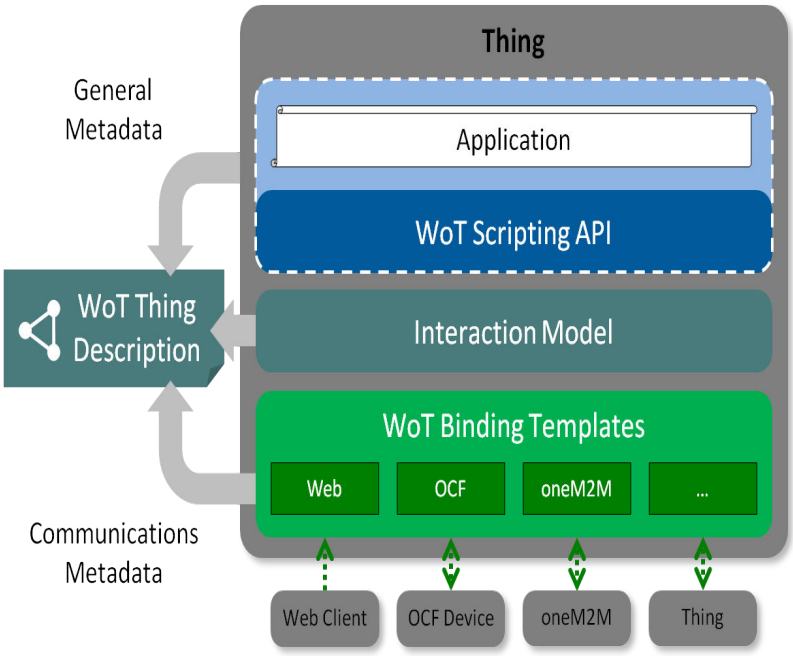


Figure 2.5 – Functional Architecture of WoT Servient [4] [1]

- Application Script: represents the applications running on a Servient.
- Security Metadata: defines how the scripts must be isolated, and includes keying material or certificates to authenticate the Things.
- WoT Scripting API: aims at having a runtime system for IoT application similar to a Web browser, instead of having the device logic implemented in the firmware, which eases, hence, the IoT application development.
- WoT Runtime: interface in one side with the Protocol binding in order to access the remote Things and on the other side with the system API in order to access local hardware and properties of the Thing means for communication.
- Thing Description: containing mainly the metadata about the Thing such as the interaction model, data model, the communication, and the security mechanisms.

[4]Copyright ©, 2017-2018 W3C ®(MIT, ERCIM, Keio, Beihang). W3C liability, trademark and permissive document license rules apply. Status: "Draft". http://www.w3.org/Consortium/Legal/2015/doc-license

It enables in one side the machine-to-machine interactions, and also can serve as a common and uniform format for the developers to interact with the IoT devices and to access their data.

- Protocol Binding: to convert the interactions with the Thing in accordance with the used communication protocol (Zigbee, CoAP, MQTT, etc.).
- System API: allows the access to local hardware or system services.
- Binding Templates: In order to enable the interoperability with the different IoT platforms, informal collection of communication metadata blueprints were collected. These metadata blueprints explain mainly how to interact with different IoT Platforms.

This architecture is still an ongoing work, and the document is updated regularly. In our implementations, only some of the components have been implemented such as the protocol binding and the application script. In our contribution, only the logic of the WoT was adopted, and not the full architecture, since, firstly, the WoT architecture was and still not stable, and secondly, the security mechanisms are still missing in the current version of the architecture. Thus, a derived and simplified version of the architecture with a security layer was implemented in our WoT gateway.

2.2.6 Conclusion

In our thesis, the WoT is seen as a very interesting alternative to the IoT, where developing an application able to control several kinds of devices becomes easy using the Web protocols. And since WebRTC is also a Web technology. By having a centralized Web application with both WebRTC APIs and WoT APIs, an innovative architecture with novel use cases can be achieved. However, and as we mentioned before, the WoT is only a specialization of the IoT, which offers more simplicity and flexibility for the developers, and hide/abstract the complexity of the low levels of the IoT. Thus, still understanding the different communication protocols and architectures of the IoT is needed.

2.3 Application domain: e-Health

The new vision of the smart city aims at providing better services to the citizens, by taking advantage of the current advancement of the information and communications technologies (ICT), the public resources, and the data which can be collected from all over the city using the IoT/WoT connected devices. Furthermore, the city can also be seen as a structure of services (i.e. Health, Energy, Traffic, Transportation, Public services, etc.), where each service have its own challenges and needs to be deeply analyzed. In this thesis, a particular attention was given to the Health services.

"Health for all", which ranges from sensitization and the delivery of information to keep people healthy, to care and support systems, has been a preoccupation of countries all around the world. In 2005 and under the roof of the World Health Organization (WHO), the countries members made a commitment to aim at achieving a Universal Health Coverage (UHC). However, with the previously mentioned issues, the traditional methods for delivering health and well-being services becomes insufficient and requires more developed solutions, such as eHealth (electronic Health), mHealth (mobile Health), TeleHealth, Electronic Health Records (EHR), Big data, Social media implication, etc., [31]. Moreover, in the last few decades, with the advances in the networking technologies and database systems, Ambient Assisted Living (AAL) researches have been applied in several domains in order to ensure end-users' safety and quality of life (QoL) at home. As a result, several new concepts have been introduced such as personal health management and monitoring, health prevention, healthcare and welfare, in order to provide better medical services, in particular for the persons with special needs such as elderly and disabled persons. In this thesis, a particular interest was given to the eHealth, which will be the main focus of the next sections. However, mHealth and TeleHealth services are also components of eHealth, thus they will be discussed.

First, we will start with the definition of the main notions that will be used in this thesis, The first main notion is eHealth, or e-Health, where the WHO defines it as "*eHealth is the cost-effective and secure use of information communication technologies (ICT) in support of health and health-related fields, including health-care services, health surveillance, health literature, and health education, knowledge and research*"[32]. They also provide fundamental eHealth developments strategies in order to build it on a strong foundation, which is described in [33]. The aim is to provide the engaged countries with clear plans and strategies in order to enforce the delivery of eHealth to their citizens. The notion of e-Health had emerged, since an enormous amount of health data and valuable medical knowledge have been electronically available and remotely accessible by different entities, especially through the Web. Hence it becomes easier to get, store, disseminate and exploit health information, especially with the improvement in the medical embedded sensors (or medical smart objects).

A practical solution to provide eHealth is called Telehealth, also known as telemedicine, which is defined by WHO as "*the delivery of health care services, where patients and providers are separated by distance. Telehealth uses ICT for the exchange of information for the diagnosis and treatment of diseases and injuries, research and evaluation, and for the continuing education of health professionals. Telehealth can contribute to achieving universal health coverage by improving access for patients to quality, cost-effective, health services wherever they may be. It is particularly valuable for those in remote areas, vulnerable groups and ageing populations*". [31].

Therefore, the main goal of the Telehealth, is to enforce the interactions between the patients and the health professionals, whenever and wherever they are, using communication technologies. These technologies can be either synchronous interactions, by

a phone call or by an audio/video calls with a doctor or a healthcare provider, or asynchronously interactions, where the doctor or the patients sends the requests and waits for the response by emails for instance. Consequently, the primary advantages of the use of distant health solutions, which asserts its importance as a key component toward the UHC vision of WHO, can be summarized as follow:

- A complementary solution to the traditional access to healthcare services, in particular in rural and deserted areas. Hence, more access to medical services and more coverage, as well as, access to specialized professionals, which are usually present only in big cities, becomes easier.
- To increase the access speed to the medical services, in particular regarding the regular checks and follow-ups, where the doctor do not need to see the patient physically. Thus, a simple multimedia session with the patient, at a predefined time, rendez-vous, can simplify the process.
- To reduce the tiredness, stress and the mobility issues for the already tired and stressed patients, and in particular for elderly and chronic patients.
- Provide more care and more follow-up to the dependent and elderly persons, in their homes.
- To encourage the patients to maintain their treatment, in particular for the elderly and chronic persons, by having more frequent remote consultation with the heath professionals.
- And finally to reduce the cost of the medical services as far as possible.

In this light, and as a result of interviews with doctors in the hospital of Rennes, several types of remote medical services were identified, and which are currently present in the hospitals:

- Telemetry: were patients equipped with medical sensors (such as a holters, alarm from artificial heart, pacemakers, Electrocardiogram (ECG) ,etc.) are monitored in a hospital, and sometimes even outside up to a certain perimeter. These sensors are able to send alarms to a dedicated service in the hospital in case of detection of an abnormality. Advanced algorithms can be also used to detect any abnormality in the behaviors of the patients for instance for the ones suffering from Alzheimer.
- Tele-expertise: where doctors from different specialties exchange their expertise regarding the case of a particular or common patient. This is usually done with elderlies, and patients suffering from chronic diseases, in order to provide the best treatment and to avoid conflicts in the prescribed medicines.
- Tele-consultation/Telehealth, or Telemedicine: which was previously mentioned.

And finally, the mHealth, which can be seen, for some particular case, as a complementary to the Telehealth. The literature definition of the mHealth is "*the use of mobile devices – such as mobile phones, patient monitoring devices, personal digital assistants (PDAs) and wireless devices – for medical and public health practice*" [34]. According to the survey report in [31], several types of mHealth programs, or application, exists in different countries, which can vary from the simple use of the phone to contact a doctor, either by message or by call, for the use of very complex system such as the patient monitoring by capturing the data continuously and transmitting them to a monitoring system using mobile ICT.

Moreover, the world population is aging. According to the statistics from the World Population Prospects in 2017 [9], elderlies aged 60 and over represents 13% of the world population, around 962 million, and it estimates that globally, population aged 60 or over is growing faster than all younger age groups. Moreover, the growing rate is by 3% per year, thus a quarter and more by 2050. Europe represents the continent with the highest percentage of 25% currently, with a projection to reach 35% in 2050. In France, the percentage of the population aged more that 60 reached 26% (Germany 28%, Denmark 25%, Greece 26%, Italy 29%, Spain 25%, Portugal 28% etc.). Thus, several challenges appear in response to the population aging, in particular regarding the healthcare services. Undoubtedly, elderlies, in particular and dependent persons in general, need more attention and long-term care, with more regular follow ups and more frequent medical assistance. On the other side, the medical desert issues and the decreasing number of doctors, particularly in France, and generally in the World, makes the previously mentioned issue more complex to solve, where, the medical desert, is defined as the areas characterized by insufficient medical services or difficulties to access the medical cares.

The e-Health solutions aim at solving these issues that face the current healthcare system. Hence, a special focus will be given to the Telehealth services, by proposing a new telemedicine architecture, which will be presented later in the chapter 3, and integrated in a real smart home in chapter 4.

Chapter 3 | Coupling WebRTC and the Web of Things

Contents

This chapter presents our first contribution, which couples the communication capabilities of WebRTC with the WoT. It starts with an introduction of the proposition, followed by the related works. It provides, then, the newly designed architecture. This architecture is illustrated in the e-Heath domain through three main use cases. Next, it introduces the security layer, which is then implemented and integrated together with the architecture. Furthermore, a privacy and risk analysis is provided, since the architecture is dealing with sensitive data, and since it is a requirement under the new GDPR law.

3.1 Introduction

Based on the previously presented technical context, this chapter proposes the coupling of the real-time communication capabilities of WebRTC, with the WoT, in an innovative way, through a newly designed architecture. The main purpose of this contribution is to provide real-time communication services, enhanced with contextual data collected from the environments of the communicating users. These contextual data can be gathered from various sources, such as the one collected from IoT sensors. As it is known, these sensors can collect an incredible amount of data for a wild range of applications, such as for weather, health, energy, traffic, home security, etc. The application of this architecture is then demonstrated in the domain of e-Health, for mainly providing remote medical services to the patients, and in particular the disabled and elderly ones, in their homes.

This new smart e-Health architecture is also illustrated by several innovative use cases in order to demonstrate its interest. Generally, the global goal is to provide remote medical services to the patients, and in particular to the dependent and the elderly ones, by taking advantage of the medical Things that they own. This need also comes from the medical desert issue in France, particularly in the rural areas. Hence, one of the objectives of this contribution is to:

1. Improve the doctor-patient relation, where the doctor can get additional data from the medical Things of the patient in order to get a better understanding of the state of the patient.

2. To provide a complementary solution to the traditional medical services, by providing remote medical assistance and services, which is also known as "Telemedicine", in order to cover more areas.

3. To facilitate the access to the medical services to the dependent and elderly persons.

However, the large scale deployment of WoT/IoT technologies in e-Health requires particular attention to the security and privacy issues. Especially the critical health data used by the architecture. They must be protected against cyber attacks that might eavesdrop or tamper personal data or cause harm to the patients, by guaranteeing the confidentiality and the integrity of these data. Additionally, in such environments, the insurance that only the authorized persons are allowed to access their data is mandatory. Hence, our contribution proposes, firstly, a new e-Health architecture and secondly, enhances it with a security layer, including end-to-end encryption (confidentiality and integrity), authentication, and a fine granular access control. Additionally, a privacy analysis is specially conducted for this architecture in order to identify the privacy issues regarding the protection of the data characterized as personal sensitive data.

3.2 Related works

The literature provides several examples of Web-based infrastructures aiming at improving the management of patients' diseases. Some of them are based on traditional communication technologies such as in [35] [36] [37] [38]. We present a summary of the main solutions that use WebRTC below. A recent work [39] uses in addition to the audio/video communication part of WebRTC, the DataChannels to exchange medical data between different medical entities, and where the patients have access to biomedical sensors. In [40], the authors propose an e-Health platform, providing specific services to diabetic persons (Type 1 diabetes mellitus (T1DM)). The proposed solution uses medical sensors and a humanoid robot. The robot interacts/dialogs with the patient in order to get better information on his state. All of the data (sensors and dialog) are collected by the robot, and then sent to the healthcare workers. The work of [41] presents an extension to WebRTC to enable peer-to-peer exchange of sensor data, and a proposal to enable Web applications to access sensor data and to bring nearby sensor streams to Web applications and multimedia communication over the Web. They mainly propose to extend the MediaStream API of WebRTC in order to manipulate also the sensor streams and to provide them to the customers. The approach is only theoretical without any implementation. In [42], the authors present a video conferencing system allowing online meetings between remotely located care coordinators and patients at their homes. They use a special device (TeleMedCare) to monitor the vital sign of the patients. In this work, there is no interaction with medical sensors apart from the TeleMedCare.

It is interesting to point that all these papers do not discuss how the security aspects are managed. The only security measurement taken in these systems is based on the security mechanisms natively provided by WebRTC. For instance, they do not discuss how they protect the communication between the browser and the biomedical sensors, and how the access to these sensors' data is controlled. Another work presents a vision of the remote monitoring and medical devices control in e-Health in [43]. They discuss the need of a unified platform that can provide real-time interaction, remote control and a scalable solution to manage a large number of medical devices and their data. The solution is based mainly on using in one hand an IoT gateway capable of communicating with different devices using different communication protocols, and in the other hand is able to establish a WebRTC communication. Moreover, the solution offers remote connection to cameras and medical appliances, alert notification to remote users, remote control of medical appliances and data aggregation, processing and storage through the Edge Cloud. However, with the WoT we can have better interoperability rather than trying to support the different communication protocols of the different devices, which have not been exploited in the previous solutions. Yet, the security issues such as access control, authentication, confidentiality and integrity of data between the IoT gateway and the different devices was not presented in those solutions. Especially in the case of e-Health where each private data of the user matter.

The following table summarizes the related works. The columns of this table are: references of the related works; If the solution uses WebRTC (“W“) or VoIP (”V“) and (”No”) if there is no multimedia; If there is an interaction with the IoT/WoT devices (sensors and/or actuators) (”IoT/WoT”); If they take into account the different security issues (confidentiality, integrity and authentication) (”Additional”), they just rely on the native security of WebRTC (”Native”), or they do not address it at all (”No”); If they propose an access control solution to the IoT/WoT devices (”Access Control”). And finally if it is implemented (”Impl”) or not.

Table 3.1 – A summary of the state of the art

Reference	V|W|No	IoT/WoT	Native|Additional|No	Access Control	Impl
[35]	No	No	Additional	No	Yes
[36]	No	No	Additional	No	Yes
[37]	No	No	No	No	Yes
[38]	V	No	No	No	Yes
[40]	No	WoT	No	No	Yes
[39]	W	No	Native	No	Yes
[42]	W	No	Native	No	Yes
[41]	W	IoT	Native	No	No
[43]	W	IoT	Native	No	Yes
Our	W	WoT	Additional	Yes	Yes

The summary shows, firstly, that only two propositions couples WebRTC and the IoT. However, they use only the native security provided by WebRTC and do not add any additional security to protect the IoT related flows toward the WebRTC clients. And secondly, none of the solutions implement the access control mechanism in order to protect the IoT devices against the illegal access. We argue that it is their main drawback.

3.3 Architecture

3.3.1 General architecture

The idea consists in adding new functionalities to control, manage and send data from/into the smart objects available in the WebRTC endpoints surrounding. Thus, during an ongoing WebRTC session, the data are collected from the smart objects and sent to the remote peer in real-time. This enables the creation of new opportunities, since in this case, each WebRTC endpoint is considered as a gateway to its own smart objects. Moreover, the access to these smart objects can be delegated to other persons for instance to the remote medical assistance. The security and the privacy aspects are very important, and they are at the core of our proposition. Hence, a particular interest is dedicated to the security mechanisms in order to avoid any security incident within the architecture.

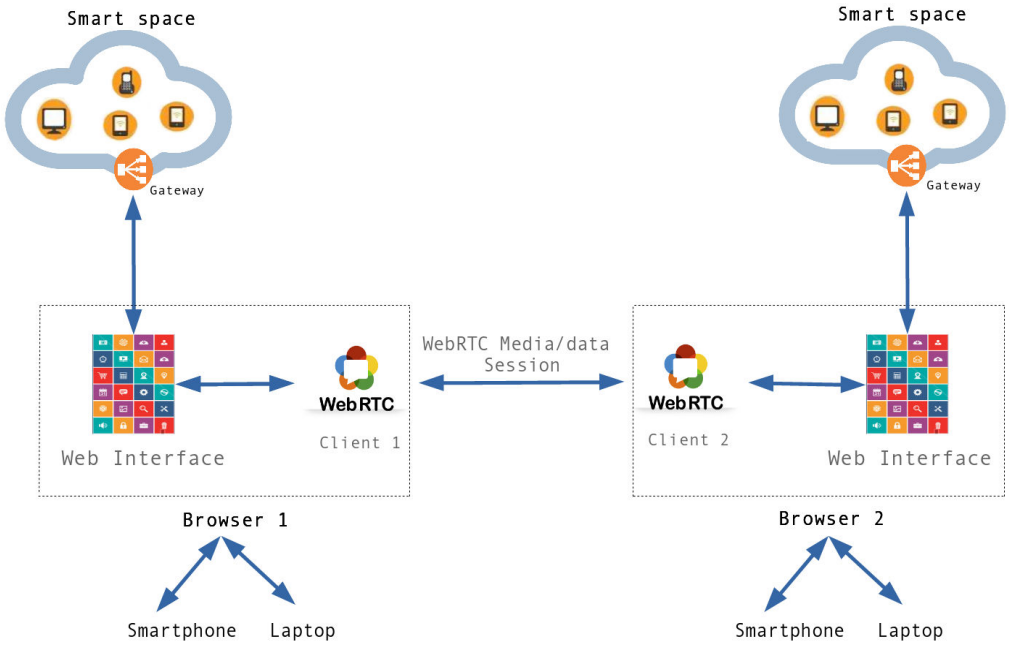


Figure 3.1 – Proposed architecture

A simplified view of the architecture of our platform is shown in Fig. 3.1. Using a WebRTC compatible browser, a user can, in one hand, communicate in real-time with a remote peer, and in the other hand, access his smart space. Accordingly to the WoT approach, the client application should take care of both the communication and the access to the smart objects. Each smart object is supposed to be identified using a unique URI and offers a Web API which allows the access to its resources. The smart objects are located in the private network of each user (home, office, etc.). These objects have their own organization, that we have previously introduced as a smart space.

3.3.2 Abstract architecture

In this section, an abstraction of our architecture is presented, where the different involved elements as well as how they communicate internally and externally are defined.

The Fig. 3.2 provides the main interactions, and distinguishes three main components: 1) the "WebRTC client" which is located in the browser. 2) The "Gateway" and finally, 3) the smart objects, which can be either constrained devices or less-constrained devices with enough capabilities to communicate and compute. The WebRTC client, can communicate either directly with the smart objects if they have enough capabilities to establish a channel (HTTP or CoAP), or with the gateway which will be the intermediary between the users and the constrained devices. The gateway is able to communicate with any sensor/actuator, either directly or through a controller that can be an Arduino, a Raspberry Pi, etc. The gateway uses different stacks in order to be able to use different communication technologies, such as WIFI, Bluetooth, Zigbee, etc. The main component of the gateway is a "NodeJS server", and we can differentiate between two types of communication, either inside the local network (i.e. i-int), or outside the local network (i.e. i-ext). The server also interacts with some databases

in order to store data such as the access control policies or log data. The database can be either internal (inside the local network of the server) or external (outside the local network). Finally, the smart space is represented by the gateway and the different sensors/actuators.

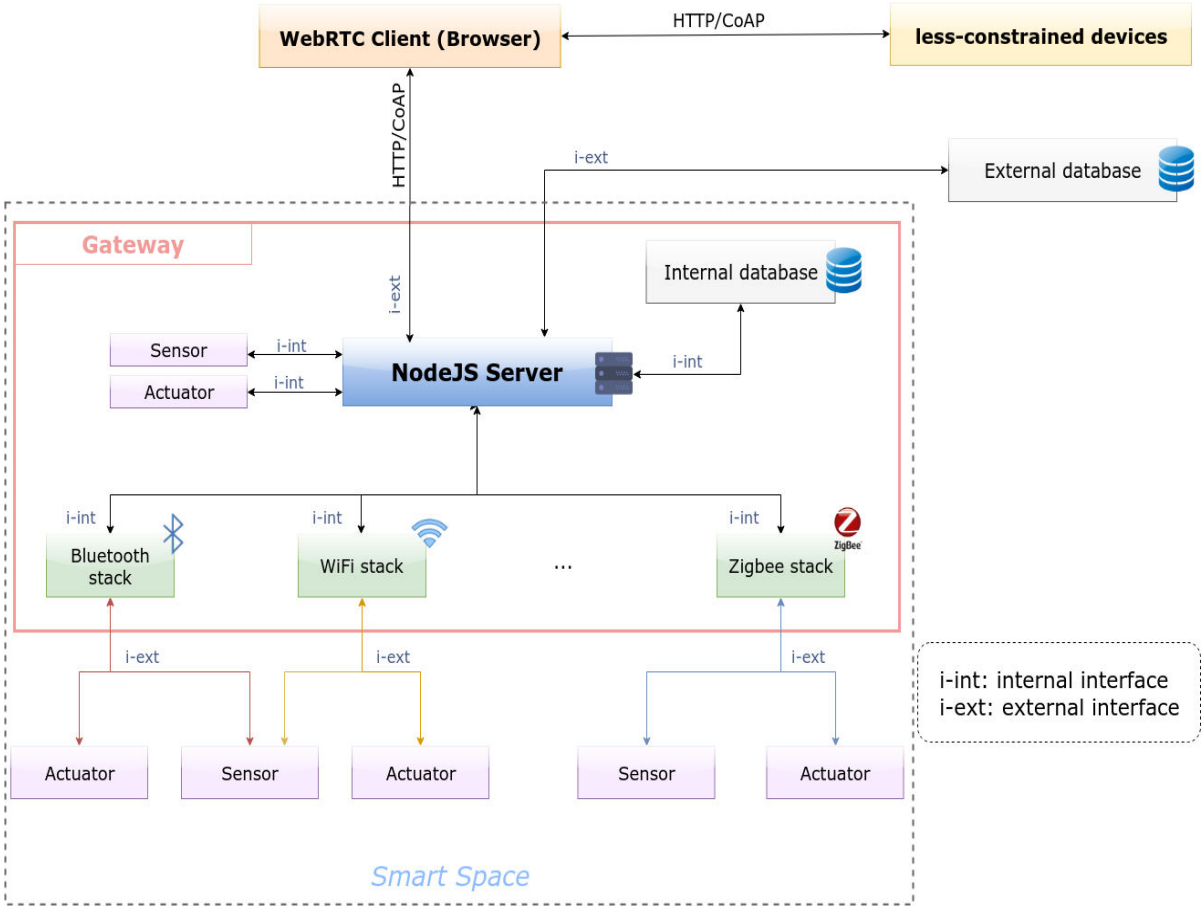


Figure 3.2 – Abstract architecture coupling WebRTC and the WoT

Additionally, the presented architecture can be refined regarding the possibility of intensive processing data at the edge of the network using the Fog Computing or Mobile Edge Computing (MEC) paradigms. Fog computing allows software applications to run on the edge of the network rather than on the core network in datacenters and servers. It helps in solving several issues related to the cloud as high latency, location awareness, reliability, etc. Moving data to the best location for processing can be resolved by fog computing according to [44, 45, 46]. This solution is relevant for IoT applications, such as smart grid and smart homes which are characterized by their latency sensitivity and therefore require immediate analysis of data and decision making as a conduction of action. Generally, the fog architecture is a three layer architecture composed of the device layer then the fog layer, and finally the cloud layer. This architecture, as described in many works [46, 47], is the same also for IoT networks. Namely, the fog layer analyzes the data collected in real-time from the device layer and then send the results to the cloud layer. Moreover, in order to avoid overwhelming the cloud with enormous amount of data, collected from the IoT networks, only the relevant and important data are reported by the fog layer, for instance abnormal processed result from an IoT medical sensor. In our proposed architecture in Fig. 3.2, the gateway can

be considered as a fog node which will collect the data from the different devices and then send them to the fog layer.

In our architecture, the gathered data from the IoT sensors are transfered directly in real-time through WebRTC without any processing, thus the omission of the use of fog computing at the edge of the network. However, the importance of such approach is not negligible and may be explored for future works. Processing IoT data at the edge may provide rapid responses in particular regarding the sensitive abnormal data, such as high fever from a temperature sensor, which requires quick reaction when detected.

3.4 E-health scenarios

The previously introduced architecture is illustrated with use cases in the e-Health domain, as shown in Fig. 3.3, where the users can be the patient/elderly/injured person in one endpoint of the WebRTC session, and the remote medical support, such as doctor/medical relief center/hospital, in the other endpoint. The medical Things represent the different medical embedded devices. Finally, the different stakeholders that can interact directly with the patients such as nurses, paramedics, and so on. They also have the authorization to access the personal medical devices of the patient.

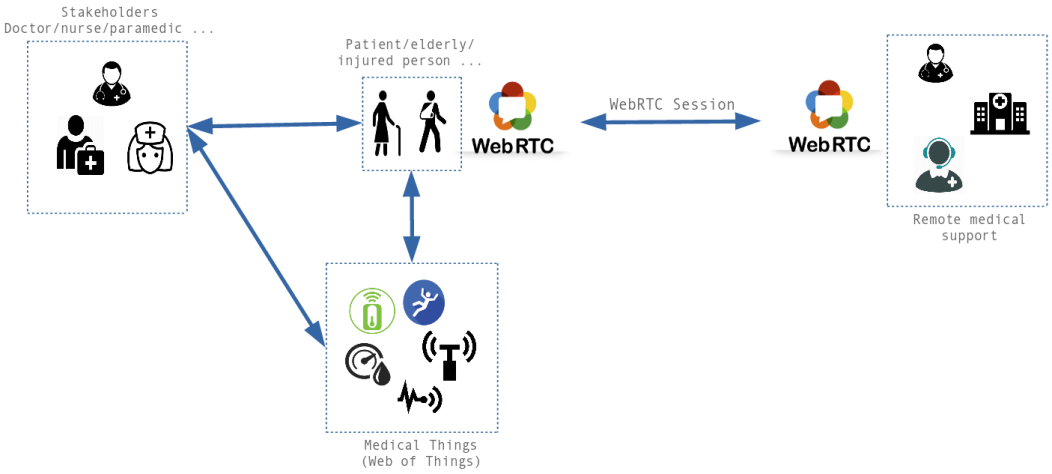


Figure 3.3 – Adapted architecture for e-Health scenarios

To gain a better understanding of the idea behind this proposed architecture, we will illustrate it with a set of scenarios applicable in the field of the smart health (or e-Health).

3.4.1 Continuous monitoring

The first use case is the continuous monitoring of a patient or an elderly person. In this case, wearable devices (such as a fall detection sensor, pressure sensor, peacemaker, etc.) are continually monitoring the state of the patient, as shown in Figure 3.4. In case of detection of an abnormal behavior (falling, heart problem, and so on), an alert is sent to the emergency relief center or to a doctor using WebRTC. In this case, the WebRTC

session is supposed to be either always set on or established urgently. The **mobility** of the patient can be also discussed, where the health status of a mobile patient are being continuously monitored, in particular outdoor. The patient is wearing embedded devices that continuously send health information to the medical center. This case can be very useful to be able to detect abnormality in the comportment of the patient and to quickly provide medical help when necessary.

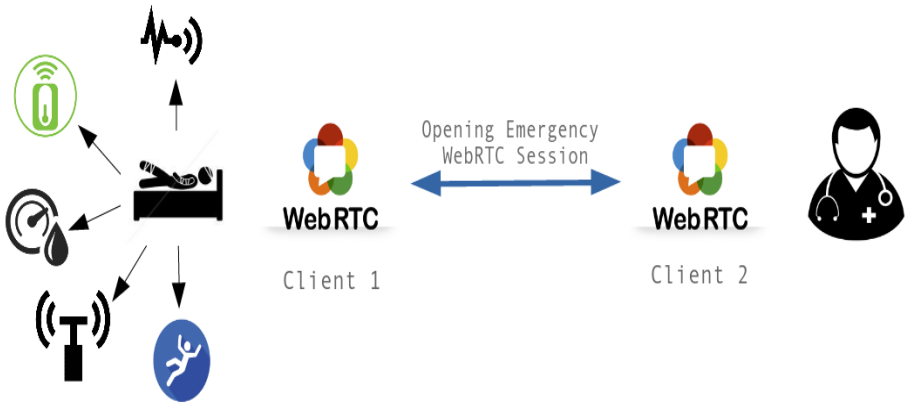


Figure 3.4 – Continuous monitoring

3.4.2 Remote medical consultation

Remote consultation, also known as "**Tele-consultation**", can be a very good alternative to the traditional consultations, and which can be used by any patient, and in particular for the elderlies and the disabled ones, where the physical consultation with the doctor is not mandatory or not possible due to some circumstances. Each patient, in this case is supposed to own a set of medical devices. Hence, the patient starts a WebRTC multimedia session with a remote doctor, and upon a request from the doctor, the patient puts on the medical wearable devices. The platform then collects these health data and send them in real-time to the remote doctor. Using these information and based on the discussion with the patient, the doctor should be able to perform a better diagnosis of the health state of the patient, as shown in Fig. 3.5. The patient remains master of his data produced by the different sensors and decides when and with whom they can be shared.

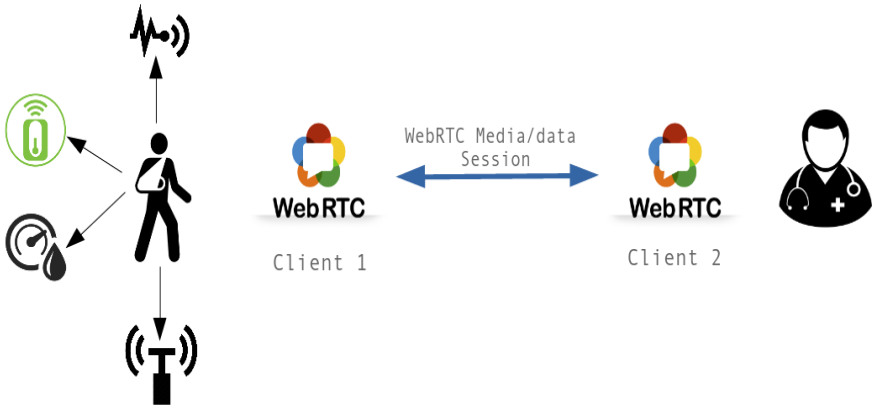


Figure 3.5 – Tele-consultation

3.4.3 Emergency intervention in case of an accident

The last use case is the one of an accident, as shown in Fig. 3.6. It describes the case of a remote medical assistance, such as an Emergency Relief Center (ERC), assisting the paramedic to provide better first aid to an injured person from an accident. First the paramedic attaches wearable medical devices to the body of the victim. The paramedic is also supposed to be equipped with a device supporting WebRTC, and communication with the ERC using it, and is able to collect the data from the sensors and send them in parallel. Hence, at the same time, the paramedic applies first aid to the injured person, and speaks with the ERC. This scenario can be interesting for instance, in case of serious injuries, where the ERC or the doctor can give real-time instructions to the paramedic, contact the different nearby hospitals in order to check their availability to receive the victim, and eventually to start preparing a surgery room in case of an emergency medical intervention.

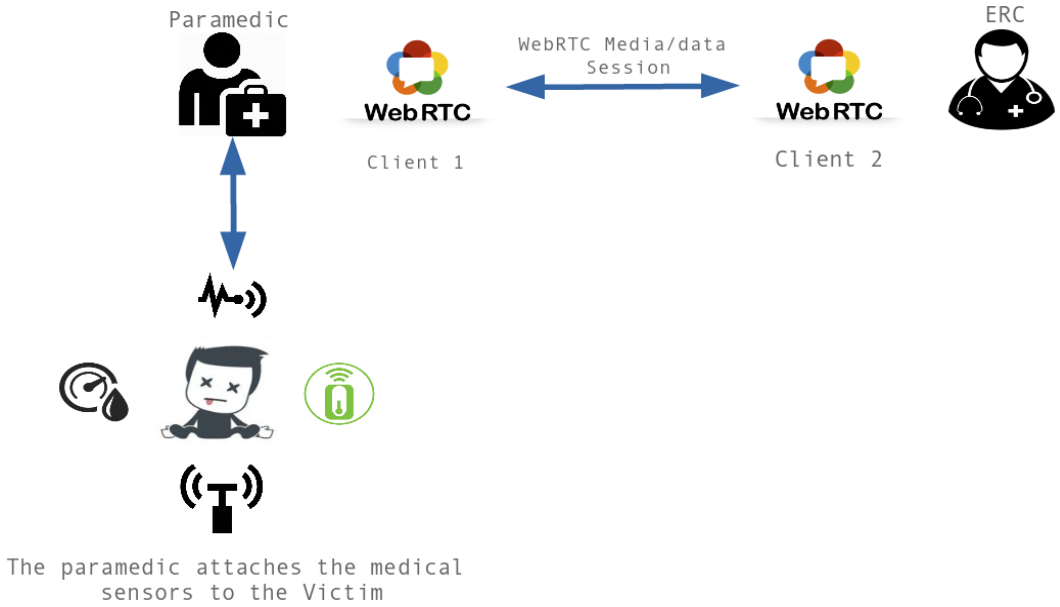


Figure 3.6 – Emergency intervention in case of an accident

3.5 Security Layer

Most of the current architectures, and in particular the e-Health related one, do not give enough importance to the security aspects, even though the users are very concerned about their data. Sending data using unprotected channels over the Internet expose them to the risk of eavesdropping and tampering by malicious attackers. In particular, health related data, which may be fatal in some cases, such as altering the attributes of a health actuator (pacemaker for instance). However, securing the exchanges only is not enough to encounter all the attacks targeting user's data. A user who is able to establish a secure channel with the resource is not always eligible to perform all the operations (modification, recovery, actuation, etc.) on the smart objects' resource (or resource in the rest of the chapter). Moreover, an unauthorized access to the resources may cause the loss of confidentiality, integrity, and availability of the resources. Hence, authenticating the users and restricting the access to the resource only to the authorized users, is required. In our propositions, both for the architecture coupling WebRTC and the WoT or for the next generation smart homes architecture, the security is crucial, specially when dealing with critical data such as the user's medical data. For this reason, several countermeasures have been considered in order to reduce the vulnerability of the system. Starting with the data confidentiality and integrity, the authentication, and finally, the access control.

In this thesis we deal only with the security aspects at the application layer. The security of the devices, the narrowband networks, and the security regarding the physical access to the gateway is out of the scope, and may be material for future work.

3.5.1 Confidentiality and Integrity

A study on the effectiveness of the end-to-end security for the Internet-integrated sensing applications, in particular for the constrained sensing devices [48], shows the viability of using end-to-end encryption both for the network and the application layer, as long as applications are able to accept a compromise between security, communication and resource usages. A particular interest was given to the security present at the application layer, in particular on top of CoAP by using DTLS over UDP [49].

Hence, in order to secure the communication between the different entities of our system, WebSocket/HTTPS is used in order to interact with the less constrained components of the system, and CoAP-DTLS (Constrained Application Protocol (CoAP) with Datagram Transport Layer Security (DTLS) as a security protocol [50]) is used in order to interact with the WoT devices. In our architecture, the WoT constrained devices interact mainly with a gateway, which plays the role of an intermediary, and also implements the security mechanism such as the establishment of the CoAP-DTLS channel and the storage of the encryption keys. However, to protect the data of our device, the gateway is supposed to be deployed in the local network of the owner, and can exchange data with only one endpoint. The gateway hardware used in our architecture is a Raspberry Pi 3. Additionally, the network between the Raspberry

and the smart objects, is assumed to be protected from the external attacks and that a physical access control mechanism is deployed in order to protect the smart objects from physical manipulation. More details can be found in [51].

3.5.2 Authentication

Another important security mechanism required for most of the architecture that processes personal data, is the verification of the identity of the user through an authentication. Basically, the user can request the access to the resource only after a successful authentication. For this, an Identity Provider [52] is used. It checks the user's identity and then issue an "Identity Assertion". This assertion can then be used by anyone who wants to verify the user's identity, as shown in Fig.3.7:

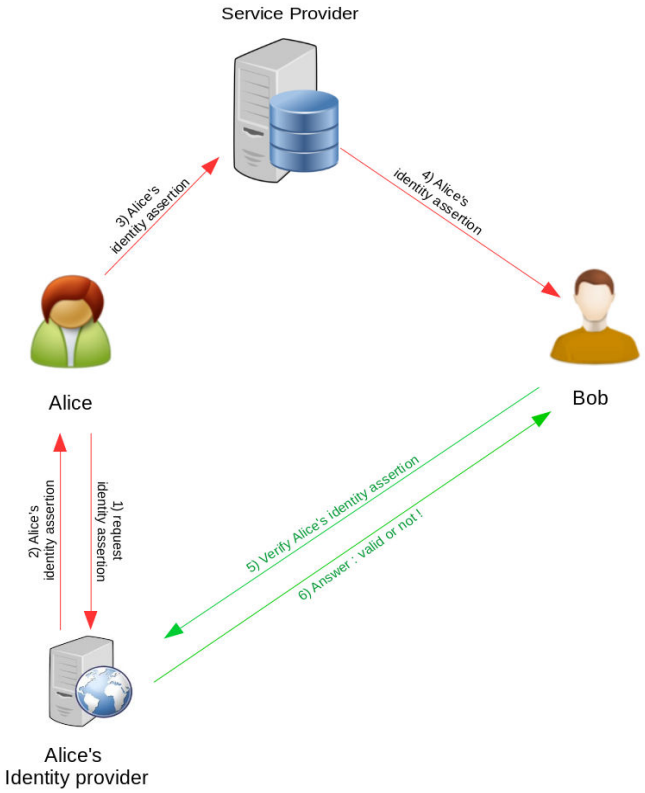


Figure 3.7 – User's authentication using an Identity Provider

3.5.3 Access Control

Traditionally, the access control focuses on the protection of data based on the identity and attributes of the users. And generally, the access control is used to protect front-end and back-end data and system resources by adding restrictions on who can access the data, which resource they have access to and what operations are allowed to be performed on the data. Ideally, access control prevents unauthorized users from viewing, modifying, copying or deleting the data.

The WoT context enables the smart objects to publish and exchange their information over the Web. However, some security preventions need to be taken into account, in order to deal with threats regarding the information exchanged over the Web. In

particular, regarding the access permissions for the Things information and resources. Such threats can be malicious clients, unwanted data sharing, or the traditional attacks, etc[1]. Hence, the question is how to allow Things to grant the clients a secure access to their resources in such environment?

For computer network systems, and in order to facilitate the access control, standard authorization models are proposed, such as Access Control List (ACL) [53], Subject/object access control matrix [54], Multilevel security using information flow [55], Role-base access control (RBAC) [56] and Attribute-based access control (ABAC) [57], dynamic authorization model have been suggested [58] and capability-based systems.

In our case, the RBAC model is judged to be the one suitable for our architectures. Since, the global vision of all the users that can interact within the architecture, and what information they can access, is already clear. Especially in the case of e-Health, where for instance, the different authorized tasks of a nurse are already known, and any other operation must be forbidden. In addition to the advantages brought by RBAC such as the least privilege principle and the separation of duties. In this case, each user have a predefined role with precise actions. Hence, the number of rules and the different users of the system (compared to ABAC), is totally controlled. Moreover, adding new users to the system become easier, since a simple role attribution allows the user to access the resource on one side, and protects the resources on the other side. In addition to the possibility of changing or revoking a role of a user within the system.

3.5.3.1 Role-Based Access Control (RBAC)

RBAC is a non-discretionary access control ensuring that access is granted only to the authorized users. It is a model proposed by Ferraiolo and khun in 1992 [59], which is based on creating a relationship between the role of a user in a given environment and what this role can do, or in other words the permissions. For instance, in many organizations, the end users does not own the information even if they have an access permission. For these organizations, the corporation or agency is the actual owner of the information. Hence, the control of theses information is often based on the employee's job rather than data ownership. Access control decisions are often determined by the roles that the users are affiliated to as part of the organization. This includes the different tasks, responsibilities, and qualification. For instance, the roles of an individual in an infrastructure such as a hospital may include: doctor, nurse, pharmacist, and so on and so forth. RBAC policies are based on the functions a user is allowed to perform within the organization that he/she is affiliated with [60][61].

A role can be seen as a set of transactions that a user or a set of users can perform in an organization. Membership in a role is granted and revoked by an administrator. Usually the different transactions are attached to the different roles also by an administrator. Such transactions, for instance, can include the ability for a doctor to

[1]owasp.org/index.php/OWASP_Internet_of_Things_Project#tab= IoT_Attack_Surface_Areas

perform a diagnosis, prescribe medication, and add an entry to a record of treatments performed on a patient.

3.5.3.2 Policy management architecture using PDP/PEP

The IETF model [62] introduces two main entities, namely the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP), as presented in Fig. 3.8. The first one, which is the smart part of the architecture, acts as a controller the goal of which consists in handling and interpreting policy events, and deciding in accordance with the policy currently applicable, what action should be taken. This one is transmitted to the PEP which has to execute it.

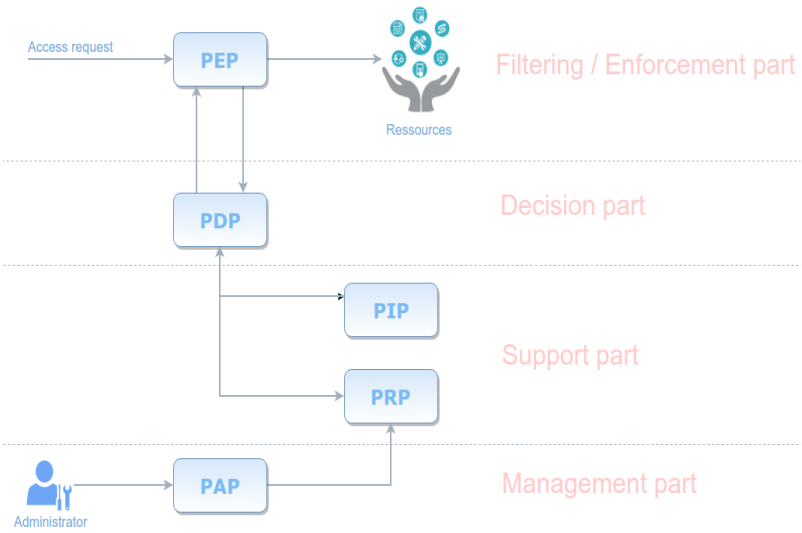


Figure 3.8 – General Policy Engine

As shown in Fig. 3.8, The model can also be divided into four layers: 1) Management: via the Policy Administration Point (PAP). 2) Support: via the Policy Information Point (PIP) and the Policy Retrieval Point (PRP). 3) Decision making: via the PDP. 4) Enforcement: via the PEP. The PRP serves as a database and stores the different policies. The PIP can be used to store the attributes of the system. Examples of PIPs may include corporate user directories (i.e. LDAP), databases, UDDI etc. The PDP may for instance ask the PIP to look up for the role of a given user. And finally the PAP which can be seen as an administrator interface for adding/removing/modifying/-suppressing policies.

3.5.3.3 Identifying users' roles

First, an example of the basic roles that can be present in such architecture are identified. These roles can be modified an tailored for each use case. These basic roles are shown in Fig.3.9. A Hierarchical RBAC (HRBAC)[63] model is used, where each role inherits the permissions of the lower role. The main roles are: "SuperAdmin", "Admin", "Owner", "Guest" and "Basic". The SuperAdmin role is the most powerful role, in most cases, the users with such role are able to perform any action within the system. The Owner, and the Admin roles are also powerful roles, they enable users to access most of

the data, and to add and remove the smart devices from/into the system. The Guest role, is mainly for external users, where the Owner can for instance delegate the access to one or to a set of his/her devices to this guest. And finally the Basic role with the least possible privileges, we separated the Basic from the Guest, since we need to have different roles for a close family and a friend for example. As a reminder, these identified roles are only for experimentation purposes for this architecture. They can be modified at any moment, since each developer or service provider has different vision on the users of their systems, hence, different roles.

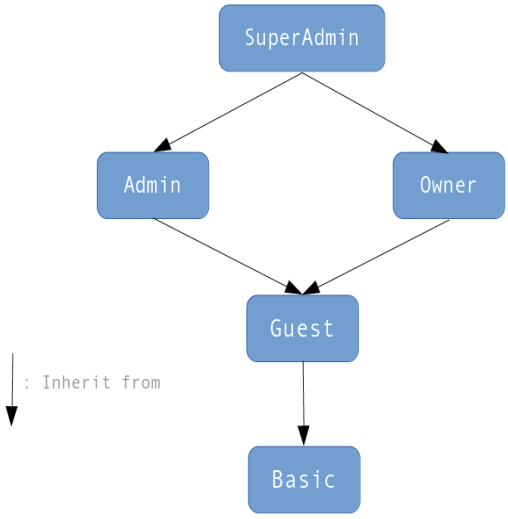


Figure 3.9 – Example of a basic RBAC roles for basic configuration

Then, the basic model was elaborated to fit the different e-Health use cases, as shown in the following figure:

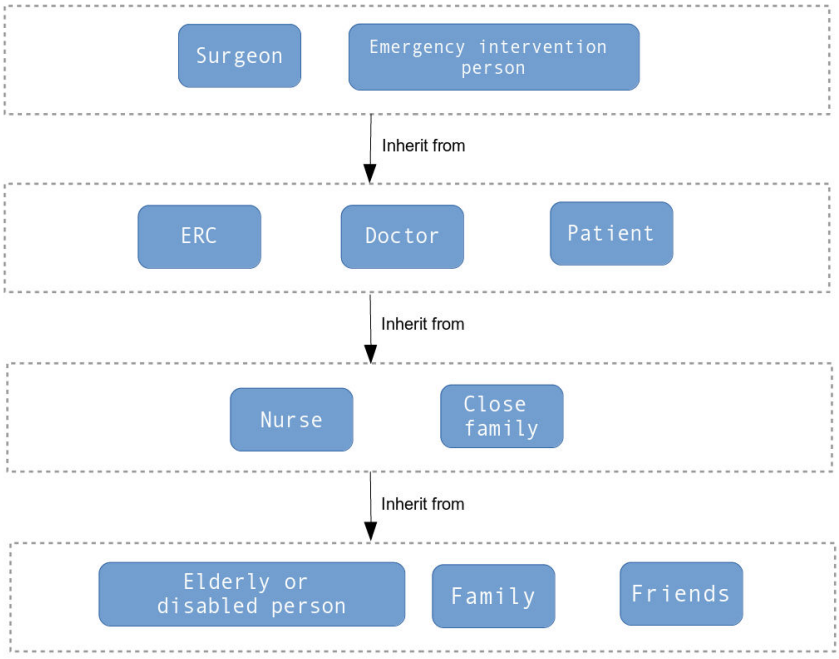


Figure 3.10 – Example of RBAC roles for an e-Health scenario

3.6 Abstract architecture enhanced with the security layer

The previously introduced security mechanisms are then integrated in the proposed architecture at the application layer. Hence, the previously introduced abstract architecture in Fig. 3.2, enhanced with these mechanisms, evolved to the abstract architecture presented in Fig. 3.11. First, all the interactions are secured using either HTTPS or CoAP-DTLS. Then, the authentication step is done at the WebRTC client level. And finally, an access control mechanism, based on RBAC, was added. This access control is based on an PDP/PEP model. A new component is needed in our architecture, which is the "DTLS Proxy", since some difficulties were encountered regarding the compatibility of the browser to support the CoAP implementation secured with DTLS. The proxy is capable of creating CoAP-DTLS channels with the remote entities. To have a secure communication, the browser uses HTTPS in order to communicate with the DTLS proxy.

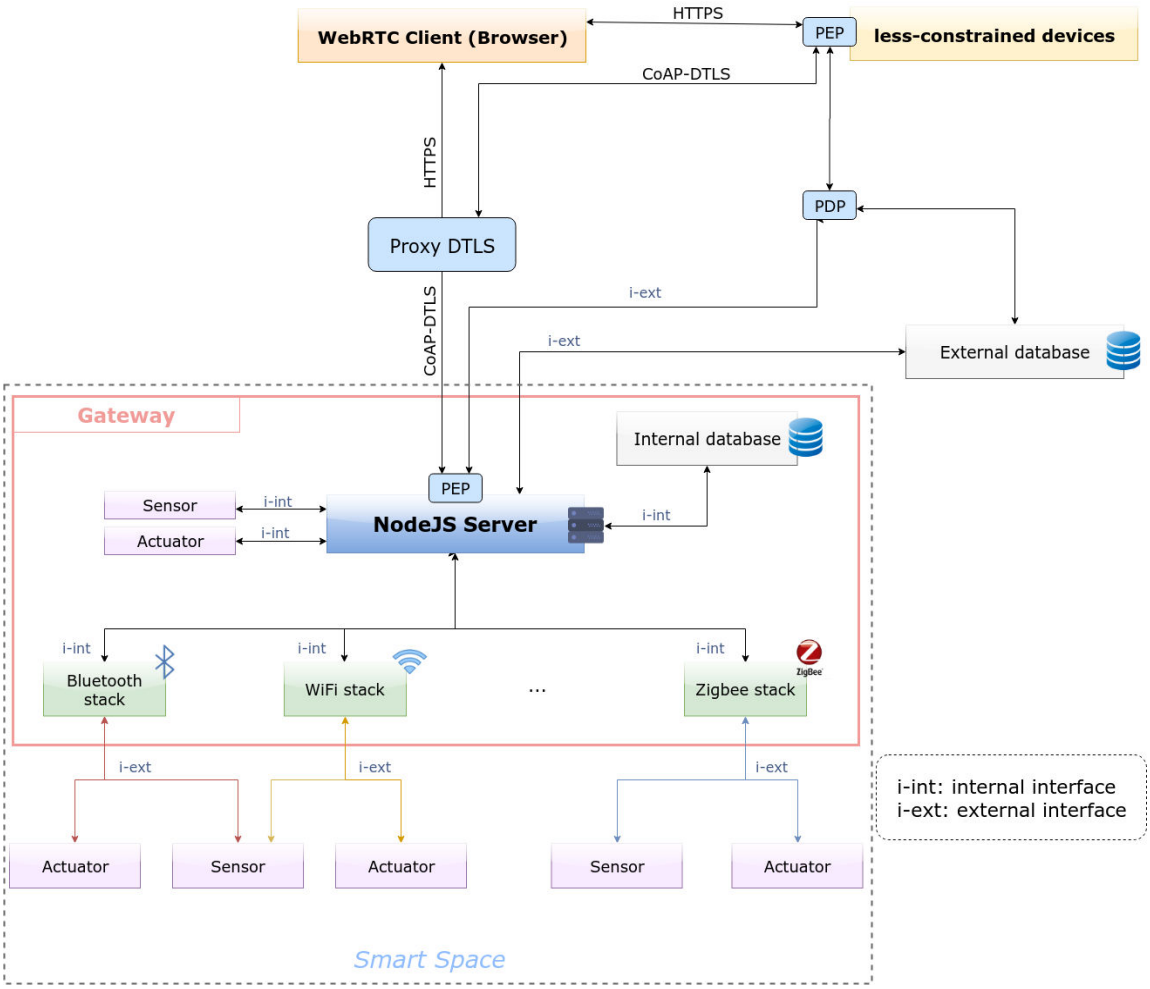


Figure 3.11 – Abstracted architecture with an access control

3.7 Implementation

The Proof of Concept (PoC) is implemented using Node.js. This PoC is capable of establishing a multimedia communication between two users using their browsers, and of allowing the users to access and retrieve data from their smart objects, and to send them to the remote peer, in real-time if needed, as shown in Fig. 3.12.

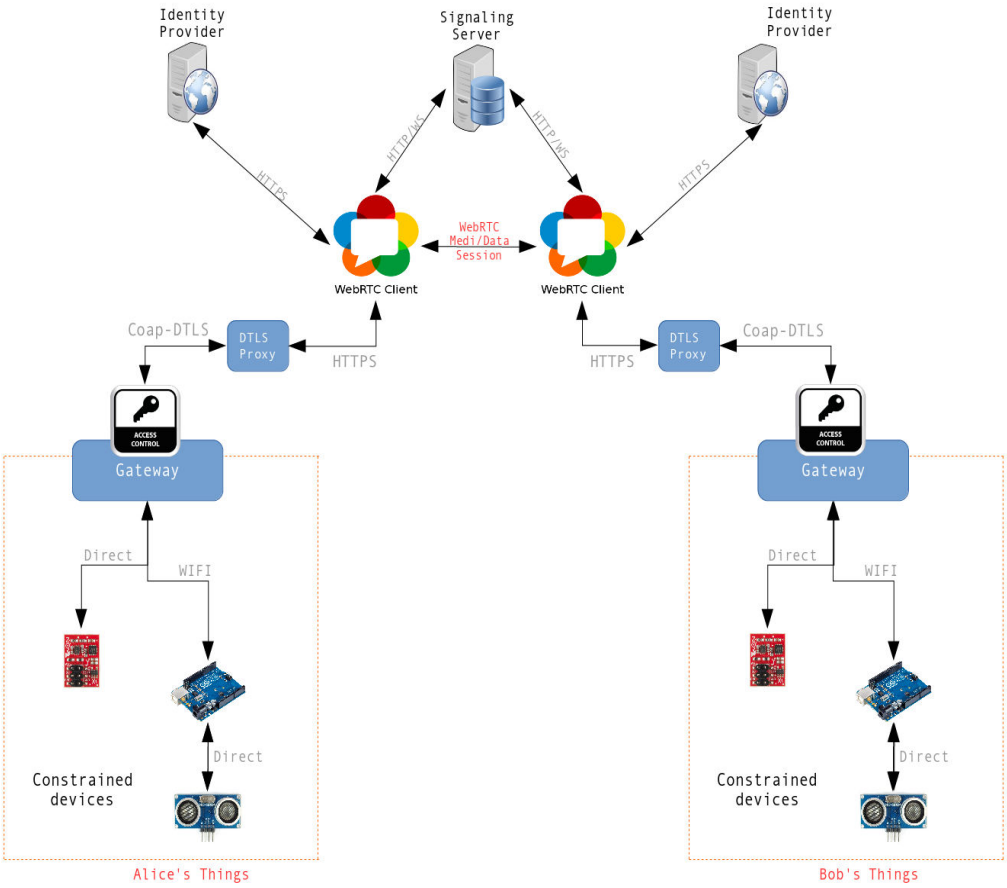


Figure 3.12 – Complete architecture with the access control

The first part of the architecture is WebRTC. It is composed of two clients, a signaling server and two IdP. First the client gets the WebRTC JavaScript application from the server, and execute it inside the sandbox of the browser. Then, the signaling process starts, where the user contacts the remote part of the communication. Finally, with all the exchanged parameters, a WebRTC media/data channel is set. As for the IdP part, it provides the users with an identity (after a valid authentication on the IdP side), so that they can authenticate themselves to the other side of the communication.

The second part is the WoT part, where each endpoint is considered as a gateway to his smart space, and have full control over all the smart objects (SO) inside. In the case of the e-Health, wearable medical sensors can replace the ones used for the PoC. These SO can communicate using any protocol, since the WoT abstracts all the complexity of the connectivity part.

Moreover, in order to protect the data of the different resources: 1) an end-to-end encryption is established between the WebRTC client and the Gateway, 2) the identity of the user is verified using the IdP, 3) and finally an access control is implemented.

Two methods were identified in order to perform the access control. The first method is a centralized one, where all the access control logic is implemented inside the gateway, and by logic we mean a local representation of the policies, controlling the request according to the different policies, and a local storage of the users' information, as shown in Fig. 3.13. In this case the PDP and the PEP are located in the same component which is the gateway (the NodeJS server in Fig. 3.11). The second method is a decentralized one, where the decision and the execution are done in different locations, and mainly, by separating the PEP and PDP. The PEP, in this case, is located inside the gateway in order to execute the decision made by the PDP located outside the gateway on another server. The policies are also decentralized and are stored in a separated database outside the gateway, as shown in Fig. 3.14.

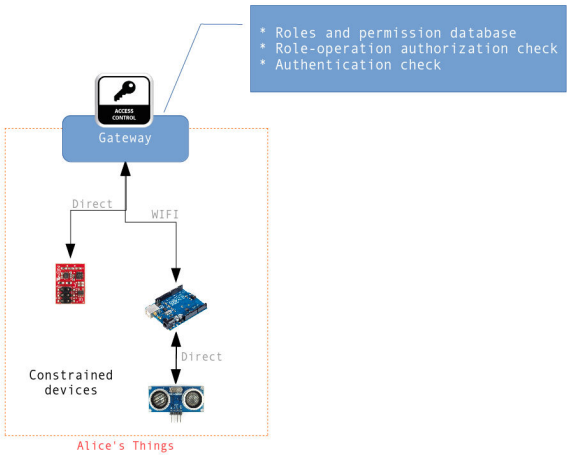


Figure 3.13 – Local access control

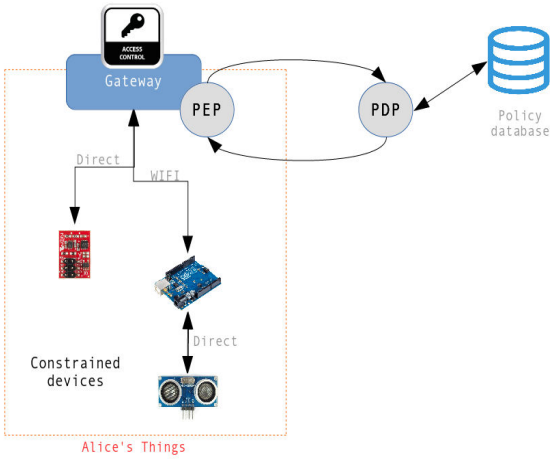


Figure 3.14 – PEP/PDP model

In the final implementation of the architecture, both the PDP and the PEP are located in the gateway, as in Fig. 3.13.

3.8 Privacy safeguard

In such architectures, in order for the users to access these kind of remote services, they need to provide and exchange personal data, and in particular, in this use case, the health related ones. Therefore, user's private information may be exposed to privacy violation and disclosure. A special focus is given to the privacy of our proposed architecture, and in particular to one of the e-Health use cases. Hence, understanding the privacy holes regarding the protection of the personal health related data, identifying the privacy leakage points and studying the privacy requirements are important in order to propose a privacy safeguard for the proposed healthcare architecture. Additionally, a risk analysis, the sources of these risks and the possible countermeasures are also conducted during this process [64].

3.8.1 Introduction

Privacy is considered as a basic requirement for consumer acceptance of any kind of services that uses personal data, and in particular the health related ones. However, most of the current healthcare architectures do not provide a deep analysis of the privacy issues, which may lead to privacy breaks in the future. This section aims at analyzing the privacy leakage issues regarding the remote health services provided by our proposition, and building a privacy safeguard framework in order to ensure the respect of the principle of the "privacy by design". The privacy analysis should be done all along the life cycle of the personal data, from the owner to the final destination and afterward. Privacy requirements should be then implemented within the architecture by including the security countermeasures required by the privacy analysis.

Among the previously provided use cases, shown in section 3.4, the privacy analysis will be conducted on the remote consultation. However, we need first to narrow down the context. In order to deeply understand the use case, several interviews with doctors in the CHU of Rennes, were done. One of the results is that, this use case is defined under the category of telemedicine services. And under this category, other sub-categories can be found. Some of these sub-categories can be provided by the already defined use cases. With regards to our proposition, the sub-categories are:

Tele-consultation: where the patient speaks with the doctor remotely using an audio/video system while sending health data coming from medical IoT sensors.

Telemetry: where patients equipped with medical sensors (such as a holter, alarm from artificial heart, etc.) can be monitored either inside a hospital or remotely, and in the case of an anomaly, an alert is sent to the corresponding caregiver. Hence, the remote monitoring is a particular case of the telemetry.

Tele-expertise: doctors exchange expertises for a particular case of a patient.

The privacy analysis will focus on the tele-consultation use case, in order to protect the critical and sensitive medical data transits in real-time between the doctor and the patient. The goal is to be able to identify the different privacy risks inside the architecture and to try to mitigate them by providing a set of countermeasures. The full user scenario of the tele-consultation use case is described in the chapter 6, section 6.4.5. This scenario allows fixing the possible interactions within the system between the patient and the doctor (which can be seen as a possible implementation), in order to identify the personal data life cycle.

3.8.2 Related works

The literature provides several solutions for e-Health services, in particular for the remote monitoring service. [65] builds an Android mobile application for the healthcare services, which uses the idea of Internet of Things (IoT) and cloud computing. The

application provides the end user with visualization of their Electro Cardiogram (ECG) waves and data logging. The only security aspect that they deal with is the encryption of the medical data and the secure upload to the cloud. However, they lack the privacy aspects and the analysis of the other security issues. In [66], the authors provide an interactive telecare system (ITCS) particularly designed for diabetic patients, providing interaction with caregivers in order to increase self-care quality by adopting IoT. Also, their system enables direct communication between patients' medical devices (in particular the blood glucose monitor) and their caregivers' smartphone. However, the only security aspect that they deal with is the encryption (using AES) of the transmitted medical data from the ITCS to the cloud. Hence, the privacy aspects and the other security issues are not mentioned. In [67], the authors discuss how to build an ad-hoc extensible healthcare remote monitoring system by using low cost wireless sensors and already existing Internet of Things technology as a communication platform. The system alerts, in real time, the patients' relatives or the medical doctors in case of detection of an abnormality for elderlies. However, they do not provide neither a security nor a privacy analysis of their proposition. Moreover, other solutions for health monitoring are present in [68] [69] [70] [71] and in [72]. However, none of them consider the privacy issues regarding the personal health data, which can be an obstacle toward the adaptation of these solutions. This paper [73], provides a privacy analysis of healthcare services in the smart city, and how the data's privacy should be protected while interconnecting the different entities. They propose a privacy engineering approach and safeguard framework are proposed for smart city healthcare services. They analyzed the privacy of the data for a particular use case of "Smart in-Home Emergency Health Service", and they identified the privacy requirements. However, the work presents only a theoretical study, with high level privacy instructions in order to evaluate the privacy of any use case.

Previously in [74], an e-Health architecture, with well developed use cases is presented, together with a deep security analysis of the different layers in [75]. Compared to the related works, and in addition to the already implemented security layer, a well defined privacy analysis especially suitable for the next generation tele-consultation services is provided. Also, particular interest is given to the personal health data protection issues and to the different rights of the patient in our architecture. Finally, and with the help of the privacy impact assessment (PIA)[76], a risk analysis is provided, together with the countermeasures to mitigate them.

3.8.3 Regulations

In order to conduct the privacy analysis, several regulations and frameworks have been studied in order to have a global view of the privacy requirement for the health related personal data, and we mention:

- The analysis of the European GDPR laws in [77]
- The use of the French Privacy Impact Assessment (PIA) framework [76]
- The European Working Party 29 [78]

- The French regulation regarding the personal health data, through ASIP Santé (Agence des Systèmes d'Information Partagés de Santé - The shared healthcare information systems agency) [79]

Hence, these regulations and frameworks will be used in order to understand the privacy weaknesses regarding the protection of the personal health related data, to identify the privacy leakage points and to study the privacy requirements, in order to propose a privacy safeguard for the proposed healthcare architecture. Additionally, a risk analysis, the sources of these risks and the possible countermeasures are also conducted during this process.

3.8.4 Data protection principles and user's rights

According to the GDPR law [77], the main principles that need to be applied to the personal data, and in particular to health related one, and that need to be always respected by the data controllers, are:

- Lawfulness, fairness and transparency
- Purpose limitation
- Data minimization
- Accuracy
- Limited storage
- Integrity and confidentiality
- Accountability

In such health related architectures, the controller of the data holds several responsibilities in order to protect the data going through the platform, which are mainly, but not limited to:

- Confidentiality and data integrity.
- Secure data storage.
- Anonymity of the data.
- Controlled access to these data.
- Sending data through secure channels.

Moreover, under the new GDPR laws, the main principles and rights that need to be respected in our architecture, and guaranteed to the patient, are:

The right to access: can be granted to the patient, by allowing him/her to access their medical records through the platform, by a health professional or an equivalent. Several other requirements must also be met such as the age and the mental state of

the patient, in order to provide this right.

The right to portability: in order to mitigate and transfer all the records related to the patient in the database in a standards format, which can be then integrated by any other entity chosen by the patient. This option must be implemented for the last version of the architecture.

Right to rectification or erasure: should be requested to the medical support, for instance regarding the medical record.

The right to be forgotten: upon valid request of the patient, all the related personal data must be erased from the database. This part is guaranteed by the medical service and not the platform.

The rights to be informed in case of a breach: where the concerned persons need to be informed as soon as possible. In our architecture, it can be done through an alert or a notification.

The right to be informed/transparency: where the information should be as clear and simple as possible, and in the native language of the patient. Also, this part is provided by the medical service and not the platform, since the medical records are stored in the health infrastructure database. However, the user can request the architecture to provide a visualization of the medical data collected by the medical sensors.

3.8.5 Assets: personal health data protection

The collected data will be mainly the patients related data (i.e. vital signs such as Body Temperature, Pulse Rate, Respiration Rate, Blood Pressure, etc.), collected via IoT medical sensors. The only persons that have the right to access these data are: the concerned patient, the related doctors, the health infrastructure where the doctor works, the administrator of the database, government and researchers. For the last three destinations, special consents need to be provided by the patient.

In this architecture, the life cycle of the health data is described in Fig. 3.15:

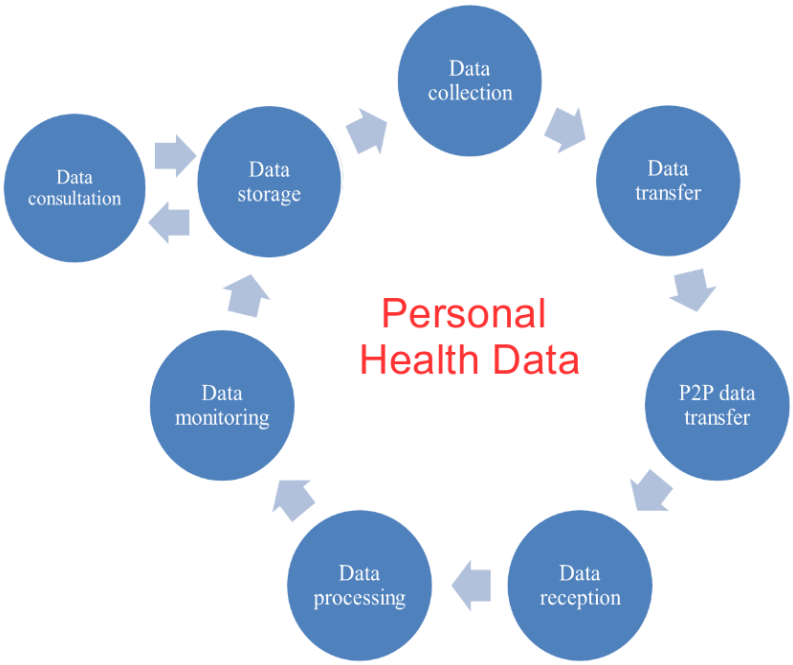


Figure 3.15 – The health data lifecycle within the architecture

Data collection: medical data (i.e. vital signs) are collected from the patient using attached medical body sensors.
Data transfer: data are sent from the sensors to the platform.
P2P data transfer: data are transferred to the remote peer (i.e. the doctor) using a secure WebRTC channel.
Data reception: ensuring the reception of the medical data from the remote peer (i.e. doctor) side.
Data processing: data are processed using for dedicated algorithms in order to evaluate the received health data and to perform a first analysis in order to detect anomalies.
Data monitoring: the result of the processed data will be shown to the doctor (i.e. graphs).
Data storage: the received data will be stored in the database of the remote peer (it can be the database of the doctor, or a secure provided database, or in the facility where the doctor is, i.e. the hospital).
Data consultation: the authorized parties can access these medical data anytime through the database.

As for the duration of the processing of such data, it depends on each step of the tele-consultation process. Since the main purpose of our architecture is to perform medical tele-consultations, telemetry and tele-expertise, we consider the collected data as medical records for the patient. The law defines the duration of conservation of lifetime plus 6 years for computerized/electronic medical records [80]. And this is applicable for our architecture in the case if the data is stored in our database. However, if the data is stored in a hospital, this regulation should be applicable to the database of the hospital. The architecture do not store the data during the life cycle of the health data, since the data are sent in real-time to the remote peer. In case the data is

stored in the platform, mechanisms for controlling the access and for suppressing data at the end of their retention period need to be deployed.

3.8.6 Personal health data protection requirements

In this architecture, in order to make the processing of the personal health data licit, several requirements need to be respected. We mention:

- The explicit consent of the patient in order to use their health data coming from the medical body sensors.
- The explicit consent of the patients when sending their data to the remote medical support (i.e. doctor).
- The explicit consent of the patient in order to use his/her personal health data for research purpose.
- The explicit consent of the patient to the doctor in order to transfer his/her data to another medical entity (i.e. to another specialist).

3.8.7 Security countermeasures and risk analysis

In this light, several security countermeasures need to be implemented in the architecture such as:

- **Encryption**: by using HTTPS and CoAP-DTLS, all the data flow in the architecture are encrypted. Ideally, the data must be peer-to-peer encrypted, from the patient's sensors to the doctor's interface or database.
- **Anonymity**: for research and remote expertises purposes, and after an explicit consent of the data subject (the patient), the data should be anonymized in order to protect the privacy of the patient.
- **Data minimization**: where only the strictly needed data should be collected and processed during tele-consultation process.
- **Website protection**: by securing the website using X.509 certificates, provided by a certificate authority, by using a strong authentication mechanism such as using OAuth, and by applying the appropriate access control mechanisms.
- **Integrating the privacy in the design process of the architecture**: which is the main aim of this analysis, by building a privacy safeguard framework, and analyzing the privacy risk regarding the health data, and to apply the results on the final design of the architecture.
- **Controlling the access to the health data**: an unauthorized access to the resources may cause the loss of confidentiality, integrity, and availability of the resources. Hence, authenticating the users and restricting the access to only the authorized users is required.

- **Integrity**: by maintaining and insuring the accuracy and the consistency of the data all along its life-cycle, in particular during the transmission, the processing and the storage of these personal data. Unfortunately, data can be compromised in several ways: transfer errors, bugs, malwares, hacked, etc. Hence, mechanisms guaranteeing that the data is intact and unaltered should be implemented using cryptographic algorithms, such as error checking and validation methods.

- **Confidentiality**: in order to prevent sensitive health data from being disclosed to the wrong and unauthorized persons, while making sure that the authorized ones can access it.

- **Secure storage**: by implementing a tiered data protection and security model, applying both logical (authorization, authentication, encryption and passwords) and physical (restricted access and locks on server, storage and networking cabinets) security of the database, in particular in case of health data.

- **Archiving**: the Health Information Portability and Accountability Act (HIPAA) [81] requires health service providers to store theses information for years, sometimes decades. Several options can be considered for the data archiving strategy, it can be ranging from on-site tiered storage within a storage area network (SAN) to off-site storage with an external outsourced regulated service provider.

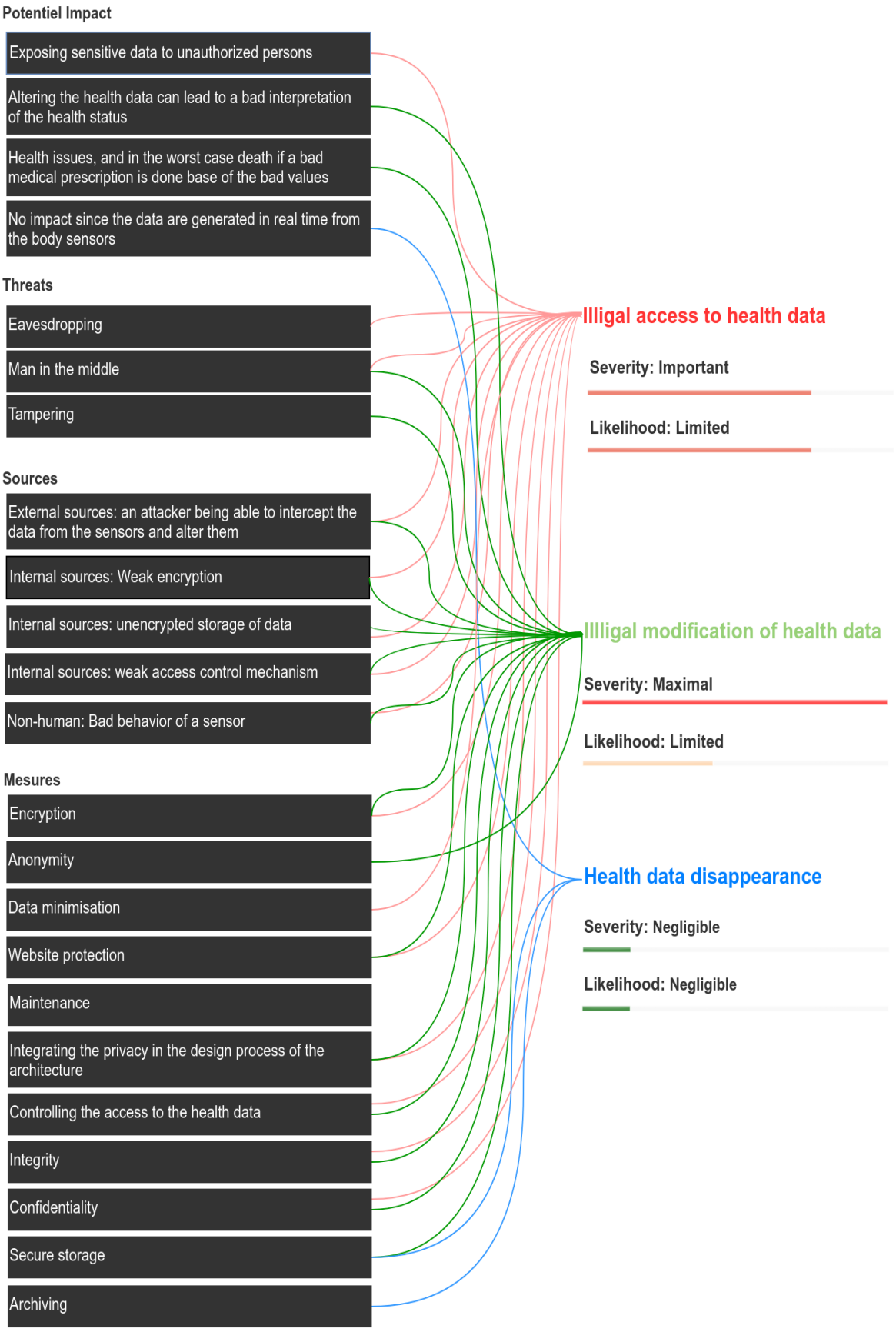


Figure 3.16 – Global view of the risk analysis

A global view of the privacy risks, together with some of the potential impact on the personal data, the threats that may trigger the risk, the sources of the risk and the

possible measures to counter them (represented by the same color), are presented in the Fig. 3.16. The risk analysis considers the overall architecture, including the communication system and all the exchanged data. However, it is worth mentioning that this is not an exhaustive list of risks and their respective analysis, other works may focus more on the risk analysis of such architecture.

For instance, for the illegal access to the health data risk, one potential impact of such risk is the exposure of the sensitive data to unauthorized persons, and some of the threats that may cause this risk are the eavesdropping and the man-in-the-middle attacks. The sources of this risk may be external such as the case of an attacker intercepting the sensor's data and eventually altering them, or it can be internal such as with weak encryption of the data, unencrypted storage, weak access control mechanism, and finally it can be a non-human behavior, such as a bad behavior of the sensor. Finally, some possible measures that can be taken in order to mitigate such risks are using encryption, data minimization, website protection, integrating the privacy by design during the early stage, robust access control to the health data, and last but not least the confidentiality and the integrity.

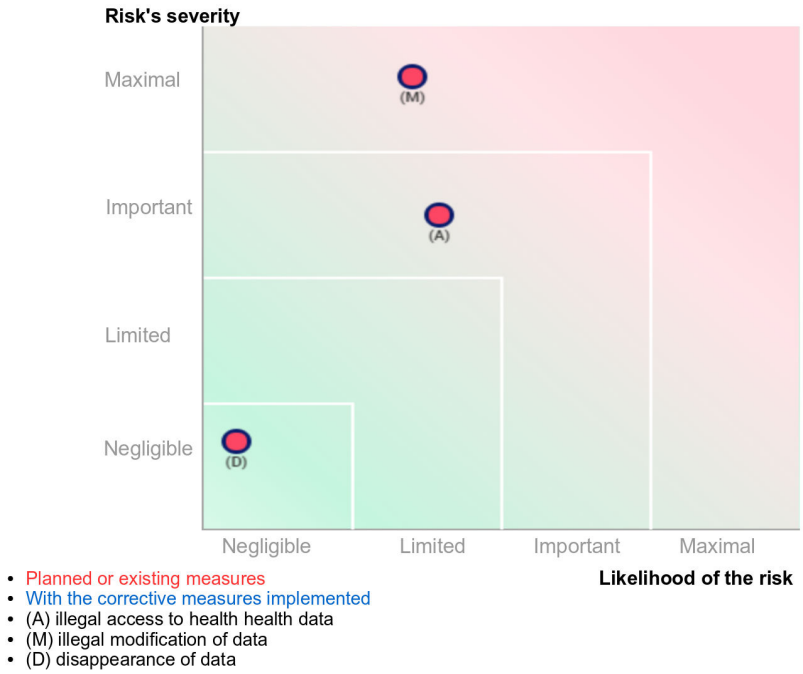


Figure 3.17 – Risk cartography

The Fig. 3.17 represents the risks generated by the process while taking into account all the corrective measures. It demonstrates that if the security measures described in the action plan are properly implemented, the likelihood and/or severity of the residual risks should be reduced.

3.9 Conclusion

In this chapter, we first introduced our global architecture allowing the enhancement of WebRTC's capacities by adding the possibility of interacting with IoT/WoT ar-

chitectures in order to provide rich and innovative use cases. Then, this architecture was illustrated using several use cases in the e-Health domain, in particular for the telemedicine services. Moreover, an original security layer was introduced, encompassing the confidentiality, the integrity, the authentication and the access control. We showed the feasibility of our approach by implementing a proof of concept. Finally, a privacy analysis regarding one of the proposed use cases was conducted in order to evaluate the potential risks. Furthermore, this architecture appears as a paradigm which can be used in numerous use cases occurring in smart cities, where real-time multimedia communications have to be coupled with data flows associated to IoT/WoT devices. In the next chapter, this architecture is integrated as a part of a real smart home, with the aim of providing medical and wellbeing services in the next generation smart homes.

Chapter 4 | Toward next generation smart homes

Contents

This chapter introduces the second contribution. It proposes a smart home architecture, managing mainly, but not only, two types of services: the healthcare services and the energy management one. Thus, the chapter starts with an introduction of the Home Area Network (HAN), since the smart home falls into this category of networks. Then, discusses the related works, followed by the newly designed architecture together with the main communication protocols deployed. Next, the different propositions, that were integrated in the smart home architecture, are presented. These propositions, are either the services implemented in the smart home (i.e. the healthcare, the energy, the visualization one), details regarding an important component (i.e. the smart gateway, the security layer and the reliability) or simply the algorithms used to test the propositions.

4.1 Introduction

This contribution proposes an implementation of the next generation smart houses, where heterogeneous data coming from multiple IoT sensors (medical, wellbeing, energy, contextual, etc.) and house equipment (smart fridge, smart TV, etc.) need to be managed, secured and visualized. As a first step, it focuses only on energy and health data. However, it aims to lay the foundations to manage any type of information towards the development of smart interactions with the house, which might include AI and machine learning. These data are securely collected using a central WoT gateway, located inside the smart home. For the e-health part, a set of possible use cases is provided, along with the current progress of the implantation. In this regard, the main idea is to be able to link the next generation smart houses with external medical entities in order to provide, first, quick intervention in the event of an abnormality being detected, and also to be able to provide basic medical services, such as remote consultations with a doctor for a particular health issue. This vision can be very promising, particularly in rural areas, where access to medical services is difficult. Furthermore, the e-Health services, in this case, represents the second type of services, which require real-time and multimedia capability. Thus, the architecture proposed in chapter 3 is integrated in this proposition.

As for the energy part, the aim is to be able to collect users consumption inside the smart house, which can be supplied from different sources (heat, water, gas, or electricity), and to be able to apply advanced algorithms in order to predict and manage this consumption and possible local production units. This approach uses data collected from smart meters, together with information on the operation of the devices (the status of smart plugs), user requests and network signals. By using a home energy management system (HEMS) that has these elements as input parameters, the operation of in-home devices and appliances can be optimally controlled according to different objectives (e.g., minimizing energy costs and maximizing comfort level). Furthermore, this part represents the first type of services provided by the traditional WoT architecture previously introduced in chapter 2 section 2.2.

Hence, this proposition presents first a new smart home architecture capable of managing several type of devices in order to provide the different service expected from the next generation smart homes, while integrating in an smooth way the architecture proposed in the chapter 3.

4.2 Home Area Network

This work is contextualized within the notion of Home Area Network (HAN), which is a type of IP-based Local Area Network (LAN). It connects multiple devices inside a small boundary, such as a house, and supports effective resource sharing among existing nodes. This concept is gaining popularity, especially with the emergence of the smart home paradigm, which requires advanced automation, efficient energy management, interoperability between the various devices, particularly constrained devices, and robust security and privacy management. From our perspective, the HAN network inside the smart home can be fairly divided into five main categories, as shown in Fig. 4.1.

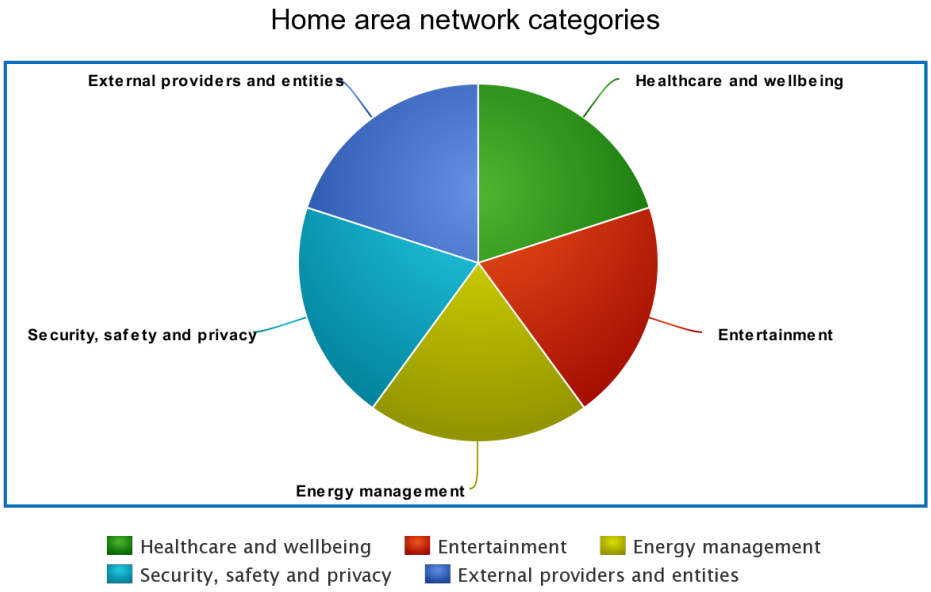


Figure 4.1 – HAN categories

4.2.1 Entertainment

It is represented by the various devices located in the smart home in order to provide comfort, joy and relax inside the home. The main types of smart devices on the market that deliver this kind of the service are:

- Multimedia: is the main entertainment service in any house. In the last decade, there has been an immense growth in multimedia traffic on the global network, due to the huge interest and demand in development and use of multimedia-based applications and services. Real-time multimedia applications, services, and solutions such as video conferencing, remote video on demand, streaming, online real-time content delivery, and online gaming, have contributed to the exponential expansion of the multimedia traffic on the Internet. The devices

can be, for instance, smart TVs, connected speakers, and a myriad of related appliances.

- Hubs and Controllers: these devices help in controlling the gadgets and appliances in the house, either by voice, such as Amazon Echo or with smartphone applications, such as Wink Hub, Samsung SmartThings Hub, Google Home, etc. They are characterized by their communication capacities, network support, enhanced security compared to the other IoT devices, and memory. The most advanced ones are based mainly on the IFTTT (If This Then That) web service, which enables the creation of simple instruction chains to connect applications, devices and services from different developers. They can be triggered in different ways, for example using voice.

- Lighting: the connected bulbs make it easy nowadays to control the light in the house to provide different light settings or even energy savings by intelligent dimming. Some examples of connected bulbs are the Xiaomi Yeelight [82] and Philips Hue [83] system among others.

- Heating and cooling: it has been proved that the use of a smart thermostat can bring energetic and economic savings. Furthermore, the most advanced ones have also the ability to learn from user habits and adapt accordingly. One of the most used is Nest [84].

- And other for: cooking, cleaning, outdoor robots, etc.

This kind of services, in particular the multimedia related ones, requires a greater network demand for low-latency, high capacity, real-time and high Quality of Services (QoS). What is more, they consume most of the network bandwidth and require higher processing and storage resources.

4.2.2 Healthcare and wellbeing

Health-monitoring systems in smart environments, such as smart homes, are gaining increasing attention in order to provide complementary solutions to traditional healthcare services. The advances in wearables, smartphones and medical sensors (blood glucometer, oximeter, blood pressure, electrocardiogram sensor (ECG), wearable sensors, etc.) allowed the possibility of collecting a large amount of real-time health data. Subsequently, these data can be processed in order to provide a comprehensive and predictive picture of an individual wellbeing and health, with the aim of maintaining better health outcomes and conducting early interventions to anticipate health needs.

4.2.3 Energy management

The main goal here is to reduce energy consumption costs and to increase the level of user's comfort regarding household task scheduling (TSC) and thermal comfort (TC) in the smart home. The main function of the Home Energy Management System (HEMS) is to optimally control the operation of in-home devices and appliances concerning different objectives (e.g., minimizing energy costs and maximizing comfort level). For this aim, it considers the measures provided by smart meters, the information on the operation of the devices, user's requests and external signals such as energy prices and meteorological information. The HEMS can also use information about the presence of people over a period and infer an occupancy model of the residents. This would enhance management of the system within the scheduling period.

4.2.4 Security, safety and privacy

In order to protect the resident of the smart home from any possible harm, several precautions need to be taken. It ranges from the safety of the residents in their homes to their security, while taking into account the privacy of their data:

- Safety and surveillance: since users are very concerned about their safety and the one of their family and their belonging, several solutions have been proposed on the market in order to respond to this need, which are mainly: smart surveillance cameras (i.e. Cisco), smart locks (i.e. August), smart trackers (i.e. Tile), smart firewall (i.e. Cujo), etc.
- Security and privacy: user's data must be protected from any external threats. For this reason, several measures need to be taken into account when building such a network, which are mainly:
  - Authentication: in order to verify the identity of the user.
  - Confidentiality and Integrity of the data: by encrypting all the exchanges and with the third-parties (i.e. external service providers).
  - Access control: so only authorized users will be allowed to access the services of the smart home. In this regards, access delegation should also be considered, as external persons coming to the smart home might need to have access to basic services inside.
  - Privacy analysis: must be carried out due to the huge amount of personal data that will be exchanged. Risks must be also studied to determine the security requirements that need to be taken into account from the design process, following the "Privacy by Design" principle.

4.2.5 External providers and entities

It concerns any communication that occurs with a node outside the HAN. These might comprise any service provider for the aforementioned categories, as well as complementary services, as shown in Fig. 4.2.

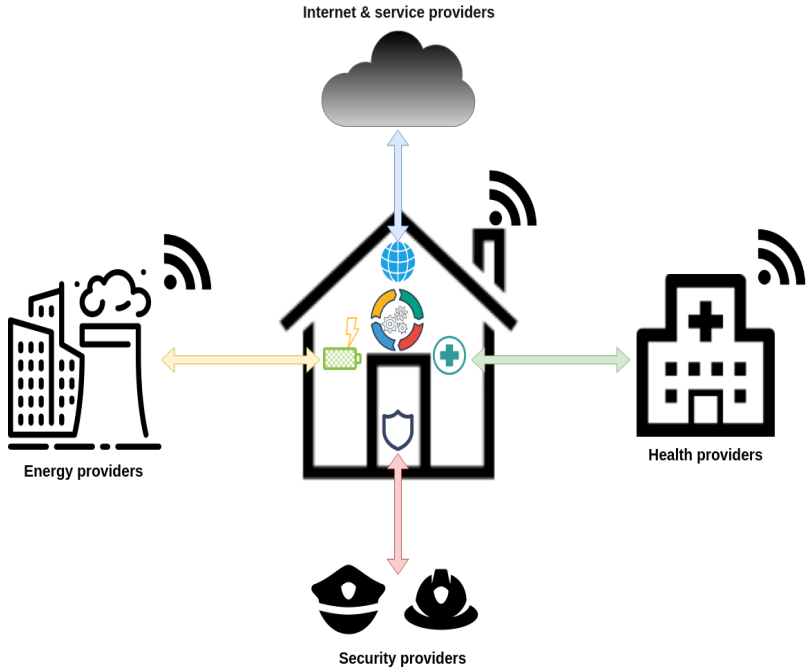


Figure 4.2 – Next generation smart home

In this work, we intend to provide a case of use applied to a real environment where the different components will be implemented in a pilot smart home in order to evaluate the effectiveness of our proposal. Particular attention will be paid to the healthcare, and energy management system, as well as to the security of the data related to these two services.

4.3 Related works

The related works is divided into two parts, since in this contribution, we focused on two types of services present in the next generation smart homes.

4.3.1 Healthcare

In the literature, several works have been done in order to provide remote healthcare services for the patients in their homes as in [85], where the authors present a cloud-based web service-oriented architecture relying on a ”home system” for the collection of information from a heterogeneous set of devices, and which includes the core elements of the REST service platform and many features required for the healthcare services to operate. However, the drawback of the work is that they do not deal with the data security issues, which is one of the main concerns of the users at any architecture dealing with health related data. In [86], the authors present their research work in a

project called SPHERE (Sensor Platform for Healthcare in a Residential Environment) developed in UK, with the aim of providing an interdisciplinary work to build a generic platform that merges complementary sensor data to generate rich datasets that support the detection and management of various health conditions. In particular, in the case of the ambient assisted living (AAL) and the case of continuous management of health conditions in smart homes. However, the only major drawback of the project, as far as we understood, is that they do not provide a deep and detailed security analysis for the overall system. In particular, there is no analysis regarding the access control in the system, and to the different IoT medical devices. Reference [87], presents a smart e-health gateway to be deployed for ubiquitous health monitoring systems especially in a clinical environment. The proposed gateway can manage various sets of smart medical objects that can be for instance present in the different rooms of a smart home or of a smart hospital. However, the development was focused only on the remote monitoring of the patients. Moreover, neither the privacy aspects nor the data visualization of the heterogeneous sources in a smart home was addressed in that work. Whereas [88] proposes an IoT-based remote health monitoring architecture (IReHMo), with the aim of taking care of the health of elderly people without compromising their convenience and preference of staying at their home. They also propose to solve the issue of the efficient transmission of healthcare data within the limit of the existing network infrastructure, especially in remote areas in Sweden, using CoAP. However, from the security point of view, they only deal with the encryption of the data, and other issues such as authentication and access control, in addition to the privacy, are not mentioned. In [89] , the authors propose a remote patient health monitoring in smart homes by using the concept of fog computing at the smart gateway. The data are gathered from the IoT sensors through the data acquisition layer, and analyzed at the fog layer. A notification is sent to the cloud layer only if an adverse event is triggered. However, they also do not consider the security and the privacy aspects, even though their architecture deals with sensitive health data. Authors of [90] presents an emotion-aware connected healthcare system. It captures speech and images of the patient in a smart home environment and then processes them in order to detect the different emotions such as pain. The latter may for instance trigger the intervention of a caregiver.

To summarize, the major drawback of the previous works is obviously the lack of deep analysis of the different security aspects, in particular regarding the access control. In our proposition, the architecture of the smart home considers the security as a priority by implementing a security layer at the application level and by encrypting all the traffic flowing between the different components.

4.3.2 Energy management

In smart home environment, where the consumers are able not only to produce/store energy from several local distributed means of generation (such as roof-top photovoltaic systems (PVs), small wind turbines, micro-combined heat and power units, and battery energy storage systems) but also to participate in demand response (DR) programs under different pricing schemes, the need for energy management tools becomes more

apparent. Over the past few years, many research works have been conducted in this subject, which mainly focus on the use of data from different energy resources and their optimal scheduling. In most of these studies, a standalone Energy management System (EMS) is implemented to collect data from the local resources in order to optimize their operation. As an example, authors in [91] propose a method for optimal operation management of smart home appliances considering the real-time energy prices. To this end, a mixed-integer quadratic-programming based predictive control scheme is proposed to optimally manage devices such as heating, battery storage, freezer, and dishwasher based on the user's economic objective. The research work in [92] considers the optimal energy management problem of a smart home during different weather conditions, where the ambient temperature fluctuations can highly affect the energy cost of the building. For this purpose, the thermal and electrical performance of the smart home is studied under price uncertainty while an information gap decision theory (IGDT) technique is used to tackle the optimal energy management problem in an uncertain environment. A state-of-the-art regarding energy visualization techniques for smart homes is presented in [93] where different techniques are compared in terms of their values and lacks, and then a visualization program for smart home environment is developed using a mobile application. The application allows the users to visualize the electricity consumption, to analyze the data and to draw optimization plan and intelligent feedbacks. However, the work is just a framework and future implementation is not considered. By the same token, a solution is proposed in [94] to reduce the energy consumption cost and the peak demand of a residential unit through integration of an EMS. The proposed design benefits from wireless communication and smart metering infrastructure in order to make an efficient scheduling for the electric appliances. The authors in [95], proposes a unified solution to manage the responsive loads within smart homes and the Plug-in Hybrid Electric Vehicles (PHEVs) inside the Internet of Energy (IoE) network using cloud computing in order to maintain the load stability in the smart grid all the times. Another work in [96] provides a comprehensive review on HEMS, with a particular focus on DR programs, the smart technologies used, and the load scheduling controllers. The problem of optimal residential energy management in both thermal and electrical zones is also tackled in [97] where the HEMS is designed to minimize the sum of energy cost and thermal discomfort cost in a long-term horizon. A Demand Side Management (DSM) model based on Time-of-Use (ToU) pricing and different algorithms such as the binary particle swarm optimization, genetic, and cuckoo search algorithms for scheduling the appliances in three residential cases (traditional homes, smart homes, and smart homes with renewable energy) is proposed in [98]. Additionally, the authors use on-site renewable energy and backup storage systems, in order to reduce the electricity bill and avoid high electricity price peaks. Authors of [99] describe an autonomous energy management-based cost reduction solution for peak load times using a HEMS, by controlling the smart home appliances linked to the smart meter. The proposed approach is based on Dijkstra algorithm, which reduces the complexity of the optimal scheme. In the same subject area, [100] proposes a self-learning home management system (SHMS) using computational and machine learning technologies in order to enhance the system capabilities such as price forecasting, price clustering and power alert system. The overall system is composed of a HEMS, a DSM

system, and a supply side management system, inside a real smart home in Singapore. As for the smart metering systems, [101] presents the integration of the advanced smart metering infrastructure (AMI) in the context of the smart building with an EMS in order to enhance the management capabilities of each individual consumer, using the daily energy profiles collected from this architecture. As a complement to the previous work, [102] explains an AMI for future smart homes which allows not only monitoring of the electricity, gas, water and heating consumption of the end-user, but also provide useful information regarding the power quality. As for the application layer, the authors of [103] propose a centralized system implemented in software application to manage the home appliances through a Wireless Sensor Network (WSN). It uses several management algorithms based on consumption models combining timing schedule, power, temperature or ambient light measurements and prioritization. The proposed system uses stochastic models for the prediction of household consumption and generates dynamically proper power limits depending on the day and the week periods. Also, the research work in [104] describes the development of an application to control the in-home energy consumption. The proposed application performs the management of energy by means of a wireless personal network based on ZigBee standard and predicts the demand using stochastic models hosted on a Web Server. Likewise, [105]proposes an application (mainly based on Zigbee communication) to manage the power consumption at home optimally by controlling household appliances and other loads by means of setting the maximum power suitable and a priority algorithm based on timing schedule, temperature or ambient light.

Although different aspects of energy management in smart home environments have been studied in the reviewed literature, the major drawbacks of the mentioned works are threefold: first, most of the designed infrastructures are based on local area networks where interoperability of various devices with different communication technologies cannot be supported. Moreover, majority of the developed architecture cannot interpret and process the sensed and measured information from different nodes at home. They also suffer from lack of data management systems to be run by the HEMS to manage and store contextual data for later retrieval. Last but not least, none-of the aforementioned research works deal with the security and privacy issues associated with implementation and realization of the energy management systems, where diversified physical sensing information must be stored, processed and dispatched to the actuator components and users.

4.3.3 Security in the smart home

In the next generation smart homes, both IoT and traditional devices and services are integrated in order to enhance the quality of life of citizens. However, it also raises some additional security threats due to the variety, the heterogeneity and the interdependencies of these devices. In [106], the authors investigate these issues, by providing a holistic approach of security together with recommendations and good practices for all the stakeholders involved in the smart home environment. The study focuses mainly on the IoT devices inside the smart home (which can be either constrained or with high capability), the interaction and data exchange with remote services and finally

the interaction and data exchange with mobile applications.

Thus, according to the ENISA [106], the good practices for a secure smart home environment, for all the stakeholders, which range from the industry and manufacturing actors to the end-user of the IoT device in the smart home, can be summarized in the following points:

- Development of smart home devices and services: it focuses mainly on one side on the security of the development process to mention the design, the development and the testing phases. These phases involves several security measures such as a privacy assessment, defense in depth and security audits and so on and so forth. On the other side on the security functions for hardware and software. The latter includes mainly the security audit, communication protection, cryptography, user data protection, identification, authentication, authorization and self-protection.

- Integration of devices into the home area network: it requires guaranteeing, first, a minimum level of reliability of the software and the hardware of the IoT devices, by being able to handle failures, errors and malfunctioning. Secondly, by introducing the trust notion and defining the level of trust for each device connected to the HAN, which can be achieved through a trust infrastructure. And finally, the network security. It relays mainly on the gateway in order to secure the smart home environment from internal and external attacks at the network level, by including for instance the firewall, IDS, IPS functionalities. Additional measures may include network segregation, using SDN and VLANs for instance, in order to ensure the security of the devices and services that do not need to interact with each other or outside the HAN or the local network.

- The usage until end-of-life: it concerns the good practices related to the devices from the installation until its disposal. It includes mainly the protection of data exchange in all the networks accessible to the devices, by ensuring first the user access right (explicit consent), by relaying on gateway in case of constrained devices, and by applying network segregation. Secondly, providing operation and maintenance of the security of the device through updates with regards to new vulnerabilities and during device management. And finally, the control of user data, by guaranteeing the right of erasure and backup of the different types of data related to the user in the HAN.

The security and the privacy are not the core of this proposition, since, the main goal of this work is the proposition of a new smart home architecture encompassing both the energy and the health services, a deeper security and privacy analysis is subject of future work. However, since we propose a security layer, which is positioned mainly in the security function for hardware and software category of the development of smart home devices and services, a state of art is provided.

Several works have been presented in order to manage the security and the privacy inside, between, and outside the smart home. For the access control we mention [107] which provides a security and privacy solution based on Blockchain. In the proposed

architecture, the smart objects, located inside the smart home, are managed via the smart home miner. It mainly handles all the communication (in Blockchain called "transactions") between them and also with outside the smart home. This miner which is part of the local Blockchain is employed in order to provide secure access control to these IoT devices. The BC block contains a policy header in order to authorize devices and to enforce the owner's control policy over his home. Moreover, the confidentiality and the integrity of the data is achieved through symmetric encryption (using Diffie-Hellman shared keys) and hashing. [108], the authors propose a framework to enforce IoT access control in the smart home at the network layer. It is based on the user's policy (via Discretionary Access Control (DAC)), the manufacturers of the IoT devices (via enforcing Mandatory Access Control (MAC)) and the security services (via dynamic access control). The proposed solution uses SDN, NFV and a security agent interacting with the northbound API of the SDN in order to monitor the traffic and mitigate the attacks targeting the smart home. The framework is illustrated by an NFV IPv4 ARP server as a security service in order to mitigate ARP spoofing attacks. The authors in [109] focuses more on the access control for the API-enabled IoT device in a smart building. In order to handle securely the constrained IoT devices, the complex security computation are handled by a third party security manager. The security manager is splitted into an authentication manager, which issues tokens to the verified users, and an access control manager, which is based on predefined access control policies (written with the XACML language) by the administrator for each API.

As for the authentication, [110] the authors propose the design of a user authentication key management protocol called UAKMP for generic networks such as the smart home, which they consider as a hierarchical IoT network (HIoTN). The schema is based on three factors in order to authenticate remote users in HIoTN deployments, which are user smart card, password, and personal biometric. In [111], the authors present a dynamic and energy aware authentication scheme for the Internet of Things (DAoT) in smart homes. The idea consists in dynamically selecting the suitable authentication mechanism for an IoT device based on its energy constraints. Also in [112], the authors propose an authentication service by enhancing the traditional authentication mechanism with the context of the user. The motivation is to be able to handle mobile environment, where users intervention is inquired for identity clarification and authorization. Thus, reducing the burden on the user for verifying information each time an access is requested. Using these contextual data (user profile, location, calendar, historical information) and the authentication credentials a confidence level is calculated in order to take an access decision. The [113] presents a voiceprint-based authentication for remote users in order to access their smart home environment with their mobile phones.

As a result, the main observation is that there is a need for standardized mechanisms in order to manage the security in the smart home. Moreover, the integration of the security and privacy in the design phase is still not mature enough, or even missing in most of the cases, an particularly for the smart home environment.

4.4 Architecture

The smart home vision aims to enable, to a certain extent and in an independent and easy way, a better interaction with the user, in order to improve the quality of life of the residents. This work represents real experiments carried out in a smart home built at Aalborg University, Denmark.

4.4.1 Architecture layers

The architecture is a preliminary design toward such vision. The proposed architecture is composed of four layers, as shown in Fig. 4.3:

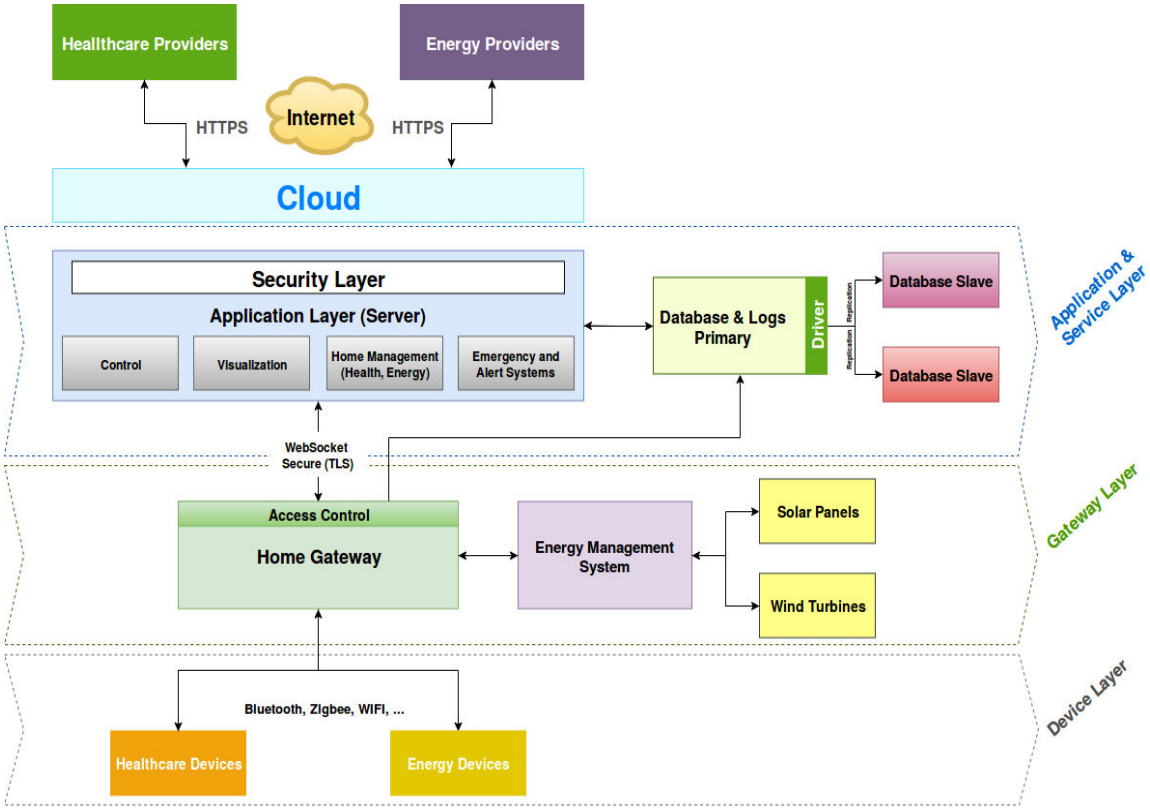


Figure 4.3 – next generation smart homes

The Device Layer: represented by all the IoT devices in the smart home, which can be sensors, actuators or even connected smart home appliances. In this contribution, a special focus is given to the medical IoT devices and smart plugs in order to measure the energy consumption of the different components. Moreover, renewable energy sources are also equipped with sensors that measure the current, voltage, power and send the data using the ZigBee protocol. These data can be then visualized or used in the EMS at the application layer for different control and forecast strategies. Moreover, the smart meters are used in order to collect the energy, heat, and water consumption inside the smart home, and then to send them to the data concentrator located in the smart home gateway.

The Gateway Layer: represented by the smart home gateway. It provides the ability to communicate with the different smart devices in the smart home (energy, health

and smart meters), using different communication protocols (e.g. Bluetooth, WIFI and Zigbee). Details about the gateway will be provided in section 4.4.3. Different stack of communication protocols were deployed in order to enable the interaction with different kinds of IoT devices. The gateway also interacts with databases in order to store the data coming from all the sensors, errors, alerts and operational logs. From this layer a secure web socket is established between the application server and the home gateway in order to exchange data and commands in a secure way. Both components are located in a private local network of the smart home and connected to a secure WiFi.

The Application and Service Layer: provides the main services of the smart home, in particular, the ones related to the health and energy, together with the possibility of interacting with external entities, and controlling different smart home appliances. Through a simple visualization interface, it provides all the data collected from the different sensors of the smart home and the possibility of controlling and managing them. Another important feature is the ability to collect alerts and error notifications and to react accordingly. A security layer is added on top of the application layer so users have to authenticate in order to be authorized to access certain resources and data for each user. What is more, all data exchanges are encrypted using AES128 to provide confidentiality and integrity. A Node.js server providing a Web API is deployed. This server can either extract real-time data directly from the gateway for the case of remote tele-consultation as will be explained later, or query a database for historical records. Thanks to the selected development technologies, all these services can be deployed on the Cloud in a seamless way. Moreover, in order to receive notifications and alerts from some particular sensors (in particular from the most critical ones), a simple android application was developed. The application uses Bluetooth Low Energy (BLE) in order to communicate with the gateway, and can either read, write or subscribe to a particular sensor to receive notifications. For the energy part, the EMS is used in the application layer in order to infer control strategies from the measured data (i.e. sensors and smart meters data collected through the utility) and use the smart plugs to carry them out.

The third-party providers are added in order to connect the smart home with the different entities and service providers in the city. It can be also seen as a tentative to build a smart city network, where each smart home is connected to the different smart facilities of the city. For instance, and as we will explain later, a connection to the healthcare provider can be established to allow for quick interventions in emergency situations, as well as another service for weather forecasting, used by the EMS to estimate the daily consumption profile.

4.4.2 Communication protocols

In the context of the smart home, every object, device, home equipment, etc., needs to be able to exchange information with multiple devices and to be connected to the local network of the smart home, in order to interact and control them in an efficient way. However, there are currently no common standards and specifications regarding a unified communication protocol for all these devices. The same myriad of protocols can

be found for the IoT. Hence, issues such as interoperability, compatibility and security still exist and require more efforts to achieve such goals [114]. Efforts are made by the WoT community [30] in order to abstract all the complexity of the connectivity part of the smart objects, by providing a standard application layer based on Web standards to simplify the creation of IoT applications.

In our experiments, a special focus was given to the most widely-used wireless personal area communication protocols such as BLE and Zigbee, as well as WIFI for less constrained devices and Ethernet as a wired protocol. Mainly, a gateway, exposing a REST API and implementing several communication stacks, is used to interact with the different smart objects.

4.4.3 The smart gateway

The interoperability issues can be solved, by using a secure gateway, implementing the main communication stacks in order to be able to communicate with all the devices. The gateway also acts as a data aggregator that processes data traffic coming from different devices in the home network, independent of the means of physical transmission, i.e., wired or wireless. Since the gateway is able to communicate either by using a wireless communication protocol (i.e. Zigbee, Bluetooth, etc.) or by using Ethernet directly linked to the devices. The data provided by this gateway will be then sent to the application server.

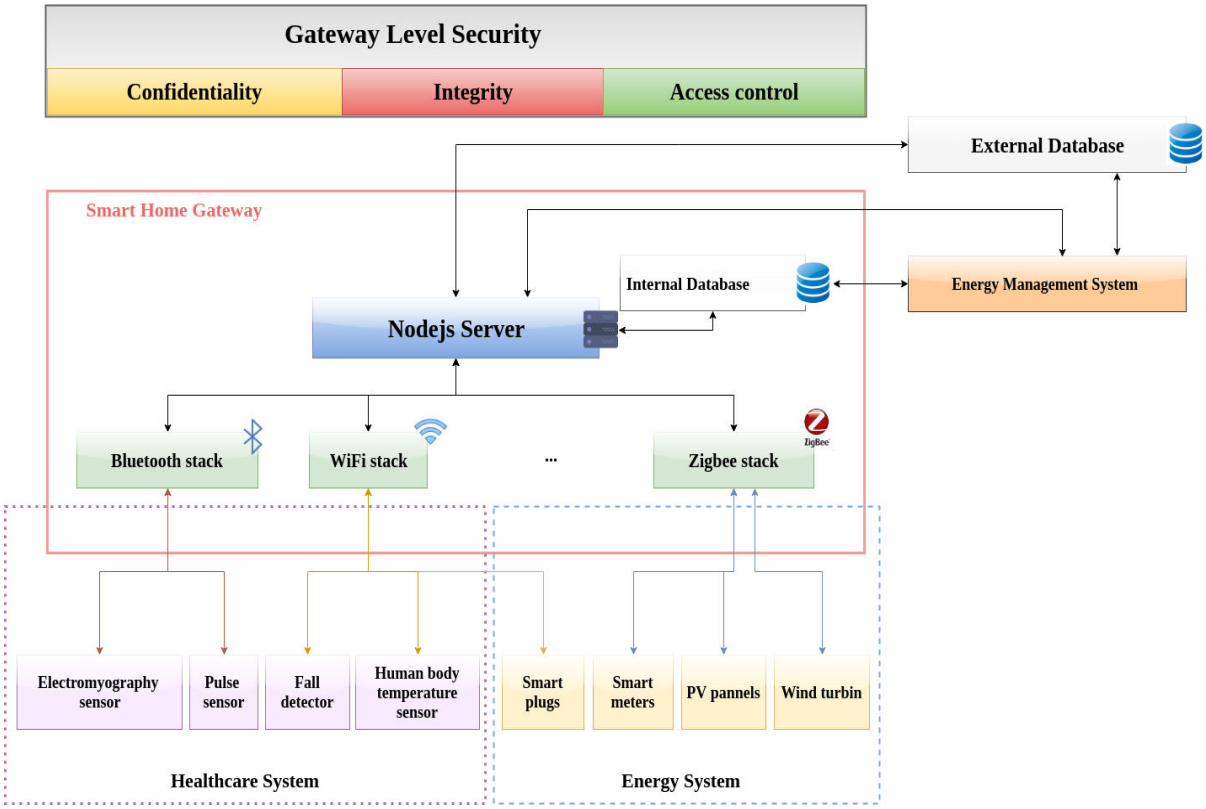


Figure 4.4 – next generation smart home gateway

As shown in Fig. 4.4, the smart gateway is a Node.js server running inside a raspberry pi 3, for experimental and open source purposes. The raspberry is also connected to

other gateways in the smart home in order to have a full view and to manage them all in a centralized way. The gateway provides the possibility to interact with various devices using different communication protocols toward an interoperable architecture. Since the scope of the contribution is narrowed down to the health and energy services, the connected devices can be divided into two categories: the health-related medical devices, such as pulse sensor, muscle sensor, temperature sensor and fall detection sensor using an accelerometer, and the energy-related devices, which can be smart plugs, energy harvesting sensors or smart relays. Moreover, for each sensor, an actuator is attached. The idea is that if an abnormality is detected, a command is sent to the actuator in order to perform some actions. For Proof of Concept (PoC) purposes, "Servo Motors" and "LEDs" are used. The gateway is also connected directly to a database in order to store the data coming from the sensors, the errors and the alerts. A security layer is added to the gateway in order to protect the data against unauthorized access (using an access control) and against eavesdropping and tampering (using encryption).

4.4.4 Security layer

Confidentiality and Integrity: in all the cases, and in particular in sensitive domains, such as the health, data exchanges between the parties need to be kept confidential so only the authorized ones are able to access them. A falsified or altered message must be detected. The most commune solution is to use strong encryption algorithms and strong security protocols that implement those algorithms. In our PoC, and as recommended in the RFC3565 [115], AES128 encrypts the data exchanged between the gateway and the application server, using pre-shared keys.

Authentication: in order to verify the identity of the user. For the proof of concept, an authentication middleware called "passport" [116] is used. Passport is a simple, unobtrusive authentication for Node.js. It provides several authentication strategies, in our architecture a local authentication strategy is used, which is based on user credentials (username, password). However, other strategies can be developed later, in order to provide more robust authentication schema such as using single sign-on using an OAuth provider (i.e. Facebook or Twitter), or using identity providers, as explained in section 3.5.2.

Access control: in this contribution RBAC is chosen in order to provide a PoC. ABAC seems more suited for the general smart home use case, however, further studies need to be done in order to determine the best model. The access control is done in the gateway, after a successful authentication of the user, the application server requests the gateway if the user is authorized to access the resources or not. If yes, the user will be redirected to the web page containing all the data collected from the gateway. Otherwise, an error will be raised.

4.5 Services

4.5.1 Healthcare services

For the health services, the idea is to be able to provide remote health services for the patients in their homes, through several use cases, such as:

Wellness follow-up: the patients can use the medical devices in their home in order to follow their health status for wellness purposes through the visualization interface. For instance to check the body temperature, or to regularly check the cardiac parameters such as the Heart Rate.

Health issues self-follow-up: where the patient with chronic disease, such as diabetic, can follow its health status daily using its medical sensors. In case of detection of an abnormality, two cases are presented here: For independent patients, they can call the medical service (a doctor for instance). If the patient is dependent, in this case, either simple solutions for calling the medical service such as a button can be used, or an automatic abnormality detection algorithm can notify the medical service.

Remote medical consultation or Tele-consultation: patients can consult their doctors remotely, using multimedia enhanced with IoT contextual health data [74]. This can be a solution to solve rural areas issues regarding the access to medical services. This solution can also solve the problem of medical follow for elderly and persons with chronic diseases in rural areas, where usually they do not need to be physically present in front of the doctor in order to have the follow-up. Hence, reduce the stress, tiredness, time and money for them.

Remote monitoring: where the patient is usually equipped with medical connected devices, and in case of detection of an abnormality in the behavior or the vital signs of the monitored persons, an alert is sent to the closest medical service [74].

Additionally, one of the main issues is to be able to notify the concerned parties in case of detection of an abnormality in the behavior of the user (e.g., fall detected) or from the collected data. In our PoC, all the alerts and errors are sent to a central database, which can be for instance the database of the hospital. Moreover, solutions such as sending emails and SMSs are feasible and can be included in future work.

4.5.2 Energy management services

As it will be explained later in Fig. 4.17, the proposed HEMS introduces a hybrid AC/DC platform with different means of local generation (such as wind turbine (WT) and roof-top PV panels) and energy storage as well as controllable loads, here named as IoT devices (e.g., smart washing machine, tumble dryer, freezer, etc.). Moreover, the platform provides practical, optimized and comfort-aware energy management solutions for residents using real-time control and monitoring systems.

4.6 Services testing and reliability

The previously mentioned services are then tested through a set of algorithms in order to validate our approach, for instance in order to detect new alerts (e.g. fall detection). Moreover, since we argue that the proposed services must be reliable, a reliability discussion, mainly regarding the databases and the different sensors is provided as a first step. Future works may provide a deeper discussion regarding these two aspects.

4.6.1 Algorithms

In order to validate our approach, we had to test several use cases inside the home. We developed two elementary algorithms (which can be improved in future work) related to the e-Health (fall detection and temperature evaluation).

4.6.1.1 Fall detection

Several proposals and studies have used accelerometers to objectively monitor a range of human movement [117] [118], for instance, to measure metabolic energy expenditure, physical activity levels, balance and postural sway, gait, and also to detect falls. These systems have usually a hardware part, which is attachable to the body of the patient (usually around the waist), and a microcontroller, such as Arduino in our case, which is used to classify the person's actions and to detect any possible falls.

In our demonstration, a three-axis accelerometer ADXL330 board is used. Additionally, a simple algorithm, Algorithm1, is used in order to detect quick variation of one of the values of the accelerometer (X or Y or Z axis), and upon exceeding of a certain limit, an alert is generated and sent from the controller to the application. A LED is lit when a fall is detected. Additionally, a button also can be used in order to prevent fault positives, where the user is given 30 seconds to push the button in case of normal movement. A calibration function is used in order to detect the maximum values of each axis of the accelerometer. The value in this case are collected every 1 millisecond, as long as the calibration button is pushed, and usually it takes up to 1 second to get the maximum values for each axis. Generally, the accelerometer values are collected each 2 seconds.

Data: X axis
Data: Y axis
Data: Z axis
Data: X_{old} = previous X value
Data: Y_{old} = previous Y value
Data: Z_{old} = previous Z value
Data: X_{max} = Max variation of X axis
Data: Y_{max} = Max variation of Y axis
Data: Z_{max} = Max variation of Z axis
Result: Fall detection
Calibration X_{max}, Y_{max}, Z_{max};
while *True* **do**
    read X;
    read Y;
    read Z;
    if $|X - X_{old}| > X_{max}$ OR $|Y - Y_{old}| > Y_{max}$ OR $|Z - Z_{old}| > Z_{max}$ **then**
        A fall is detected ;
        Led On;
        if *Button not pushed after 30 second* **then**
            Send Fall detection Alert;
        else
            Led Off;
        end
    else
        Store these values as old values;
        Wait for 2 seconds;
    end
end

Algorithm 1: Simple fall detection algorithm using an accelerometer

4.6.1.2 Wellbeing monitoring sensors

For the data collected from these two medical devices, an evaluation is done in order to decide whether the measured data is normal or not. However, a classification of the data needs to be done to contextualize the information. For the pulse sensor, depending on the age and the activity of the user, measured data can have different meanings in each case. For instance, for a person aged more than 12 year, for a normal activity, the normal heart rate is between 60 and 100 bpm (beat per minute) [119]. As for the thermometer, for babies and children, the average body temperature ranges from 97.9°F (36.6°C) to 99°F (37.2°C), for adults the average body temperature ranges from 97°F (36.1°C) to 99°F (37.2°C), and for adults over age 65, the average body temperature is lower than 98.6°F (36.2°C). Hence for instance a temperature higher than 100°F (37.8°C) (mouth readings) is a sign of a fever.

For our demonstration, a value which is considered as abnormal triggers an alert which will be sent to the application layer and visualized to the user. In this case, the concerned age segment was for adults, and a temperature above 40°C is considered abnormal. Moreover, the idea is to send this alert to the related health provider either

through a notification. SMS can be also sent to the relative of the person in particular if the concerned person is a dependent one.

4.6.1.3 Energy Monitoring and Control

The operational pipeline in this smart home platform is the EMS that controls the operation of in-home devices and appliances optimally with regards to different objectives considering the information about task operating status, user's requests and network signals received through the AMI system. The AMI by itself includes two layers namely physical and utility layers. The physical layer includes the smart meters and the data concentrator that collects the measurements periodically and provides a dataset of data with synchronized timestamps using a Network Time Protocol (NTP) server. The communication between the smart meters and the data concentrator is based on the standard EN 13757-5 that implements a radio mesh topology. The second layer in AMI topology integrates the logical software provided by the Kamstrup OMNIA suite, to ensure efficient interoperability with the AMI network. This software communicates over Ethernet with the data concentrator and implements all the back-end processes that are necessary to configure the system, perform on-demand operation readings and capture the periodic records.

To keep a meaningful balance between the energy saving (Obj_1) and a comfortable lifestyle (Obj_2), EMS should solve a multi-criteria decision-making problem as [120]:

$$Obj_1\{Min.EnCost = f(\rho_{el}^t, \rho_{gas}, P_{grid}^t, P_{DG_i}^t)\} \forall t \in T, i \in N_{DG} \tag{4.1}$$

Where $EnCost$ denotes the energy consumption cost of the smart home which is a function of power exchanged between the building and the utility at time t (P_{grid}^t), power produced by domestic generation unit i ($P_{DG_i}^t$), and the real-time grid electricity and natural gas prices, respectively (ρ_{el}^t and ρ_{gas}). Moreover, the second objective which is formed based on the user's convenience level about household task scheduling (C_{task}) and thermal comfort level ($C_{thermal}$) can be modeled as follows:

$$Obj_2\{Max.Comf = f(C_{task=}, C_{theraml})\} \tag{4.2}$$

$$C_{task} = \Sigma_{t\in T}\Sigma_{j\in M}(\omega(j)SL(t,j)) \tag{4.3}$$

$$C_{thermal} = \Sigma_{t\in T}CL(T_{in}(t)) \tag{4.4}$$

Where, $SL(t, j)$ is the user's satisfaction level when task j is executed at time t and $CL(t)$ is the level of thermal comfort (regarding the indoor temperature T_{in}) observed by the inhabitants at each time step. More detailed information in this regard can be found at [121, 122].

4.6.2 Reliability

Undetected Sensor: the system should detect when a sensor is not connected. In this case, an error is raised and sent to the application, as shown in Fig. 4.5. An unavailable sensor, for certain reasons, can have serious consequences, where for instance

in case of home monitoring of a patient, if the fall detection sensor is not available, the fall cannot be detected, or in case of a smart plug which is not working, wrong energy consumption values will provide. In the current state of the work, our system detects if a sensor is missing and send an alert. This alert is displayed on the user visualization interface and registered in the database.

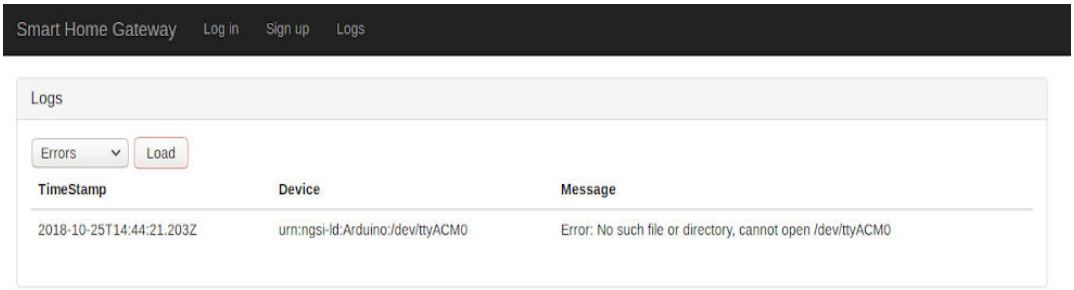


Figure 4.5 – Undetected sensor in the log

Database fault tolerance: the architecture implements the NoSQL database MongoDB. In order to provide more reliability to the system in particular regarding the data provided by the different sensors of the smart home, we used what is called "replica set" in MongoDB. As defined in [123], "A replica set in MongoDB is a group of mongod processes that maintain the same data set. Replica sets provide redundancy and high availability, and are the basis for all production deployments". In practice, replication provides redundancy and increases data availability, and to be able to recover in case of a disaster. By creating multiple copies of data on different database servers, replication provides a level of fault tolerance against the loss of a single database server, in addition to better performances, increasing data locality and availability in particular for distributed services. The implantation used the master-slave (primary-secondary) replication, as shown in Fig. 4.3, where the primary server receives all the writes and reads requests, and then replicate the data to the slaves. In the same way, if a primary node is down a reelection is triggered between the members of the replica and if possible, a new primary is selected. The Fig. 4.6 and Fig. 4.7 show this behavior of the primary database and the secondary one respectively when the primary server is down.

```
2018-10-26T13:45:45.122+0200 I REPL     [conn10] Stopping replication storage threads
2018-10-26T13:45:45.123+0200 I FTDC     [conn10] Shutting down full-time diagnostic data capture
2018-10-26T13:45:45.125+0200 I STORAGE  [WTOplogJournalThread] oplog journal thread loop shutting down
2018-10-26T13:45:45.125+0200 I STORAGE  [conn10] WiredTigerKVEngine shutting down
2018-10-26T13:45:45.320+0200 I STORAGE  [conn10] shutdown: removing fs lock...
2018-10-26T13:45:45.321+0200 I CONTROL  [conn10] now exiting
2018-10-26T13:45:45.321+0200 I CONTROL  [conn10] shutting down with code:0
2018-10-26T13:46:41.739+0200 I CONTROL  [main] ***** SERVER RESTARTED *****
2018-10-26T13:46:41.757+0200 I CONTROL  [initandlisten] MongoDB starting : pid=8187 port=27012 dbpath=/home/ubuntu/Documents/talent/ehealth-server-saad/mongoDBconfig/data/2 64-bit host=ubuntu-VirtualBox
2018-10-26T13:46:41.758+0200 I CONTROL  [initandlisten] db version v3.6.3
```

Figure 4.6 – Primary database log

```
2018-10-26T13:45:50.342+0200 I ASIO     [NetworkInterfaceASIO-Replication-0] Dropping all pooled connections to 192.168.1.20:27012 due to failed operation on a connection
2018-10-26T13:45:50.342+0200 I REPL_HB  [replexec-38] Error in heartbeat (requestId: 182451) to 192.168.1.20:27012, response status: HostUnreachable: Connection refused
2018-10-26T13:45:50.342+0200 I ASIO     [NetworkInterfaceASIO-Replication-0] Connecting to 192.168.1.20:27012
2018-10-26T13:45:50.342+0200 I ASIO     [NetworkInterfaceASIO-Replication-0] Failed to connect to 192.168.1.20:27012 - HostUnreachable: Connection refused
2018-10-26T13:45:50.342+0200 I ASIO     [NetworkInterfaceASIO-Replication-0] Dropping all pooled connections to 192.168.1.20:27012 due to failed operation on a connection
2018-10-26T13:45:50.342+0200 I REPL_HB  [replexec-35] Error in heartbeat (requestId: 182453) to 192.168.1.20:27012, response status: HostUnreachable: Connection refused
2018-10-26T13:45:54.056+0200 I REPL     [replexec-39] Starting an election, since we've seen no PRIMARY in the past 10000ms
2018-10-26T13:45:54.057+0200 I REPL     [replexec-39] conducting a dry run election to see if we could be elected. current term: 5
2018-10-26T13:45:54.058+0200 I ASIO     [NetworkInterfaceASIO-Replication-0] Connecting to 192.168.1.20:27012
2018-10-26T13:45:54.058+0200 I ASIO     [NetworkInterfaceASIO-Replication-0] Failed to connect to 192.168.1.20:27012 - HostUnreachable: Connection refused
2018-10-26T13:45:54.059+0200 I ASIO     [NetworkInterfaceASIO-Replication-0] Dropping all pooled connections to 192.168.1.20:27012 due to failed operation on a connection
2018-10-26T13:45:54.059+0200 I REPL     [replexec-36] VoteRequester(term 5 dry run) failed to receive response from 192.168.1.20:27012: HostUnreachable: Connection refused
2018-10-26T13:45:54.059+0200 I REPL     [replexec-36] VoteRequester(term 5 dry run) received a yes vote from 192.168.1.20:27011; response message: { term: 5, voteGranted: true, reason: "", ok: 1.0,
operationTime: Timestamp(1540554344, 3), $clusterTime: { clusterTime: Timestamp(1540554344, 3), signature: { hash: BinData(0, 0000000000000000000000000000000000000000), keyId: 0 } } }
2018-10-26T13:45:54.059+0200 I REPL     [replexec-36] dry election run succeeded, running for election in term 6
2018-10-26T13:45:54.067+0200 I ASIO     [NetworkInterfaceASIO-Replication-0] Connecting to 192.168.1.20:27012
2018-10-26T13:45:54.069+0200 I ASIO     [NetworkInterfaceASIO-Replication-0] Failed to connect to 192.168.1.20:27012 - HostUnreachable: Connection refused
2018-10-26T13:45:54.069+0200 I ASIO     [NetworkInterfaceASIO-Replication-0] Dropping all pooled connections to 192.168.1.20:27012 due to failed operation on a connection
2018-10-26T13:45:54.069+0200 I REPL     [replexec-35] VoteRequester(term 6) failed to receive response from 192.168.1.20:27012: HostUnreachable: Connection refused
2018-10-26T13:45:54.071+0200 I REPL     [replexec-40] VoteRequester(term 6) received a yes vote from 192.168.1.20:27011; response message: { term: 6, voteGranted: true, reason: "", ok: 1.0, operationTime:
Timestamp(1540554344, 3), $clusterTime: { clusterTime: Timestamp(1540554344, 3), signature: { hash: BinData(0, 0000000000000000000000000000000000000000), keyId: 0 } } }
2018-10-26T13:45:54.071+0200 I REPL     [replexec-40] election succeeded, assuming primary role in term 6
2018-10-26T13:45:54.071+0200 I REPL     [replexec-40] transition to PRIMARY from SECONDARY
2018-10-26T13:45:54.071+0200 I REPL     [replexec-40] Entering primary catch-up mode.
2018-10-26T13:45:54.071+0200 I ASIO     [NetworkInterfaceASIO-Replication-0] Connecting to 192.168.1.20:27011
2018-10-26T13:45:54.071+0200 I ASIO     [NetworkInterfaceASIO-Replication-0] Connecting to 192.168.1.20:27012
```

Figure 4.7 – Secondary database log

4.7 Implementation results

4.7.1 Hardware Implementation

Figure 4.8 – Smart home appliances attached to smart plugs

As far as the energy devices are concerned, the smart home is equipped with several home appliances as shown in Fig. 4.8. Their energy consumption is monitored using smart plugs from the Danish company Develco. These smart plugs use ZigBee as the communication protocol and are connected to the energy home gateway.

Figure 4.9 – Kamstrup Smart meter

In addition to the smart plugs, a smart meter from the Danish manufacture Kamstrup was also deployed for measuring the total consumption. This model is widely installed in the Danish homes. The meter is connected to the energy home gateway using a ZigBee module for local measures as well as to a data concentrator, which provides the utility with the demand profiles for billing purposes. The energy home gateway, shown above the smart meter on Fig. 4.9, was also manufactured by the Danish company Develco and contains the embedded ZigBee coordinator attached to an ARM9 CPU which runs a Linux OS. The system is fully flexible and programmable so a JAVA application was deployed. This application exposes a RESTFul API that allows for the data acquisition and actuation on the smart plugs.

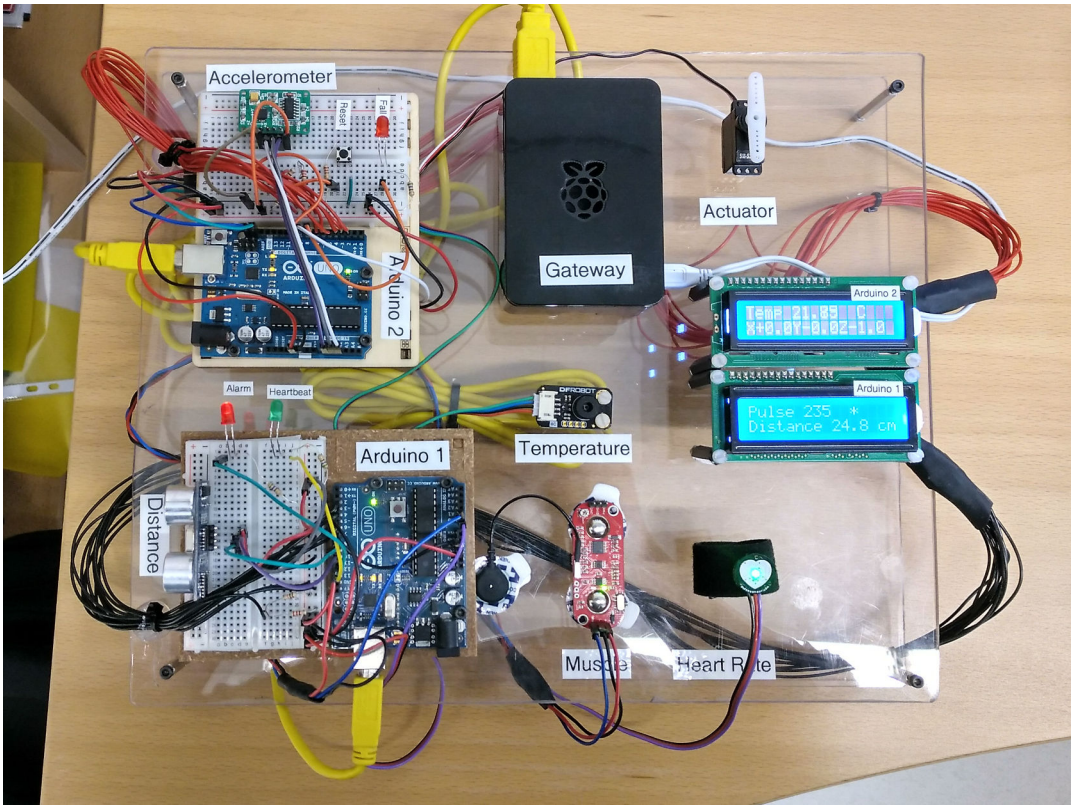


Figure 4.10 – eHealth platform hardware

On the wellbeing part, a PoC e-Health platform was developed as shown in Fig. 4.10. This system can be used by the patient on a daily basis to monitor his/her health state. It contains several sensors in order to measure the vital signs of the patient. These sensors are connected to two Arduinos UNO. The Arduino 1 measures the heart rate of the patient and the muscle activity. It also includes an ultrasound sensor that detects the presence of the patient next to the platform. The system shows the heartbeat frequency using the green led while the red one shows an abnormality on the patient pulse. The Arduino 2 is responsible for measuring the body temperature and the acceleration in order to detect a fall. The system is connected to an actuator that simulates an alarm based on an abnormal body temperature. It also includes a red led that indicates a fall alarm, which can be reset by means of the provided button. Moreover, two LCD displays are included to provide the user with a local interface.

All of these sensors are linked to the smart home gateway represented by the raspberry pi. This gateway gathers the data from the two arduinos using the USB ports. In addition, it also implements the communication with the energy home gateway.

4.7.2 Data visualization

The application layer provides a simple visualization web page to the user with simplified data of each sensor used inside the smart home, and with graphs representation of these data. Since the smart home gateway collects various types of data, it was judged better to separate the visualization of each type in a different page. For instance the health data are displayed in the “Health data visualization” section, as shown in Fig.

4.11, and the energy data are displayed in the “Energy data visualization” section, as shown in Fig. 4.12. Moreover, the “Tele-consultation” section allows the establishment of multimedia session with a remote peer (e.g. a doctor).

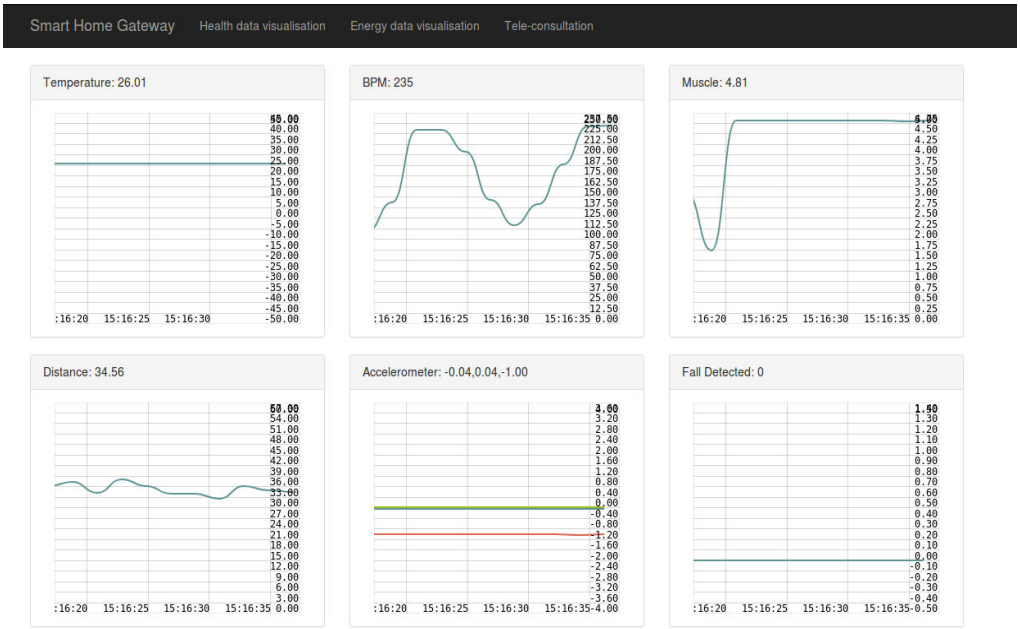


Figure 4.11 – Health data visualization

Figure 4.12 – Energy data visualization

Another portal is added in order to visualize the energy consumption of the different smart home appliances, measured by the smart plugs, all along the day as shown in Fig. 4.13.

Figure 4.13 – Energy consumption from the smart plugs

Additionally, a login/sign-up portal, Fig. 4.14 and other dedicated pages are provided especially for the logs (ie. Fig. 4.15), errors and the alerts detected (i.e. in case of an abnormality) inside the smart home.

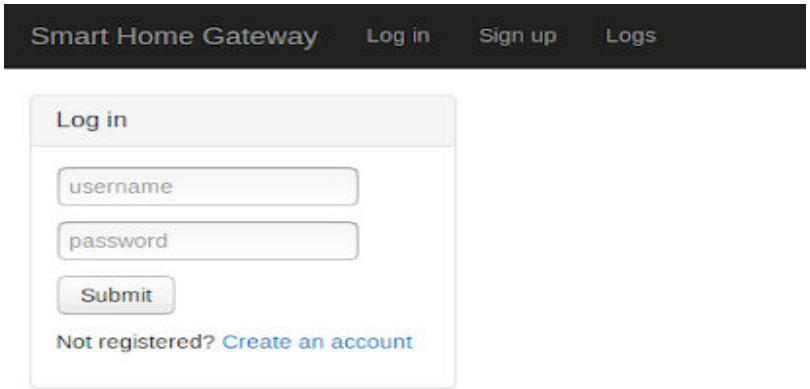


Figure 4.14 – Log page

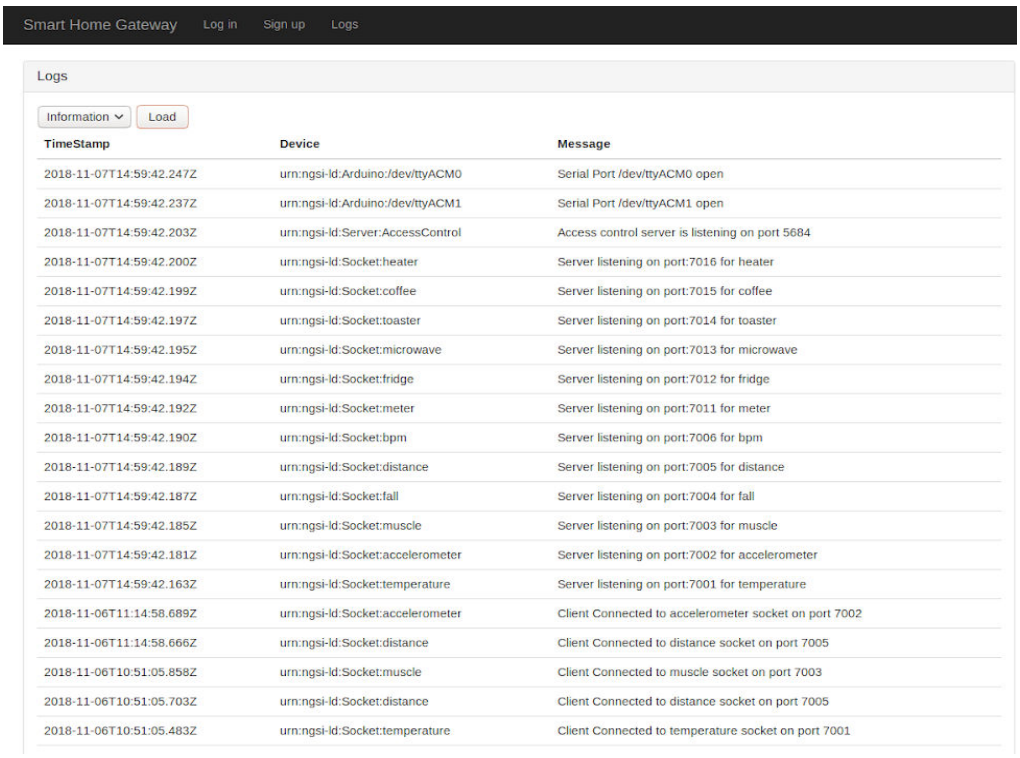


Figure 4.15 – Log page

4.7.3 Tele-consultation implementation

In our demonstration, a special focus is given to the tele-consultation service. Providing the ability to have remote consultation with a remote medical staff, such as a doctor, is one of the health services that can revolutionize the next generation smart homes, equipped with medical devices. These medical devices can be then used to measure the health status of the patient and to provide insight to the remote doctor in order to provide a better diagnosis for the patient. As explained in details in [74], a proposition is to use WebRTC based architecture enhanced with contextual information of the medical IoT sensors.

As shown in Fig. 4.16, the patient starts a WebRTC [13] multimedia session with the remote doctor, and in the same time, the wearable medical devices attached to the body of the patient, and also connected to the platform, starts sending real-time health data. Thus, using the information from the direct interaction with the patient and also the information provided by the medical sensors, we believe that the doctor can provide a better remote diagnosis of the state of health of the patient, and thus better remote medical services.

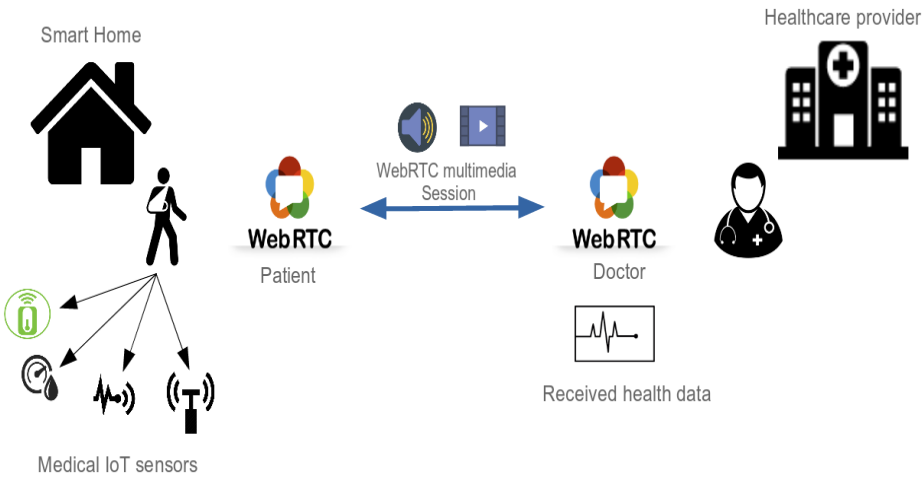


Figure 4.16 – Remote medical consultation in the smart home

4.7.4 HEMS Implementation

In this section, the architecture of an intelligent EMS is described for a smart home application. As shown in Fig. 4.17, the proposed home energy management system (HEMS) introduces a hybrid AC/DC platform with different means of local generation (such as wind turbine (WT) and roof-top PV panels) and energy storage as well as controllable loads, here named as IoT devices (e.g., smart washing machine, tumble dryer, freezer, etc.).

The physical layer in this platform includes smart plugs and activity sensors such as motion sensor, as well as smart meters. These sensors and devices use different communication protocols including Wi-Fi, Bluetooth and Zigbee to communicate their data to a Gateway. In parallel, the smart meters establish a radio mesh communication with data concentrator to provide the energy measures.

In the data link layer, two parallel tasks are carried out simultaneously. On one side, the gateway collects all the information of the smart plugs and sensors in the MongoDB database. It also provides the necessary interfaces for the actuation commands. On the smart meters side, the data concentrator establishes an encrypted communication with the metering head-end which in this case was tested locally but that usually belongs to the utility company. This head-end system is part of the Kamstrup Omnia Suite and provides a short-term and a long-term storage for the energy information. Another MongoDB database is used to store the on-demand measures as well as auto-collection data for a period up to 3 days. Behind this, a Microsoft SQL database records all the historical measurements. Nevertheless, these measures are always accessed by means of an API provided by the manufacturer.

The application layer is mainly composed by the service to access the HEMS and perform the control strategies. The HEMS is conceived as an external service the user can subscribe to. This system has to be provided it with the necessary input data and in return it will response with the necessary control actions according to the algorithms

presented in section 4.6.1.

Furthermore, there exist are two main application, which use the data-driven services of a smart home. The first one, which is a LabVIEW-based application, is used for querying on-demand readings from the Advanced metering infrastructure (AMI) network, logging system's alarms, events or malfunctions. This service could also be used to provide a simple and easy-to-use user interface, with all the significant data. The second application which is the control unit collects all the required information from the energy server and sends optimal reference signals and set-points to related actuators and device-level controllers. This application also provides a supervisory control over the system performance and changes the plan of actions in different working conditions (e.g., normal or faulty) based on a predefined scheme.

The HEMS was developed as a JAVA EE application also interfaced with GAMS solvers in order to carry out the optimization algorithms. In order to complete a user optimization request, the HEMS must be connected to three main information pipelines from two external services:

- Forecasting pipeline composed of two sub-modules:
  - Weather forecasting pipeline, which use meteorological information in the household and external service to estimate the Wind Turbine (WT) and Photovoltaic (PV) production.
  - Load forecasting pipeline. Based on the historical information recorded by the Smart Meters as well as stochastic models, it estimates the behavior of the consumers in terms of demand for evaluation the most beneficial control actions to be taken.
- Electricity price pipeline, where the utility provides real-time prices for the energy that are included in the optimization process.

It will provide the current information regarding consumption and load state when requesting the HEMS an optimum scheduling. Therefore, after the computation the HEMS API will provide a response with the control action that will be transmitted to the gateway and finally be carried out in each device.

The aforementioned HEMS architecture implemented in the smart home is presented in Fig. 17.

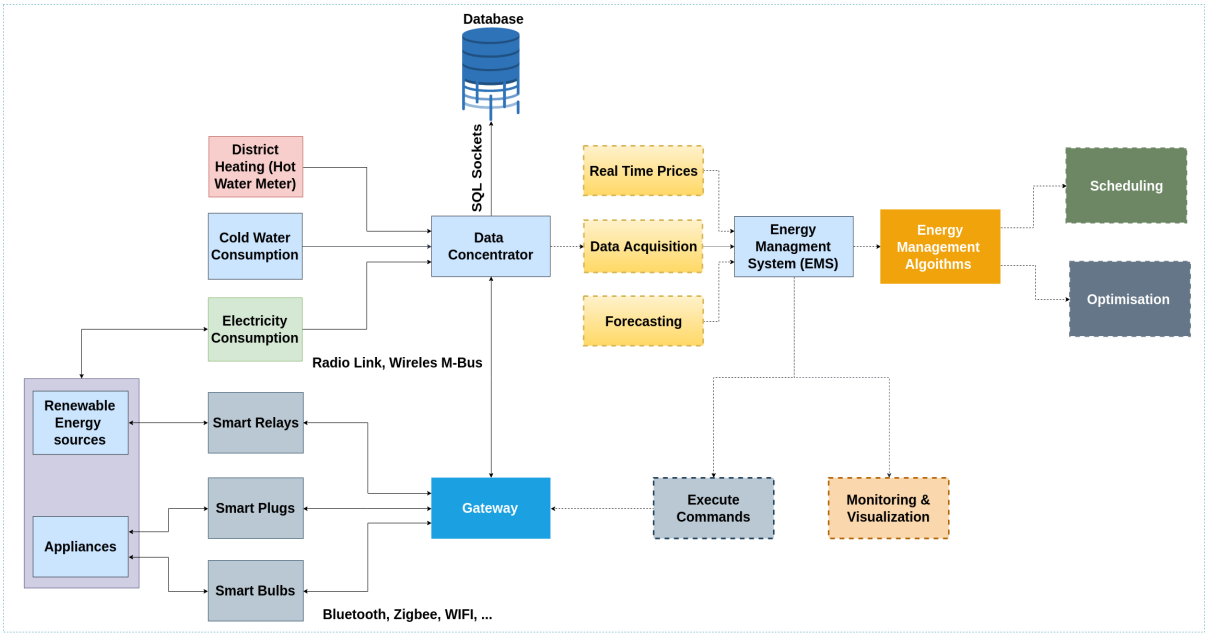


Figure 4.17 – Architecture of the HEMS

4.8 Conclusion

This chapter presented an architecture of a next generation smart home, which is considered as a smart space. It encompasses different types of services in order to provide a better quality of life for the occupants. In a first step, these services are narrowed down to only the healthcare and the energy management ones, future works may extend the architecture to support more services. These services are then implemented and tested. Moreover, through a user friendly interface, the user can have a clear visualization of the different data (i.e. health and energy) collected from all over the smart home, together with an alert system in case of detection of an abnormality (i.e. a fall detection). One of the main objectives of such architecture is to link the smart home to the external service providers, such as the energy utility, in order to first improve these services (i.e. to have a cost efficient energy consumption), and to secondly take a step toward the vision of the smart city. In the same vision, since in real life, several smart space exists (i.e. several smart homes), the next chapter provides a solution to manage them in an efficient and centralized way.

Chapter 5 | Access to multiple smart spaces using SDN

Contents

This chapter proposes the third contribution, and is structured as follows. It starts with a global and general presentation of the main challenges tackled in this chapter. Then, provides the related works of the architectures that propose a coupling between the WoT/IoT and the Software Defined Networking (SDN). Next, the principles of the proposed architecture are presented. The implemented solution is described with an

explanation of the different components and the different interactions, followed by the technical details regarding the implementation. This architecture is then illustrated in the e-Health domain. Furthermore, the access delegation challenge is discussed. Lastly, a FCAPS (Fault, Configuration, Accounting, Performance and Security) analysis is provided together with a discussion of the architecture related issues, before concluding the chapter.

5.1 Introduction

In the previous contributions, the different challenges to access a single smart space, in the case of a smart home for instance, were discussed. It results that several requirements needs to be guaranteed in order to have a secure access to these smart spaces. To mention: the authentication, the confidentiality and integrity, and the access control. This smart space is then used by the different services, such as the telemedicine ones.

On the other side, the new vision of the smart city aims at delivering multiple services to their citizens, in order to improve the quality of their life. It systematically implies the presence of multiple smart spaces all over the city, and in different networks. Thus, the main challenge, in this case, is how to manage all of these smart spaces in an efficient way. Particularly, how to guarantee the previously mentioned security requirements for all the smart spaces.

Consequently, this contribution aims at discussing this challenge and then presents our proposed contribution to solve it.

5.2 Challenges and issues

Lets first start by defining the main issue to solve. The main objective is to be able to secure the access to all the SOs belonging to a given entity (which can be a user, a service provider, an infrastructure, and so on and so forth). Currently, most of the access control solutions are focusing on controlling a single SS using a local or remote access control management. However, the objective of this contribution is to be able to manage the security of several SSs located in different LANs. For instance, a user may have a SS in the smart home, a second SS in the smart garage and another one in the smart office, as shown in Fig. 5.1.

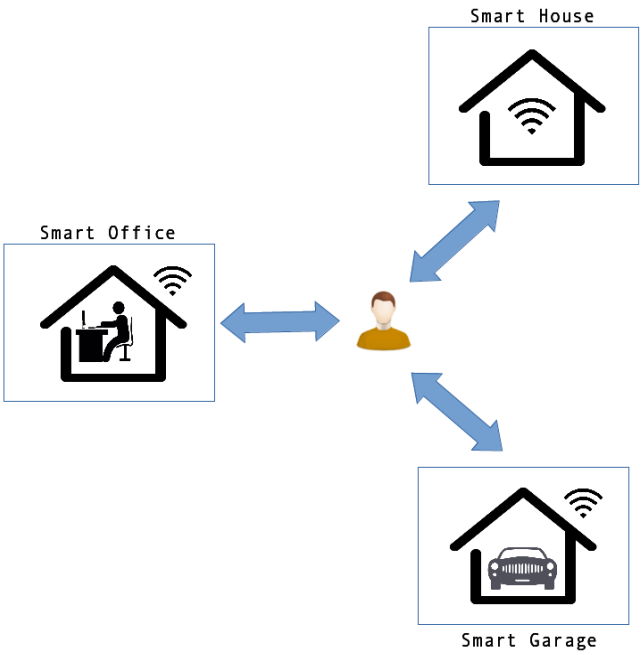


Figure 5.1 – Multiple Smart Spaces owned by the same entity

Knowing that all of these SSs belong to the same entity, and that their number may vary from one use case to another (for instance the rooms of a Hotel). Managing each SS separately is an arduous task for an administrator. Moreover, the administrator needs to be careful while updating the access policies in each SS, to guarantee their consistency, to avoid conflicts, and so on.

A solution is to manage the security of all of these smart spaces by using a centralized entity. This entity should be able to guarantee all the security requirements in order to allow a secure access to one of the smart spaces, while allowing the establishment of secure routes between the user and the requested smart space.

A trivial solution can be the use of a centralized server, which controls all the smart spaces. This server will have to manage the authentication of the requesting user, the management of the access control policies of all the smart spaces, and the creation of secure channels first between the server and smart spaces and between the user and the smart space.

The major drawbacks of such solution is that, first, in this vision, all the smart spaces need to be exposed to the Internet using public IP addresses, which adds a new attack surface for the malicious peers. Secondly, two secure channels needs to be established. One between the server and the smart space, and then one between the smart space and the user. Finally, the server cannot manage the routing and configure the network equipments, such as the firewalls, at the network layer between the user and the smart space.

Hence, to keep the advantage of having a centralized decision point together with a solution to manage the network related issues, such as the routing, and to avoid the

previously mentioned issues, we propose the use of an **Software Defined Networking (SDN) controller**.

SDN [124] has attracted great interests from both academia and industry. It changes the traditional way of managing the network, where routing decisions are taken by the network equipments themselves (the routers). However, several drawbacks are raised. Mainly, every single router is a quite complex machine that has to communicate with others through dedicated protocols. And any modification/reconfiguration of a network equipment must be done manually by the administrators. In [124], the concept of the SDN is explained, where mainly, they propose to separate the control plane (i.e. the place where routing decisions are taken) from the data plane (i.e. the physical data transfer). This concept emerged in order to respond to the important traffic engineering problems raised by the traditional networks.

Hence, by having a centralized decision point for access control directly linked to the controller, together with the authentication mechanism, a dynamic and efficient administration of the security of the different SSs can be achieved. Moreover, since the SDN controller knows how to route the different packets to the corresponding smart space, there is no need for the smart space to be directly exposed to the Internet.

Hence, with these third-party applications interacting with the northbound interface, the attacks surface can be reduced and the user's resources can be protected. The use of the SDN in our architecture is justified with the following arguments:

- To ensure granular security:
  - To be able to manage the security of all the network equipments, in particular the firewalls, in a central way. Indeed, using an SDN in simple case of managing a single SS does not show its real potential. However, the problem lies in the case of having several SSs. Managing the rules of each firewall, managing the NAT traversal, updating the access policies, updating the different databases, etc., while guaranteeing the consistency is a difficult challenge.
  - To encounter the (D)DoS attacks: SDN enables first the detection of DDoS traffic and to immediately install blocking rules to drop this malicious traffic, and hence, to mitigate the attack [125].
  - Centralized management of the security, where most of the security checks and countermeasures are done by the SDN controller such as the authentication, the access control, cryptographic key provisioning and the creating of secure channels. With a special focus on having a fine-grained access control, where a first control can be done in the SDN in order to filter the maximum of unauthorized users, and a second control is done in order to block the malicious intent of even the legitimate users in each SS. Additional details are provided later in this chapter.

  - Security event management (SEM): SDN can be interfaced with SEM [126] which will analyze and manage the network traffic for any abnormality or process that can affect security or availability. Eventually, by also interfacing an Intrusion Detection System (IDS) and an Intrusion Prevention System (IPS) with the northbound interface of the controller.

- To have a better QoS: being able to efficiently manage data traffic makes it easier to implement quality of services (QoS). For instance, in case of the voice over IP and multimedia transmission, more bandwidth can be allocated for this particular services using the controller. Hence, ensuring a better QoS. A practical example can be related to WebRTC, where, for instance, a client needs to use a camera located in one of the owned SSs in order to communicate in audio/video with another user. The SDN will facilitate the access to this SS and to this camera, and allocate more bandwidth to the multimedia flow in order to have better QoS.

- To facilitate the connectivity of the users, by creating direct and secure routes to the requested resource.

The Fig. 5.1 summarizes the main advantages of using an SDN in such architecture. The security at low level layers is mainly the firewall rules regarding the control of the traffic, and the security at the high level layers consist in the authentication, authorization, access control, confidentiality and integrity.

	Without SDN	With SDN
Confidentiality	Yes	Yes
Integrity	Yes	Yes
Authentication	Yes	Yes
Access control	Yes	Yes
Traffic control (3)	No	Yes
DoS management (2)	No	Yes
Applications control (1)(4)	No	Yes
Quality of Services	No	Yes
Dynamic management of network equipments	No	Yes
Connecting the client to the gateway	Hard (need to handle NAT traversals and firewalls)	Easy (all the control can be done with SDN)
Scalability	No	Yes (conditionally see discussion section 5.9)

Table 5.1 – Comparison between an architecture that does not use SDN and another that does

(1) Managing unknown traffic with strict rules (eventually blocking them by default).

(2) Identifying DoS attacks and preventing them by controlling the flows of data.

(3) Identifying and controlling the applications listening on any port, not only the standards ones.

(4) The possibility of controlling the flows and the security of the different applications interacting with the system in a simple way.

5.3 Theoretical architecture

Generally, using an SDN controller brings several advantages including:

Centralized network provisioning: in order to have a centralized view of the entire network, making it easier to centralize the management and the provisioning of the network equipments.

Interaction with the application layer: allowing applications to issue commands and requests to the controller, and to provide additional functionalities that cannot be done by the controller.

More granular security: having a central point of control over the overall topology, which might often change, can facilitate the management of the security policies. In addition to be able to have consistent security policies over all the topology. Indeed, centralizing the security control into a single SDN controller, has the disadvantage of creating a single point of failure. However, those problems can be solved by having a secure and proper implementation of SDN, and by having multiple instance of the controller.

Thus, in order to solve the previously mentioned issues, in particular the security ones, a WoT-SDN architecture is proposed. The idea is to have a centralized security control over the different SSs, and to allow users to access them anytime and anywhere. The security of the system starts from the low level security checks by filtering and controlling all the flows going through the firewall and by dynamically modifying the rules of the firewall depending on the needs. Then, by authenticating the users at the application level. In addition to be able to encrypt the whole traffic between the users and the requested resource in the SS. For this purpose, security keys need to be delivered to the communicating parties. And finally, to be able to manage the access control in a fine-grained an granular way.

Furthermore, the reliance on the northbound API of the SDN controller is expected to shape the large-scale deployment of SDN, where multiple "third-party" applications may interface with the northbound interface in order to provide more services, in particular regarding the security and the privacy management. Even though, relying on third party application may expose the SDN controller to new security threats. Several countermeasures can be done by the service provider (which can be an operator for instance) in order to control the interaction between the SDN controller and the third party application such as by authenticating applications and controlling the access to

the northbound interface.

The Fig. 5.2 gives a global view of the theoretical architecture. The architecture can be divided into two main parts: The entities which interact with the northbound interface of the controller. And the entities which interact with the southbound interface of the controller.

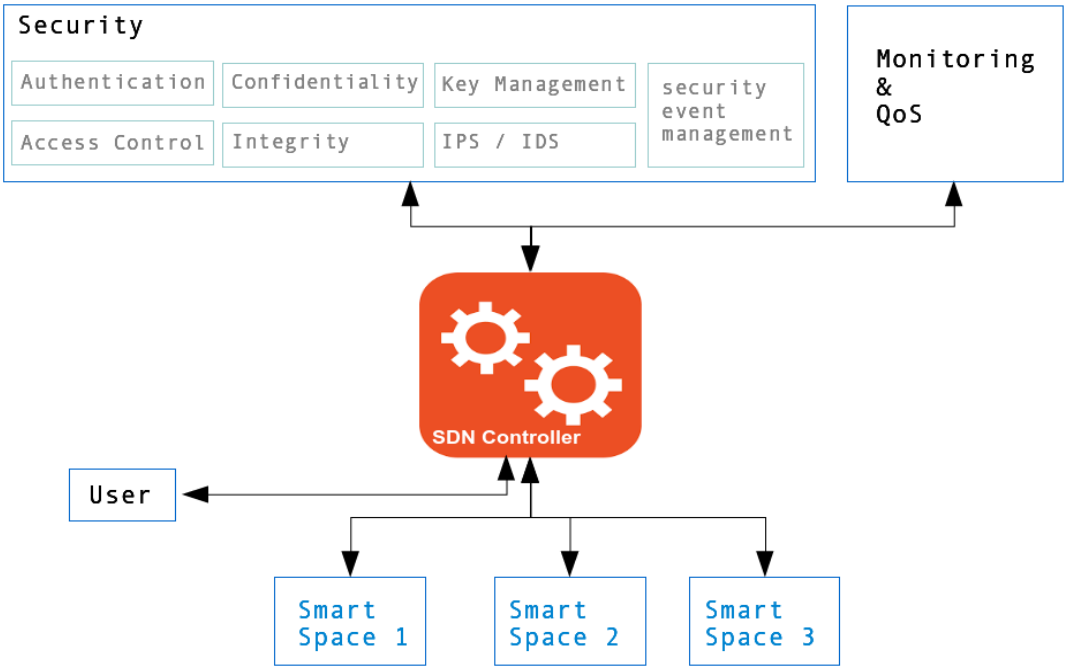


Figure 5.2 – Summary of the global architecture with the SDN controller

The first category represents the different mechanisms complementary to the core components of the SDN. They mainly add security, monitoring and QoS mechanisms to the system:

- The security mechanisms: composed of an authenticating third-party, and access control mechanisms, and a PKI to deliver security keys.
- The monitoring mechanisms: in order to monitor the traffic and eventually deploy additional security detection mechanisms, such as IDS or IPS etc.
- The QoS mechanisms: in order to manage the QoS of the different SSs. For instance to allocate more bandwidth for a certain SOs for a particular case.

And the second category represents the clients and the different SSs. The clients can be either outside the SS or inside, and they request the access to the resources of one or several SSs. The entry point to each SS is a gateway, which also exposes a RESTful API in order to interact with the different SOs.

A particular interest is given to the security mechanisms. The monitoring and the QoS mechanisms are out of the scope of this thesis, and can be considered as perspectives for future work, since they fit perfectly with the kind of services which can be provided by our architecture. The main security mechanisms that we propose are: 1)

the authentication: using either an external entity such as Identity Providers (SAML, OpenID Connect, etc.), or by using internal authentication. 2) Access control: which is the main discussion of this chapter. 3) A PKI (Public Key Infrastructure): in order to provision the different communication parties with the cryptographic keys in order to secure the exchanges. Guaranteeing, hence, the confidentiality and the integrity of their exchanges.

5.4 Related works

This section presents, first, the related works of the architectures coupling the SDN with the WoT/IoT in general, and for e-Health in particular, in order to manage several SSs, together with a comparison with our proposition. Then provides a quick background related to the use of SDN to manage the security of the network and IoT infrastructures. Finally, summarizes and compares the related works in a table.

5.4.1 SDN-IoT/WoT architectures

Several works have been proposed in the literature for the architectures combining the SDN control capabilities and the advantages brought by the WoT/IoT paradigms. The authors in [127] propose to consider the end-devices without routing or switching capabilities, but only transmission and reception of data, in addition to the traditional network elements such as the routers and the switches. They propose to delegate the control and the management of the network connecting many devices of a smart environment (SE) to a software entity, while letting the control of the network devices to the end-user. They introduce the notion of the SD-LAN (Software-Defined LAN) in order to create a virtual environment that can regroup several objects from different networks as if they were on the same LAN. However, it is still an ongoing work, and the security issues are not detailed in the paper. A second work is proposed by X. QUI and al. in [128]. They propose to build the architecture of resource oriented WoT based on SDN. They argue that the use of SDN provides new opportunity to improve performance, simplify management and enhance the WoT's security. They propose to solve the challenges related to the weak management features, the lack of efficient access mechanisms and the conflicts between the openness required by the WoT and the security required by the applications using the SDN. However, they lack the access control part. In [129], presents a theoretical architecture that combines the IoT with the networking capabilities of SDN in order to deal with the network security challenges. They introduce the notion of SDN Domain, where each domain is controlled by a single SDN controller. Each domain may contain either constrained nodes or non-constrained ones with enough resources. The constrained device/node can be associated with another neighbor, which has enough resources and SDN capabilities. They also deal with the case of several SDN domains, and their interactions. For this they introduced the notion of "Border Controller" (BC) located at the extremity of each SDN domain. This BC will be then responsible for the interaction with the other BC of the other domains. However, the proposition is only a theoretical architecture, they lack experimentation and results. And additionally, they do not deal with the access control to the resources

of the devices.

The authors in [130] propose to use an SDN gateway as a distributed means of monitoring the traffic to/from the IoT devices. This gateway is also able of detecting abnormal behaviors and attacks and to eventually counter them. However, the work focus only on the traffic analysis and attack detection and mitigation at the network edge, and do not deal with the access control to this gateway, neither deals with the multiple smart space issues. [131] presents a secure IoT architecture for smart cities. The architecture is mainly composed of a black network, a trusted SDN controller, a unified registry and a key management system. The purpose consists in securing the smart city's data, and in particular the mission-critical ones, in order to prevent cyber-attacks. The only drawback is the management of the access control to the different networks, which can be done using the SDN control. Another paper [132] describes a solution to protect the IoT devices of a smart home using a network-level security instead of the device-level one. The solution is based on the use of an SDN controller to implement dynamic security rules, such as the access control one, which can evolve depending on the context and the user preferences. Offering, hence, a security-as-a-service to residential consume. Additionally the solution proposes a module called Security Management Provider (SMP) as a third party entity specialized in the security. The solution provides the user with a possibility of delegating the protection of the home appliances to the SMP, which interacts with the SDN controller in order to provide a complementary configuration control over the ISP network of the smart home on behalf of the user. However, compared to our proposed solution, this one misses the protection of the flow between the user and the corresponding home gateway and the key management, and also do not explore the complexity of managing multiple smart home (or smart spaces).

5.4.2 IoT-SDN architectures for healthcare

Some works propose to use the IoT and SDN in order to provide better healthcare services to the patients. The authors in [133] propose a softwarized infrastructure for IoT systems for smart healthcare applications and services. They propose to use Tor mechanisms in order to preserve user's anonymity and to counter network surveillance threats, and to use the Blockchain technology in order to guarantee the security of patient's records by tracking and authorizing access to confidential medical records. However, they deal only with the security of the patient records and they do not give a security solution for the different IoT devices and sensors that they deploy in their infrastructures. [134] proposes a taxonomy and an architecture for the implementation of the software-to-data-paradigm in healthcare services. They propose to use Hybrid cloud in order to provide the main security services. However, these solutions may not be very compliant with the constrained environment of the IoT. In [135] the authors propose a solution called "SCORPIO", an SDN control plane, which provides efficient multipart multimedia data sessions for Smart Surveillance applications. However, they do not deal with the security issues. And in [136] the authors propose a secure WBAN architectural system for the next generation virtual hospital where SDN is used for efficient data management and content delivery system as well as using Kerberos for networking authentication protocol and for better security of the system was proposed.

However, they do not propose how to manage the security of the IoT devices, they have just secure the exchange between the patient and the medical service. All of the previous solutions deal only with a single SS. One proposition to secure the network access and network resources, is by authenticating first the network devices by the SDN Controller. To begin, OpenFlow secures the connection between the switch and the controller, the controller then blocks switch ports directly linked to the users. Then, only user's authentication traffic will be allowed. Upon a successful authentication, and based on the authorization level of the user, the controller will push the appropriate flow entries to the switch. They also extend this to authenticate IoT devices. Moreover, they do not explain how to use the Grid of Security middleware in this architecture. And more importantly they do not deal with the access control to the resource of the devices.

5.4.3 Network security over SDN

Handling the security of the network and the IoT infrastructures using SDN is a new domain that start to gain popularity. In this section we will provide a view on the most prominent related works in this field. Most of the previous works on SDN focused on the security, and particularly the access control of the network at the flow level, for instance regarding the deployment of firewalls and the enhancement of its functionalities such in [137, 138, 139], and by controlling the traffic that goes in and out of the network equipments.

The authors in [140], provided an architecture for controlling the network access for M2M communication using SDN. They demonstrate how SDN can be used to develop basic network access control services without 802.1X (which requires hardware and software equipments, and which is not given in the constrained networks such as IoT), while offering the ability to adjust the available bandwidth and network access policy per device. The control is based on the flows between the client (which can be just a simple user or an IoT device) and the network equipment, for each connected and authenticated client, it retrieves and then installs the corresponding access rules and bandwidth capacity on the SDN controller for that user's policy. Hence, only the packets belonging to the authenticated user are allowed (based on checking certain fields). The drawbacks of such vision is that, first, they deal only with the wired part of the network, and they do not deal with the different communication constrained of the IoT/WoT networks. Secondly, it is possible to compromise the user's identity since the packets are identified by simple fields which are the Ethernet source and destination address and the IPv4 source and the destination address, the Ethernet In-port and the Band Rate. Moreover, there is no control over what the client can do on the IoT devices, since there is no policy model applied in the architecture. Only the access level (access to subnets and hosts) and the bandwidth rate for each user are defined.

Another work for controlling access to constrained resources problem is proposed in [141]. They argue that most of the current solutions cannot be applied to this particular case since it generally requires high computational and bandwidth capabilities and offers poor interoperability against standardized communication protocols for the IoT.

Hence, they propose a flexible authentication and authorization framework for the IoT using OAuth, and they call it OAuth-IoT. Their architecture is based on two entities. The authorization server which gets the authorization from the owner of the resource once a client is connected and authenticated, and then issue a token to the client. The gateway to the IoT network of nodes verifies the token of the requesting client and then lookup of the corresponding resource in the IoT network. However, they also do not use an access control model to control what a user can do on which IoT resource. In addition, this framework is intended to authenticate and authorize users within the same SS (in a single network) without using SDN. Hence, for a more complex network, they may have management and access control issues.

5.4.4 Summary of the related works

Ref	Multiple SS	IoT	Confidentiality and integrity	Authentication	Access control
[130]	No	Yes	Yes	No	No
[132]	No	Yes	No	Yes	Yes
[133]	No	Yes	No	Yes	No
[134]	No	No	No	No	No
[135]	No	Yes	No	No	No
[136]	No	Yes	Yes	Yes	No
[129]	Yes	Yes	No	Yes	No
[127]	Yes	Yes	No	No	Yes
[131]	Yes	Yes	Yes	Yes	No
[128]	Yes	Yes	No	Yes	Yes
Our	Yes	Yes	Yes	Yes	Yes

Table 5.2 – A summary of the state of the art

To summarize, the previous works do not provide a solution for controlling the access to the different IoT devices. Moreover, most of the current solutions focus on the management of a single SS, which can be less suitable for a multitude of SSs. Hence, our main contribution is to provide a granular access control for the different SSs, together with all the main security properties such as the confidentiality, the integrity and the authentication.

5.5 Proposition

In this section we propose a new and innovative design of a security management system of a Web of Thing architecture based on SDN. The proposed architecture intends to deal with the previously discussed issues and challenges in section 5.2. Hence, to manage the different security aspects such as:

- The Confidentiality and the Integrity of the data: where all the exchanges need to be encrypted. The communication between the controller and the network equipments and the communication between the user and the smart devices should be

also taken into account. For this last one, using dynamic session keys is interesting.

- The Authentication: in order to verify the identity of the users, which can be achieved by using a simple authentication, or by using external identity providers. They may issue tokens or "Identity Assertions" respectively. In our implementation, the JSON Web Tokens (JWT) are used.
- The Access Control: which is the main contribution of this architecture, allows the control of the access to the different SSs of the user in a centralized way.

Hence, the proposed architecture is shown in Fig. 5.3. It mainly describes, the process of accessing a SO's resources in a particular SS (the smart space 3 in this case), from the smartphone/laptop/tablet of a user. Hence, this figure provides the overall view of the different interactions between the different components in order to allow the 'User' to access a resource in a secure manner. And generally, to control the access to the resources so that only the authorized ones are allowed. We will start first by explaining the meaning of the different arrows. The "blue arrows" shows that the user can be anywhere connected to the Internet, within the range of the ISP. Eventually communicating with the server using a secure protocol such as WebSocket/HTTPS. The "orange arrows" for the different interactions in order to perform the security checks. The "red arrows" correspond to the OpenFlow data between the SDN controller and the different OpenFlow switches (OFS). And finally the "black arrows" correspond to the physical connection between the OFSs.

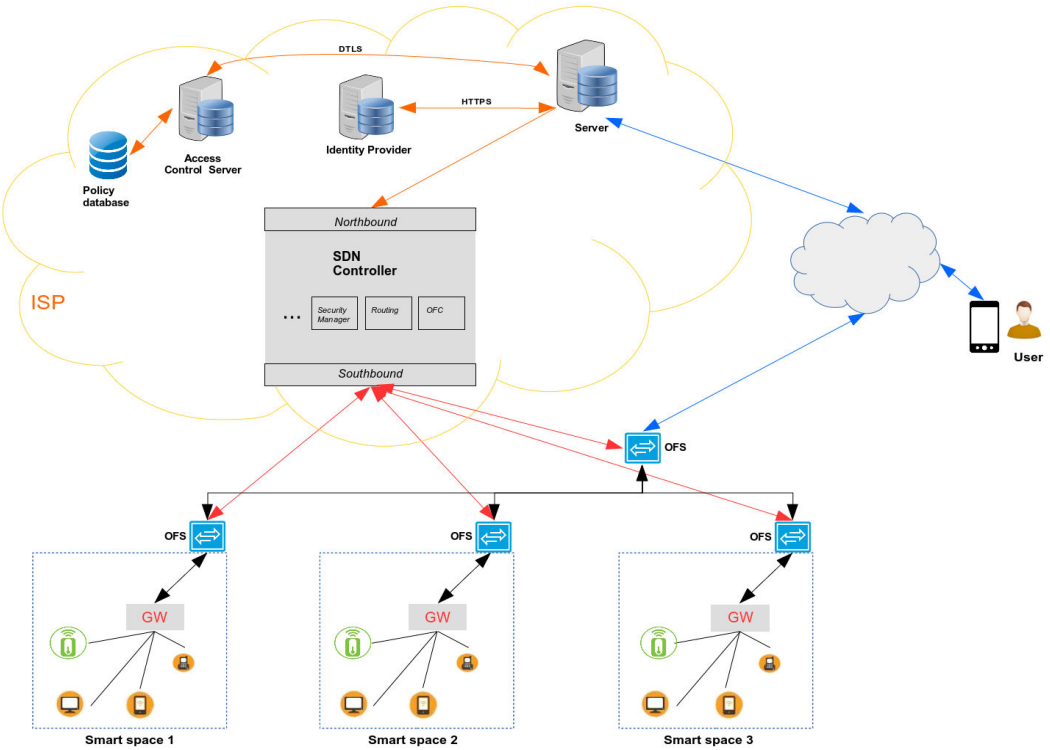


Figure 5.3 – Global view of the architecture

Now, a detailed explanation of the different components of this architecture is presented:

- User: is the one owning a set of SO in a SS, and eventually in several SSs, and the one requesting the access to a resource. The user can access the platform trough a web application provided by the server using simply a browser. The user communicates with the server using a secured communication protocol in order to guarantee the confidentiality and the integrity of the data, such as Websocket/HTTPS.

- Server: is the second main component of this architecture, since, it provides the main security mechanisms in one side, and communicates with the SDN controller on the other side. Furthermore, it contains the application logic, facilitates the authentication of the user and controls the access for each user's request. The Server in this case has a direct interaction with the Identity provider in order to authenticate the user. A proxy of the IdP can be implemented on the server side. Moreover, the communications are secured using HTTPS. Once the authentication done, the server will then interact with the Access Control Server (ACS) in order to get the access authorization. The server also establishes a connection with the SDN controller in order to exchange the different information, which are mainly the network information related to the user and to the requested resource, and to issue commands.

- Identity provider (IdP): is responsible of creating, maintaining and managing the identity informations of the different users. It performs two main tasks: 1) authenticating the user and providing an identity assertion. 2) Asserting the validity of an identity assertion of a user by whom concerned. In our implementation a simple authentication server is implemented, and which issues JWT tokens with the identity of the users.

- The Access Control Server (ACS) and the policy database: its main goal is to get the access request and to decide whether the user is authorized to access the resource or not. This component also controls the operations that a user is authorized to perform on a given resource. The set of access policies is located in a Policy database. The PEP/PDP framework is adopted in order to control the access, as explained in section 3.5.3.2. Mainly the server gets the access request of the user, then relays it to the PDP (which is the ACS in our case), which decides if the user is either authorized or not. The RBAC approach is used in order to restrict the access only to the authorized users, as explained in section 3.5.3.1. In this case, the access control is based on the role of each user in the system. Each role have a set of pre-defined rules and operations allowed to be performed. The rules are stored in the policy database, and a PDP hosts the RBAC approach in order to take the decisions.

- Public Key Infrastructure (PKI): is used to issue the necessary key in order to secure the communications between the different components of the system.

- OpenFlow Switch (OFS): which can be either a software program (virtual switch) or a hardware device (real switch compatible with OpenFlow) which is connected to an SDN controller and supporting the OpenFlow protocol for exchanging the network packets.

- SDN controller: is the main components of the system. Its main role is to configure the different network equipments related to the different SS in such a way that only the legitimate users are able to access the resource. The northbound interface of the SDN controller which enables the interaction with the third party applications and the southbound interface interacts with the different network equipments such Open virtual Switches (OvS), or OpenFlow Switch (OFS), in our case, using OpenFlow protocol.

  The SDN controller may contain other core components such as a Flow manager to control the destination of each packet, a Security Manager (for additional security needs such as the creation and the sharing of cryptographic keys, etc.), a Hosttracker (in order to get information about hosts and switches), etc. The architecture uses openSDNCore controller [2] which is developed by Focus Fraunhofer. Some of the core components of the SDN Controller are explained in details later in Fig. 5.7.

- Smart Space and the gateway: contains mainly the different resources provided by SOs. They can be either non-constrained ones with enough capabilities to interact with the user requesting the resource, or constrained ones, and in this latter case a WoT gateway is used. The gateway offers a RESTful API and communicates with the smart devices using different communication protocols (Bluetooth, Zigbee, WIFI, etc.). The architecture of this gateway is presented in Fig. 5.4:

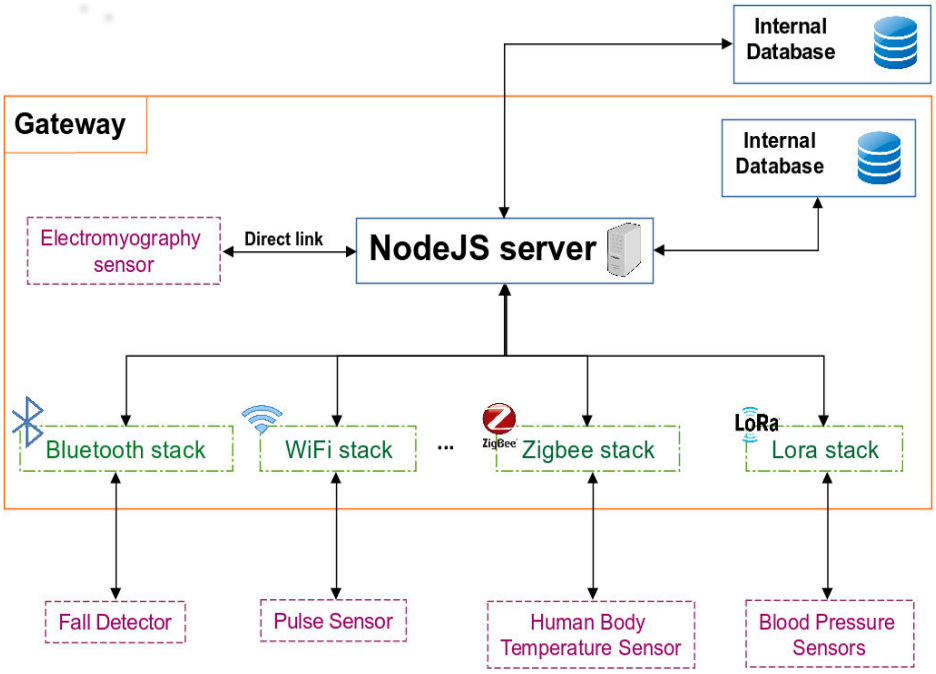


Figure 5.4 – WoT gateway

Hence, using this architecture, several SSs belonging to the same entity can be managed using SDN as a central controller. The Fig. 5.5 presents a simple diagram summarizing the main interactions between the different components in order to process a user request to access a particular resource:

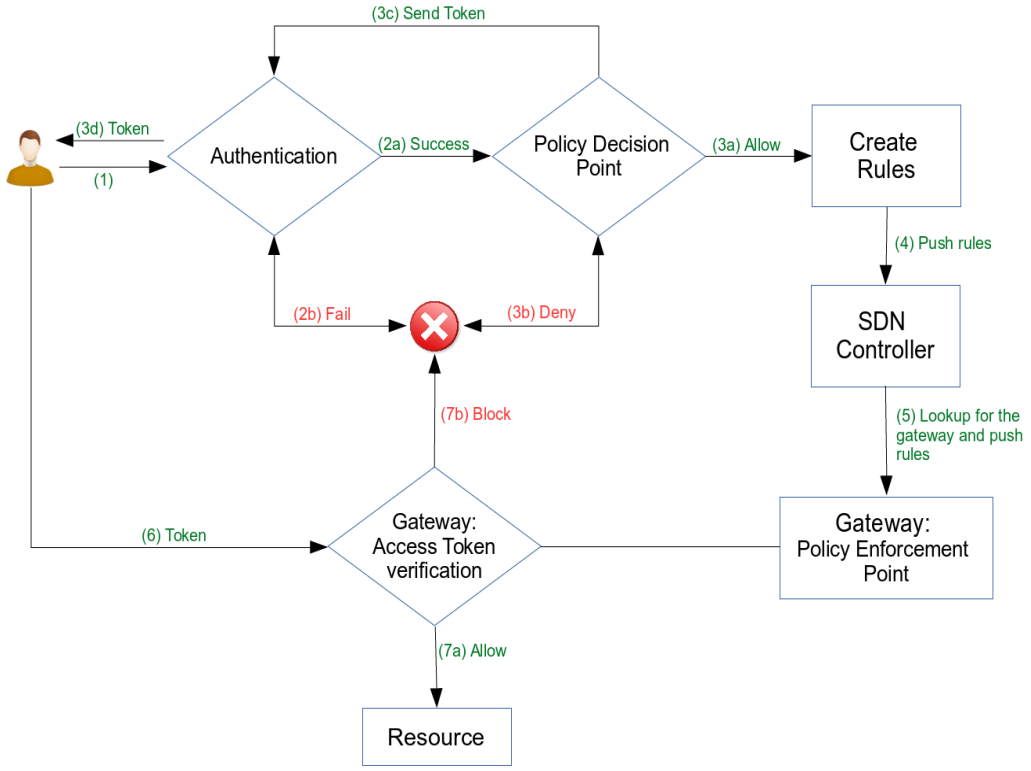


Figure 5.5 – Simple diagram of the resource access process

5.6 Technical details

Before starting the technical part, several assumptions were considered, which are:

- The SSs are already configured, where each SS has a gateway with a fixed private IP address, and connected to the set of SOs. In the PoC, three SSs were implemented, however, the assumption concerns mainly the large scale deployment of the architecture with more SSs.
- The users and their corresponding SSs are also configured in advance. An administrator can configure those profiles and SSs using the UI of the application.
- Symmetric keys are pre-shared between the server and the gateways.
- The main components are located in a secure network such as in the ISP, which can be the one providing the proposal to its clients (such as private users, private companies, smart infrastructures, and so on).

The SDN controller used in our PoC, as mentioned before, is called ”OpenSDNCore“ provided by Fokus Fraunhofer institute, which is currently closed source. However they mention that they plan to make it open source in the future.

5.6.1 Mininet network

The following figure represents the Mininet topology simulating the different networking part of our architecture. It contains mainly five OFSs. Three of them are directly

linked to a physical gateways (a raspberry pi), through one of the OFS ports, hence, representing the three SSs. One is directly linked to the client. And one creates a separation for the network, which can be seen as a router. All the OFSs are linked to the OpenSDNCore controller, and communicate using the OpenFlow protocol. The OFS can be also configured in order to create VLANs corresponding to each SS.

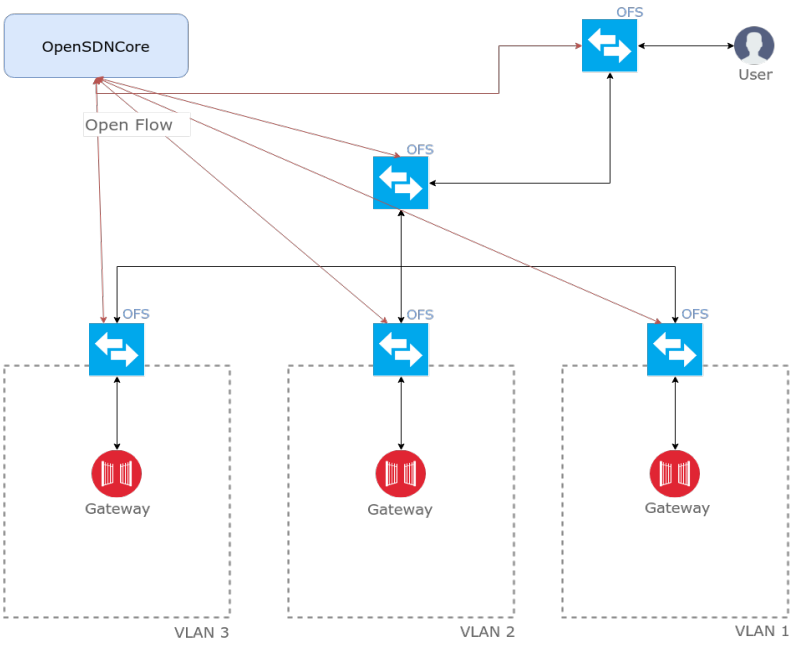


Figure 5.6 – Construction of the topology of network using Mininet

5.6.2 Token issuing and verification

The PoC uses the JWT (JSON Web Token) format [142]. It is a compact and secure way of representing claims to be exchanged between two parties. The claim is encoded using JSON (JavaScript Object Notation) format. The resulted JSON object, can then be used as the payload of a JWS (JSON Web Signature) structure or as the plaintext of a JWE (JSON Web Encryption) structure, enabling the claims to be digitally signed or hashed and/or encrypted, depending on the selected algorithm. For encoding (signing or encrypting) the claim, the secret key, the payload (or the claim) and the algorithm (by default is MAC with SHA-256), are the required parameters. Once the JWT is created, a callback is called. In the same way, for decoding (decrypting or verifying) the same secret key used for encoding is needed plus the token. A callback is then called with the decoded JWT. The Nodejs library is used to create the tokens and it is called ”jsonwebtoken” [142].

The ”aes-256-ctr” algorithm is used in order to protect the token. The choice of the CTR mode, which is a deterministic mode, is due to the variable contents of the payload, since the JWT tokens are unique. Additional parameters can be used in order for instance for the expiry of the token. In the PoC, its up to the authenticating server to generate such token. Upon a successful authentication, the server creates the JWT using the user identity from the authentication phase, and a secret key know only by the server. The token is used in two cases, the first one is by the access control server

in order to verify the identity of the user in order to provide the right authorization decision. And by the gateway when the users directly access it. On-demand, the authenticating server is also the one which verifies the validity of the token.

For the encoding of the token, the JWT standard [142], proposes two ways. Either by using symmetric keys. In this case, the secret for the HMAC algorithm (HS256 by default) needs to be provided. Or by using asymmetric keys, more precisely the digital signature, using the PEM encoded private key (for RSA or ECDSA), as shown in the signatures of the following functions:

- Encrypting the token using symmetric key :

$$S_T = Encrypt(Payload, SecretKey, Algo)$$

- Verifying the token (symmetric key) :

$$V_T = Decrypt(S_T, SecreteKey)$$

- Digital signature of the token using asymmetric key :

$$S_T = Sign(Payload, ServerPrivateKey, Algo)$$

- Verifying the token (asymmetric key) :

$$V_T = VerifySignature(S_T, ServerPublicKey)$$

5.6.3 OpenSDNCore

OpenSDNCore is an OpenFlow-based controller creating an intermediate between the network equipments and the application layer and manages the overall network. The controller is based on a three-tier architecture shown in Fig. 5.7. The three key components of the controller are: (1) the core function layer, (2) northbound interface and (3) southbound interface.

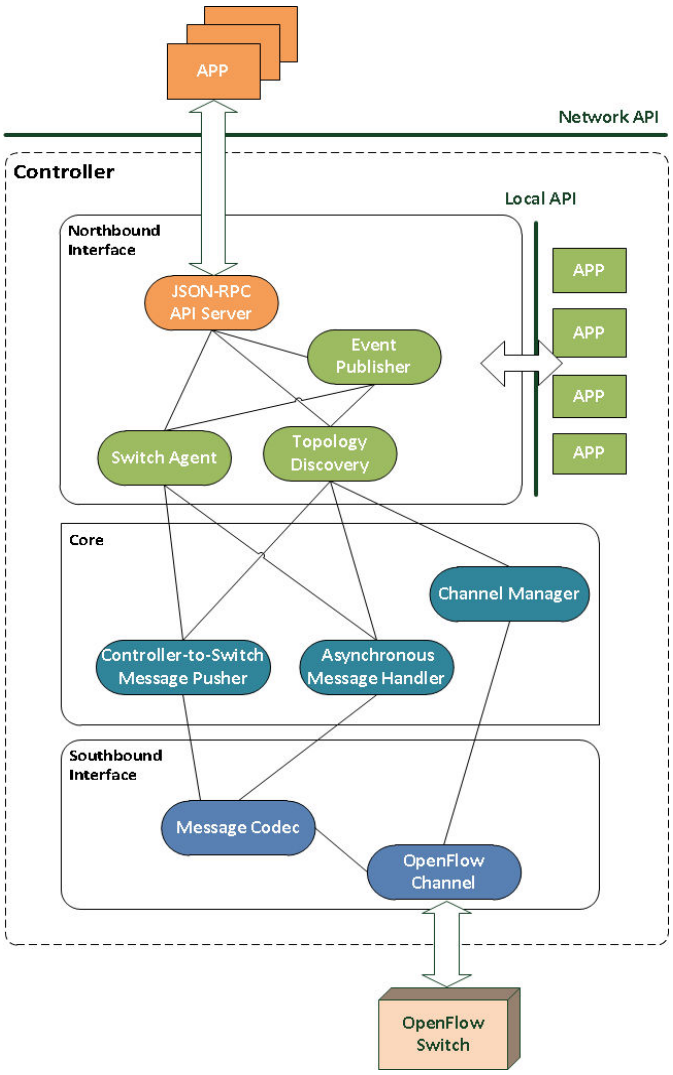


Figure 5.7 – OpenSDNCore's key components [2]

The southbound interface handles the communication with OFSs using OpenFlow Protocol. It consists of two components: the OpenFlow channel and OpenFlow message codec. The OpenFlow channel connects each switch to the controller. The OpenFlow message codec is used for encoding/decoding each time a message is sent or received.

The core function layer contains a minimized set of functions required for the deployment of the controller. The controller-to-Switch Message Pusher constructs new messages and push them to the OpenFlow channel. For asynchronous and symmetric messages coming from switch, the corresponding message handler will be invoked, which pre-processes the message and dispatches it to functions in northbound interface or other user defined applications outside the core space.

The northbound interface is the key enabler of network programmability. The presented open design enables network developers to develop applications by using two different kinds of API exposed by the controller. The key component is the JSON-RPC server, which is a stateless, light-weight remote procedure call (RPC) protocol. It intercepts the requests coming from the different applications and execute the corresponding commands. It uses an extended version of JSON as data format.

With the local API, applications are hosted in the same machine as the controller, and use the same programming language. However, it is not always the case to keep the application and controller in the same environment. Thus, as an alternative solution, the network API provides the possibility of remote communication. The controller runs an API server to handle incoming requests from other network applications.

In order to implement our approach, three main additional components were developed and added to the JSON RPC API: (1) "ARP Discovery" which allows the controller to find the location of gateways based only on its IP address, (2) "Create Route" which allows the user to communicate with the selected gateway, and finally (3) "Send Packet to Gateway" which allows the server to send any kind of data to the selected gateway. In the next sections we will go through each one of them in details.

5.6.4 ARP Discovery

The controller needs to have a global view over the overall topology. When the controller is launched in the first time, it does not know the network topology of the LANs involving the different gateways representing the different SSs. The simplest way was to ping them manually. However, to automatize the procedure, a method that send ARP requests to the different gateways was implemented. Hence, for each gateway, the server sent the corresponding command and as parameters, the "source IP", which is a random IP from the same network as the gateway, and "destination IP" which is the IP of the gateway.

The controller, then, forges an ARP packet based on the provided information, this packet will be encapsulated inside an IP packet and then again inside an Ethernet packet. This packet will be then transported using an OpenFlow message called "Packet Out" and will be sent to all the switches.

5.6.5 End-to-End routing module

As a security measurement, and before the creation of a secure channel between the user and the gateway, only the ICMP and the ARP protocols are allowed inside the topology, in order to prevent malicious users from accessing the gateways, and to minimize the attack surface targeting the architecture. For this purpose, a module called "End-to-End routing" was developed in order to allow the legitimate user to access the gateways, by enabling them to use additional protocols such as TCP and UDP. This method adds a route between two endpoints to the routing table of the learning switches. The advantage of this module is that only the source and destination IP addresses are needed to create the route. The method works as follow:

- From the server, a command with the two IP addresses of the user and the requested SS's gateway in order to create a route is issued. Both IP addresses are known since the IP address of the gateway is a static one (a DHCP can be implemented in future work), and the IP address of the user is known by the server, which can be retrieved from the HTTPS request.

- The JSON RPC server located in the SDN controller, and which represents the northbound interface of the controller, receives the request and checks the format of the request.
- From the parameters, it receives a JSON object containing the source and destination addresses, which can be either IPv4, IPv6 or MAC addresses.
- The main function is called "create_route":
    1. Takes as argument: source address, destination address, and filters if they exist.
    2. Creates a memory pool for the module
    3. Then calls the "hosttracker", which lookups for the host, with the IP address passed in the argument, in all the topology. The hosttracker returns all the information related to the Host such as the IP address, MAC address, the port where the host is connected on and the OpenFlow Datapath ID (DPID) of the switch directly linked to.
    4. The hosttracker is called twice in this case, for the source host and for the destination host.
    5. With those information, the topology can be parsed in order to search for the shortest path between the DPID of the switch directly linked to the hosts (i.e. the gateway and the user). The algorithm used here to search for the shortest path is Dijkstra [143].
    6. Again searching for the shortest path is called twice, first to lookup for the shortest path from the source to the destination. And to search for the shortest path from the destination to the source, since it may change.
    7. The two routes with the corresponding IPs are then pushed from the controller to the switches as a new entry in the routing table, using the "controller to switch message pusher".

5.6.6 Server to the gateway function

It allows mainly the server to send different types of data to the corresponding gateway. In our case, the data represents the access rules. For this method, only the IP address of the gateway and the listening port are needed. The SDN controller will then lookup for this gateway in the topology using again the hosttracker, which also allows to find the switch directly linked to the gateway. The method works as follow:

- The function is called "Send_package_to_gateway", and takes as parameters a JSON object containing:
    - ip: IP address of the gateway
    - type: UDP or TCP.
    - port: on which the gateway is listening

- payload: the data to send to the gateway (String/base64/json/etc.). The payload is encrypted using a pre-shared symmetric key between the gateways and the server, with "AES-256-CTR“ as cipher suite.

- This module works as follow:

  1. From the server, a command is sent with the IP address of the gateway, the port, the type of protocol to use to communicate with the gateway and the payload which contains the rules to be enforced inside the gateway.
  2. The JSON RPC server, located in the SDN controller, receives the request and checks the format. Then, extracts the parameters from the JSON object.
  3. The hosttracker is then called to collect the parameters related to the gateway.
  4. The creation of the Ethernet header.
  5. The creation of the IP header.
  6. The creation of the UDP header.
  7. The creation of the whole packet which is composed of a UDP Header with the payload (the access rules), encapsulated inside an IP packet, again encapsulated inside an Ethernet packet.
  8. This packet is then transported using the "packet_out" message.
  9. Finally this packet is sent to the corresponding switch (using the 'DPID' already retrieved using the hosttracket), then delivered to the gateway.

5.6.7 Final call flow

Using these modules, the Fig. 5.8 presents the final call flow of the different interactions between the different components in order to allow a user to securely access a resource. Some of the exchanges between user and server are omitted in the figure but detailed in the description below.

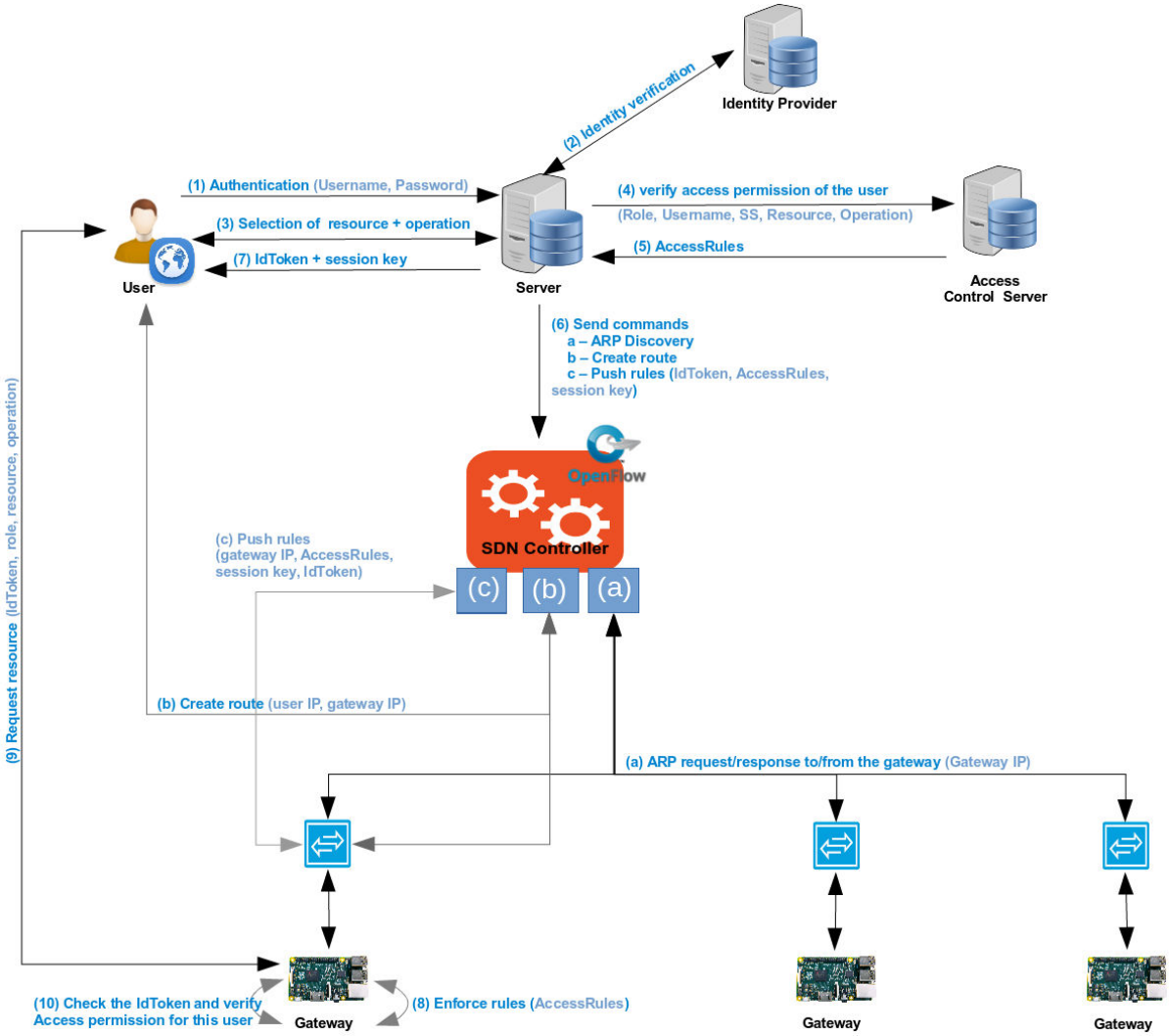


Figure 5.8 – Smart management of distributed smart spaces using SDN

1. The ”User“ connects to the ”Server“ and provides his/her credentials (username, password) for the authentication.

2. The credentials are then sent to the identity provider for checks.

3. Upon a successful authentication, the user can see a list of SSs, corresponding to the ones linked to his profile. The user selects a SS from the list. Consequently, the set of resources available in this SS will be displayed. The user can then select one of the resources with an operation, which will trigger an access control check.

4. The access control checks if the user is authorized to access this resource or not. This is done by contacting the access control server. Since an RBAC model is used, the server maintains a correspondence between the ”Username“ of the user and the corresponding ”Role“ of the user. The server then sends the user identity, the Role, the SS name, the resource name and the requested operation to the ACS in order to verify if such right exists. An optional verification of the identity can be also done here.

5. If the Role of the user exists in the ACS database, then it sends the different operations and actions that this role can perform to the server in ”AccessRules”.

It may also contain a timestamp in order to control the period of the access for instance. This is the first level of the access control process, where only the legitimate users are allowed to access the SSs, and also to avoid any interactions between the SS and the malicious peers. A symmetric session key is also created by the server.

6. The server sends then three commands to the SDN controller, as explained previously in this chapter:

   - (a) Sends an ARP request to every gateways, in order to learn the location of each one in the network topology. As a parameters we have the IP address of the gateway. This step in done only once, when the controller is first started.
   - (b) Creates a route between the user and the gateway so that they can communicate using TCP or UDP. As parameters the IP address of the user in one side and the IP address of the gateway on the other side are provided.
   - (c) The identity token, the access rules and the session key to the corresponding gateway together with the IP address and the port of the gateway are provided as parameters .

   The SDN controller executes those commands in the given order.

7. The user gets a "IdToken" with the session key which will be used to encrypt the communication with the gateway later.

8. The gateway then enforces the received AccessRules and updates its access policies database.

9. The user can then securely access the gateway, through an encrypted channel using the session key since the routing was established. As parameters the user provides the "IdToken", its role, and the operation to perform on the resource. The SDN in this case is responsible of distributing the security parameters (i.e. the security algorithms, keys, protocol mode, key management method, etc.) to the two endpoints.

10. Finally the gateway performs a final check of the Identity of the user and checks if he/she is authorized or not. If successful, the user can then access the SO's resources. This second check is done in order to avoid malicious actions even among the legitimate users. Hence, to control what operations the legitimate users are authorized to perform, and on which SOs inside the SS. Since, once the user is directly linked to the gateway, there is a risk of the user performing an operation other than the ones authorized by the controller. For this reason, by pushing the configuration to the gateway, first, the centralized control is guaranteed, since all the configurations are located in the ACS, and secondly, the user performs only the authorized operations related to his/her Role. Hence, in this case, the gateway is considered as a second PDP and in the same time as a PEP. It uses the received rules as a database in order to take decisions, and then enforces it.

5.7 E-health use case

As mentioned in the introduction, the next generation infrastructures, and in particular the health related ones, will introduce a huge number of IoT devices. Our previously explained approach is, thus, illustrated with a use case from the Health domain. It involves the management of the security of different rooms of a hospital, and where each room can be considered as a SS. The Fig. 5.9, represents the key parts of this system. The proposed architecture enables users to access the health resources of the hospital's (H) rooms, in a secure manner. Details about the different parts of the system can be found in the sections 5.3 and 5.5.

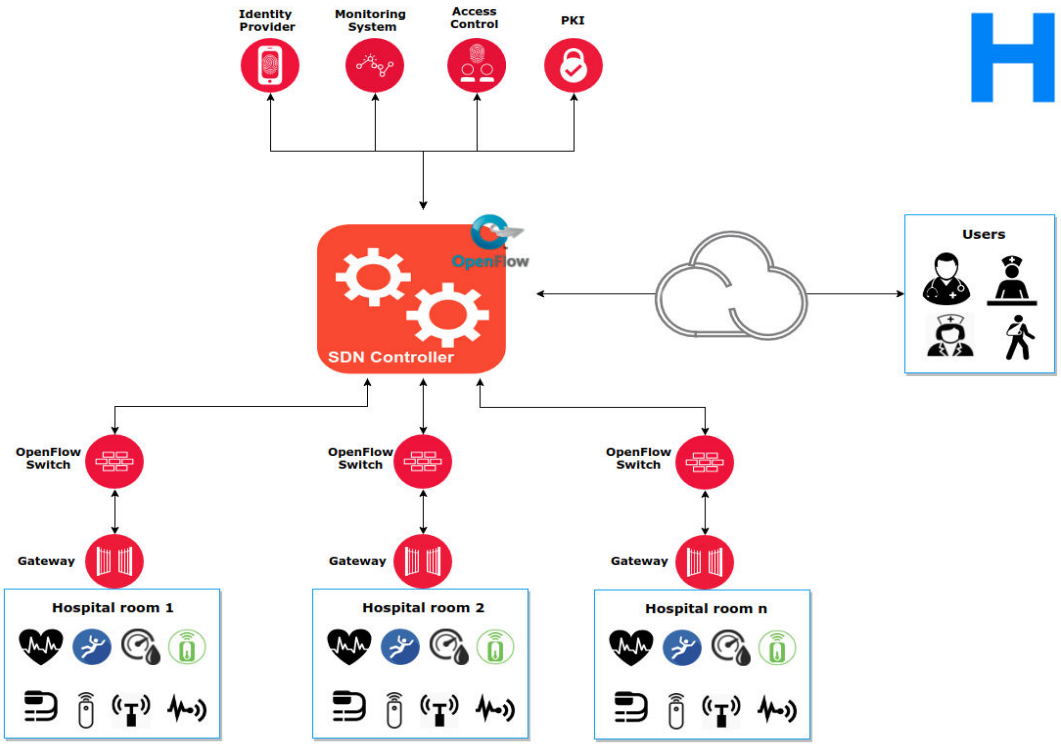


Figure 5.9 – Management of multiple SSs in a hospital

5.8 FCAPS analysis

In order to test and evaluate our architecture, the standard framework FCAPS (Fault, Configuration, Accounting, Performance and Security) is used. In next generation networks and infrastructures, and in the new vision of smart cities, managing all these properties becomes a must, in particular for the critical infrastructures such as the health related ones. In this analysis, both fault and accounting analysis are out of the scope of this thesis, and considered for future work. However, they can be both managed by the ISP (Internet Service Provider), since one of the hypothesis is that the architecture is located in its infrastructure. In the previous sections, a detailed explanation of the configuration part is provided, both for network configuration and security configuration in case of centralized access control management by the SDN. In this section, a performance and a security analysis is provided for the different components of the architecture. The analysis focuses on two parts, the SDN controller and the Server:

- The performance of the openSDNcore controller, in particular when the number of users and the number of commands sent to increases exponentially.
- The third parties applications interacting with the northbound interface of the controller, which is the front-end server, or the Server, by increasing the number of users and testing the load on the server and its impact on the host.

Using Locust [144] and Dstat tools, we evaluated:

- The number of simulated users that can continuously send requests and commands to the system. The requests are sent randomly in a distributive way.
- The average response time in milliseconds.
- For the SDN part: the load on the host in term of CPU usage, memory usage, number of running process, and network I/O send and received both the total and for each switch simulated using Mininet.
- For the servers: the load on the host in term of CPU usage, memory usage, the number of running process and finally the response time for the RTT of a request to access a resource in a given smart space.

Moreover, a security analysis is conducted on the controller and Server, by launching a set of attacks using OWASP's Zed Attack Proxy (ZAP)[145] penetration testing tool.

The performance part of both components allows to size the capabilities of this architecture by determining its limits relatively to the processing load due to the simultaneous accesses. The security part concerning the two components makes it possible to evaluate their robustness by determining some of the vulnerabilities that could be exploited by an extensive series of up to date attacks and eventually, as next steps, to apply patches.

5.8.1 Analysis of the SDN controller

5.8.1.1 Performance evaluation

In this section, a set of tests regarding the load that the controller can take in term of maximum number of users before crashing, Fig. 5.10, the number of requests send by each user per second, Fig. 5.11, and the response time, Fig. 5.12, are performed. As explained in the previous sections, for each user a maximum number of three commands were sent in order to connect a user to a smart space (as a remainder "arp discovery", "create route" and "send a packet to the gateway"). Hence, a first test is performed by sending only one command, the second test with two commands and the last test with three commands. In each test we evaluated the maximum number of users that can user the controller in parallel. However, we identified a bottleneck regarding the number of users since in the first two test, only 100 users can be launched simultaneously, and beyond that the system crashes. And for the last test, when sending three commands by each user, the limit decreases significantly to only 20 users, which raises scalability and availability issues that need to be improved in future works. However, it

is worth mentioning that the performance are also influenced by the chosen controller, since, it is still under development and closed source. Also, the number of users should represent just the entities (e.g. service provider managing their SS and not any client) that interacts in parallel with the northbound interface of the SDN controller (represented by the Server). However, the problem persists since the REST API is exposed to the Internet.

The first test is to send a single command to the controller, the command will be the "arp discovery".

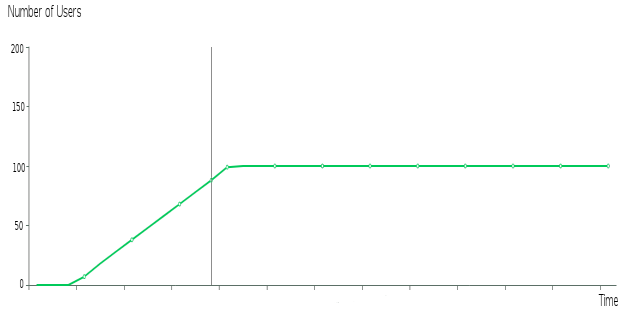


Figure 5.10 – Number of generated users before crashing

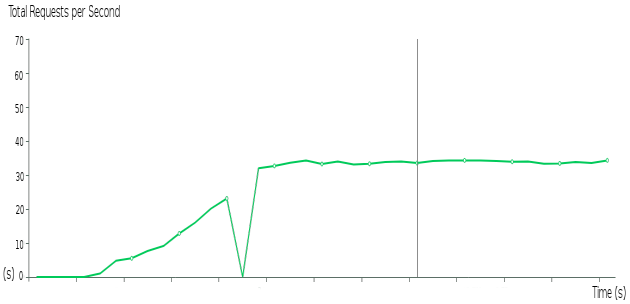


Figure 5.11 – Number of requests per second

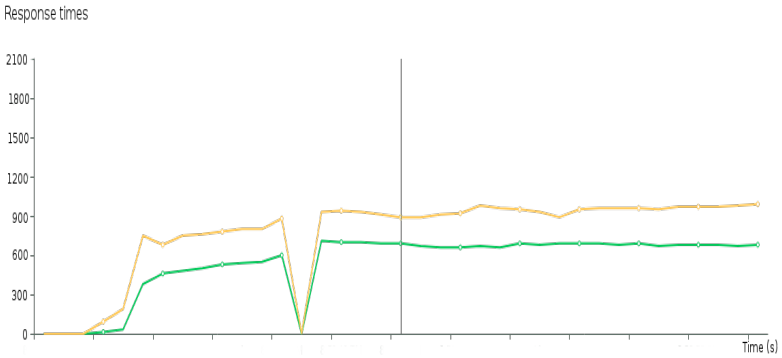


Figure 5.12 – Response time, the green curve represents the median response time and the yellow curve represents the 95% percentile

The Fig. 5.13 represents statistics about the requests sent to the controller. It contains the number of successful requests, the number of failed requests, the median response time in millisecond (ms), the average response time (ms), the minimum and maximum response time (ms), the average content size of the packets (bytes) and finally the number of requests per second.

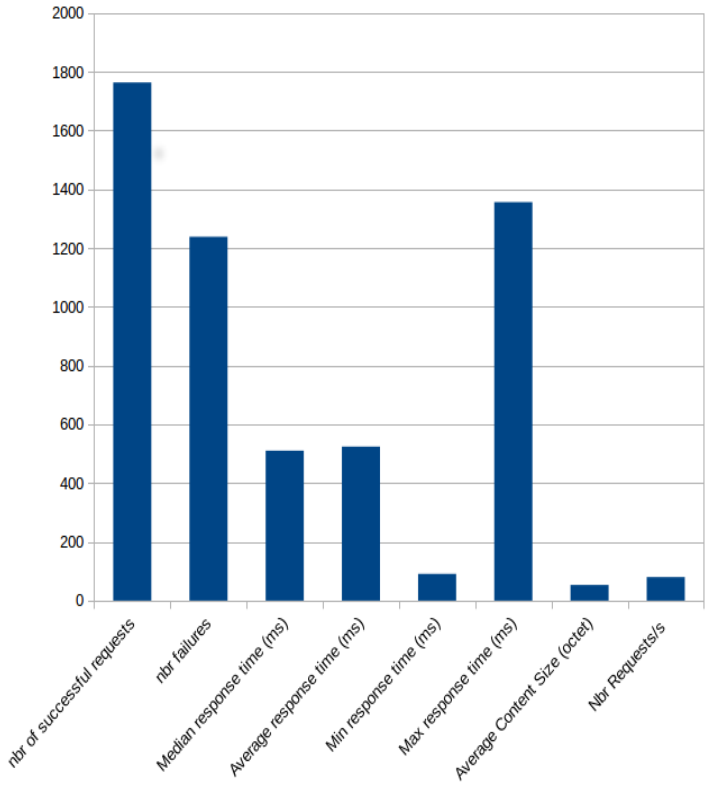


Figure 5.13 – Requests analysis

Next, the impact of the load analysis on the performance of the host is analyzed in the following figures, Fig. 5.14, Fig. 5.15, Fig. 5.16, Fig. 5.17, Fig. 5.18, Fig. 5.19, Fig. 5.20. It shows that when the maximum number of users reached, the CPU becomes very stressed and saturated, and the RAM uses up to 3 Gb of memory just to handle the user's requests. Moreover, the Fig. 5.17, Fig. 5.18, Fig. 5.19, Fig. 5.20, shows high amount of network packets sent and received in the overall network, and also for each OFS, which can be explained by the emission of the APR request in the network and the respective replies from the different OFS. For this reason, the ARP command is done only once, when the controller is first launched.

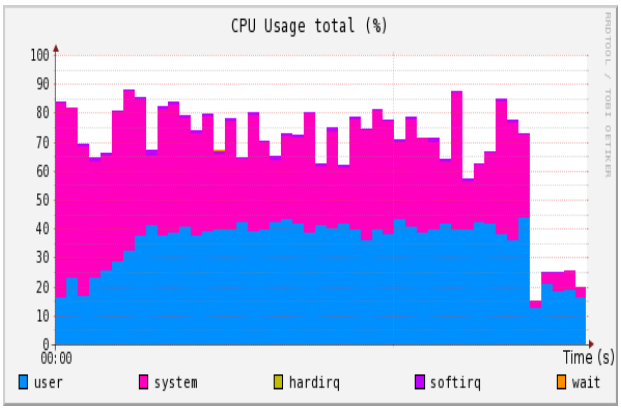


Figure 5.14 – Total CPU usage

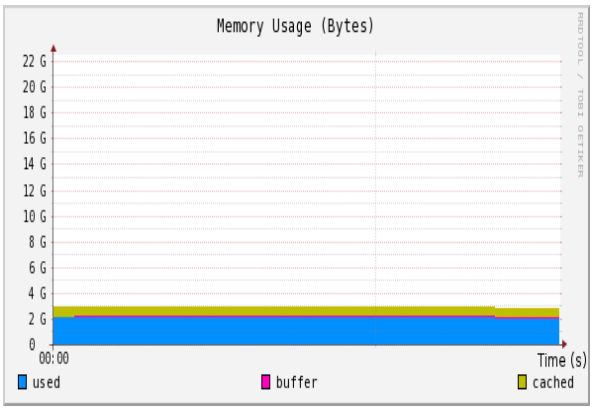


Figure 5.15 – Total memory usage

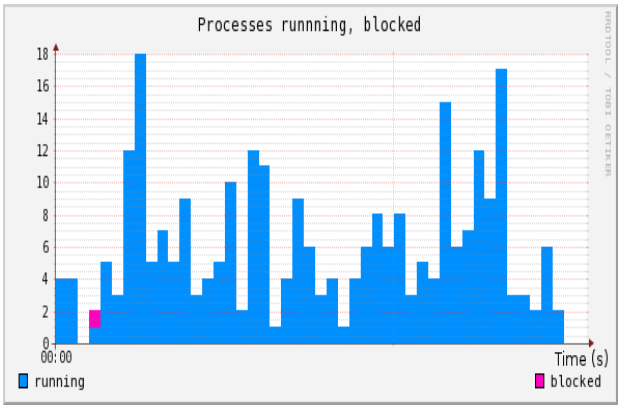


Figure 5.16 – Number of running processes and the blocked ones

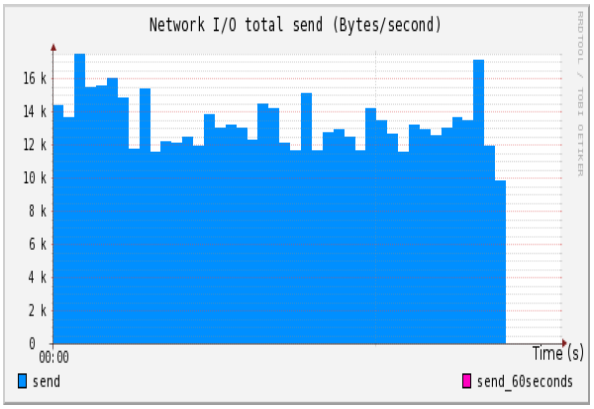


Figure 5.17 – Network I/O total sent (Bytes/second)

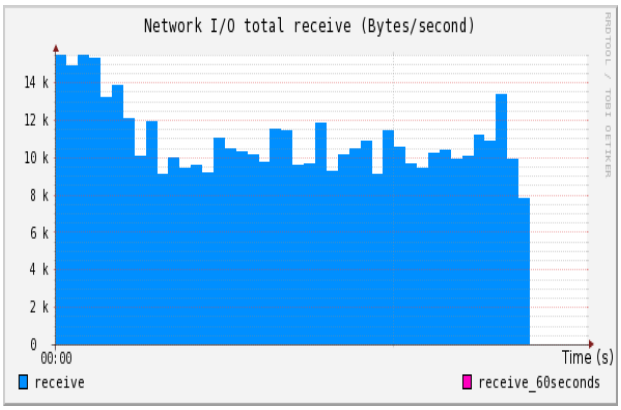


Figure 5.18 – Network I/O total received (Bytes/second)

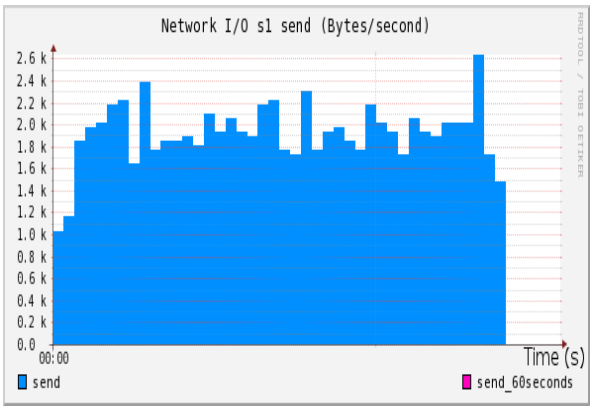


Figure 5.19 – Network I/O switch S1 of the miminet topology send (Bytes/second)

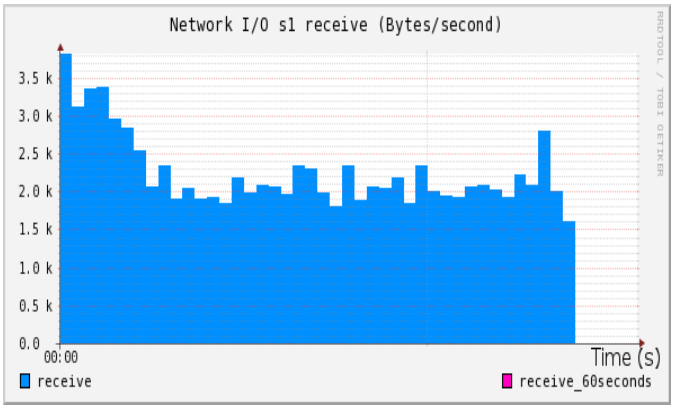


Figure 5.20 – Network I/O switch S1 of the miminet topology receive (Bytes/second)

5.8.1.2 Security evaluation

Using the OWASP Zed Attack Proxy (ZAP) tool [145], a set of web attacks were setup against the controller in order to evaluate the resistance of the controller to the traditional type of attacks. In addition to the previous results, which show that the controller is vulnerable to DDoS attacks, since beyond 100 simultaneous users the controller crashes. And the result raised 11 alerts, as shown in Fig. 5.21:

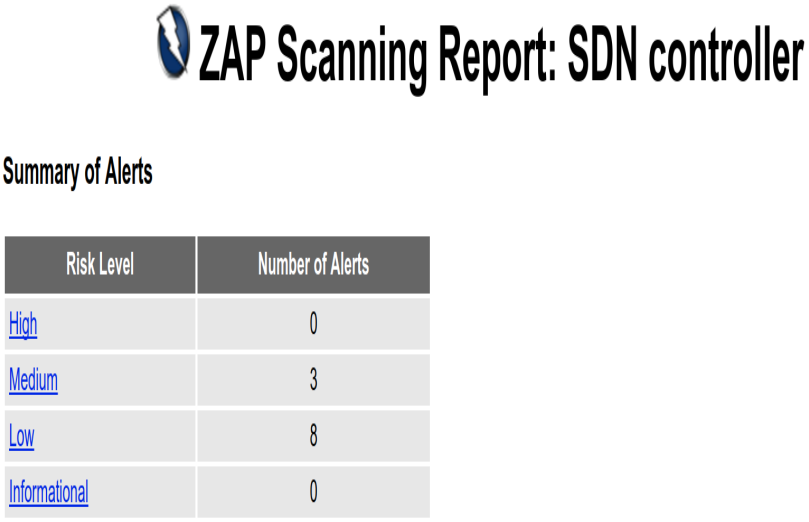


Figure 5.21 – ZAP scanning result

- High alert (based on the previous tests):
  - DDoS attack
- The medium alerts:
  - 2 alerts of : Application Error Disclosure
  - X-Frame-Options Header Not Set
- The low alerts:
  - Content-Type Header Missing
  - 2 alerts of : Cross-Domain JavaScript Source File Inclusion
  - X-Content-Type-Options Header Missing
  - 2 alerts of : Web Browser XSS Protection Not Enabled
  - Password Autocomplete in Browser
  - Private IP Disclosure

5.8.2 Analysis of the Server hosting the security services

5.8.2.1 Performance evaluation

In this analysis, and using the locust tool, 10000 users are generated with a ratio of 5000 users spawned/second. The Fig. 5.22 represents the maximum number of users before crashing. The requests vary between a simple request to the web page of the server, a user creation, only an authentication and finally an authentication plus an authorization request. The number of generated requests and the response time variation are presented respectively in Fig. 5.23 and in Fig. 5.24.

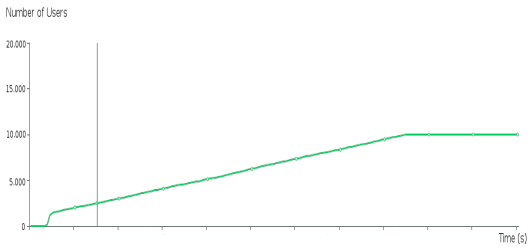


Figure 5.22 – Number of generated users

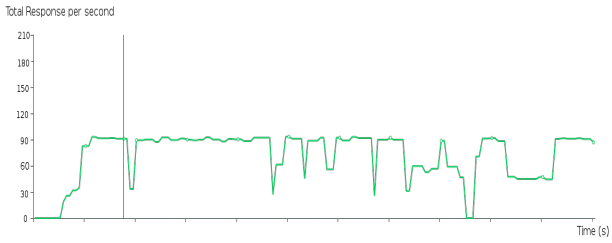


Figure 5.23 – Number of requests per second

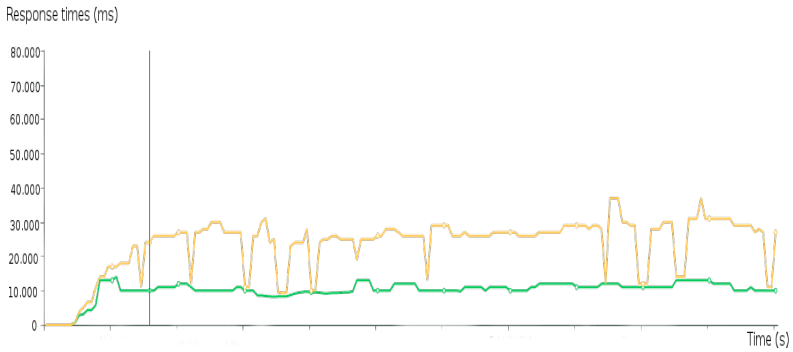


Figure 5.24 – Response time, the green curve represents the median response time and the yellow curve represents the 95% percentile

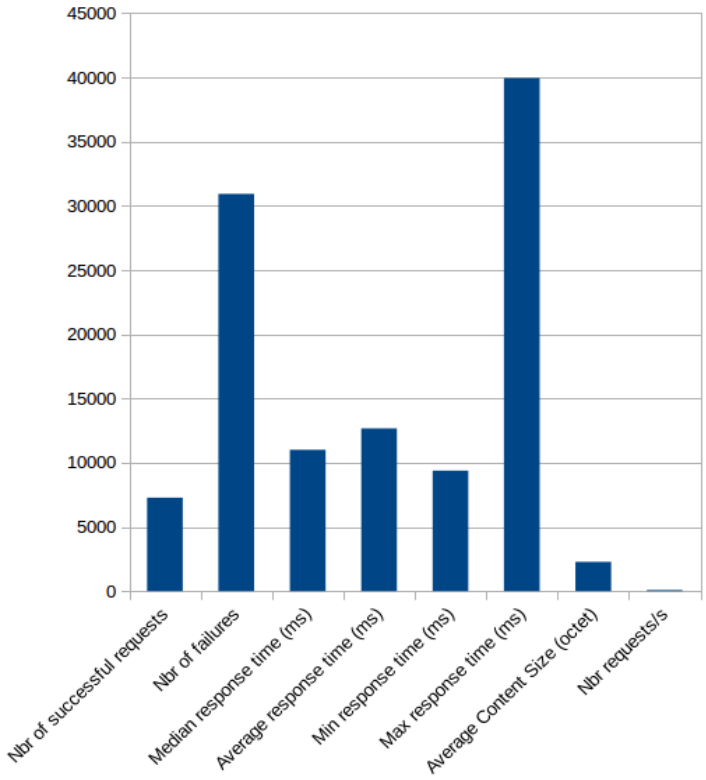


Figure 5.25 – Requests analysis

The Fig. 5.25 represents statistics about the requests sent to the authentication and authorization server. It contains the number of successful requests, the number of failed requests (since we consider also tentatives of wrong authentications and the repeated authentications using the same credentials), the median response time in millisecond (ms), the average response time (ms), the minimum and maximum response time (ms), the average content size of the packets (octet) and finally the number of requests per second.

The results shows that the number of failed requests is high which can be justified by the errors raised by the wrong authentications and the repeated authentications using the same credentials, which is banned by the application, in addition to the saturation of the server when the number of users is higher than 10000. Next, the impact of the load analysis on the performance of the host is analyzed in the following figures, Fig. 5.26, Fig. 5.27, Fig. 5.28:

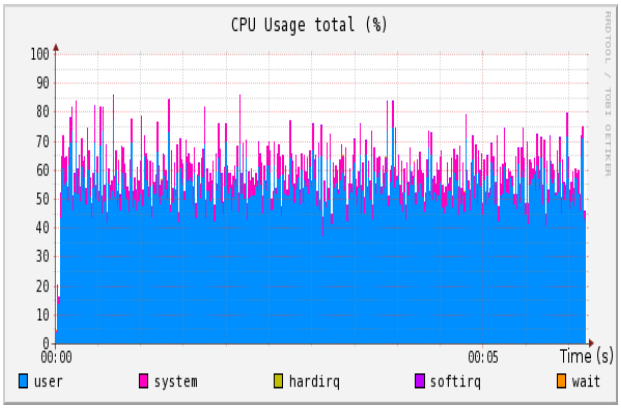


Figure 5.26 – Total CPU usage

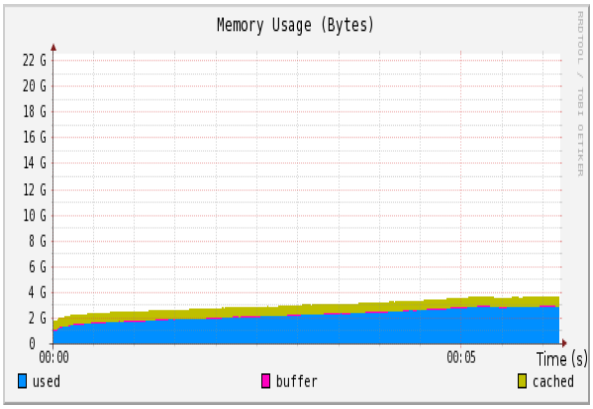


Figure 5.27 – Total memory usage

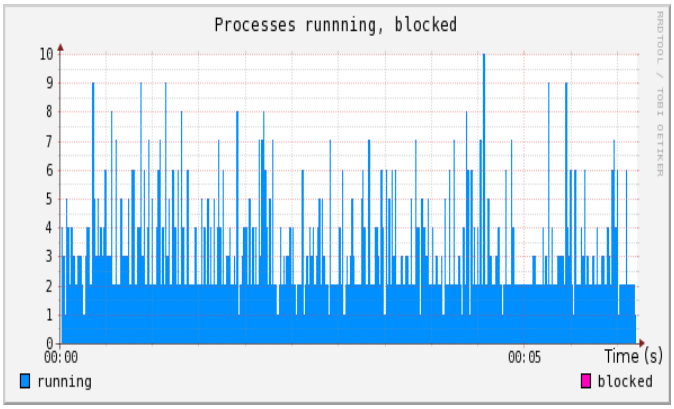


Figure 5.28 – Number of running processes and the blocked ones

The RTT is also measured, by sending one request per second from the client to the requested resource in the smart space and measuring the time for the RTT. The Fig. 5.29 shows the response time per request. The results shows that the response time gets higher with time, which can be justified by the raised number of connected users to the server.

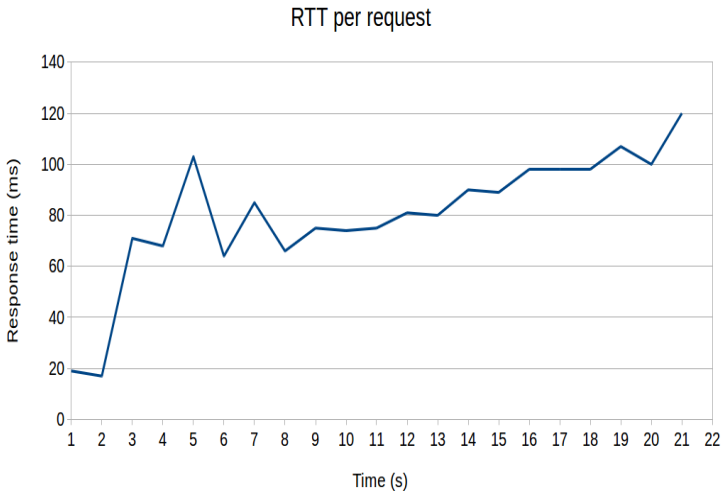


Figure 5.29 – RTT (ms)

5.8.2.2 Security evaluation

Using the OWASP Zed Attack Proxy (ZAP) tool [145], a set of web attacks were launched on the Server in order to evaluate it's resistance to the traditional type of attacks. And the result raised 5 alerts, as shown in 5.30:

ZAP Scanning Report: the authentication and access control servers

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	1
Low	4
Informational	0

Figure 5.30 – ZAP scanning result

- The medium alerts:
  - Application Error Disclosure
- The low alerts:
  - Cross-Domain JavaScript Source File Inclusion
  - Password Autocomplete in Browser
  - Private IP Disclosure
  - Web Browser XSS Protection Not Enabled

Thus, the results shows that there are no serious and severe threats that can threaten the Server. However, these alerts should be taken into account in the future patches or updates of the server. As future work, more advanced security checks, with more advanced attacks, need to be done in order to ensure the robustness of the Server.

5.9 Discussion

In this section, we discuss mainly the future directions of this work in providing a complete and robust architecture that respects the IoT requirements, and in particular the high demands and the enormous flows of data. However, even with the previous arguments regarding the use of SDN to manage these traditional issues in a more complicated scenarios such as for multiple smart spaces, a stronger proof with a better evaluation needs to be done. Hence, in this section, we will mainly discuss the scalability issues, which is today one of the main concerns in these architectures, the performance evaluation and the security and privacy issues regarding this proposition.

5.9.1 Scalability and reliability issues

In the system as it is now, the SDN controller represents a bottleneck to the architecture, for several reasons. First, because of the high traffic and high demand characterizing the IoT networks and how the users will be able to freely access their smart spaces, and delegate the access, anytime and anywhere. Secondly, because of the increasing charges in order to maintain and manage such networks, which needs additional computational power and storage. Third, to be resilient against the distributed attacks such as DDoS, which represent a real threat in such systems. Finally, it is clearly a Single Point Of Failure. Hence, we argue that a solution including the redundancy [146] and the collaboration between several controllers, together with a load-balancing may solve the issue, as shown in Fig. 5.31. Moreover, an important component need to be added, which is the Memcached [147], in order to provide a shared database to the different controllers in order to have the coherence and the uniqueness of network control commands.

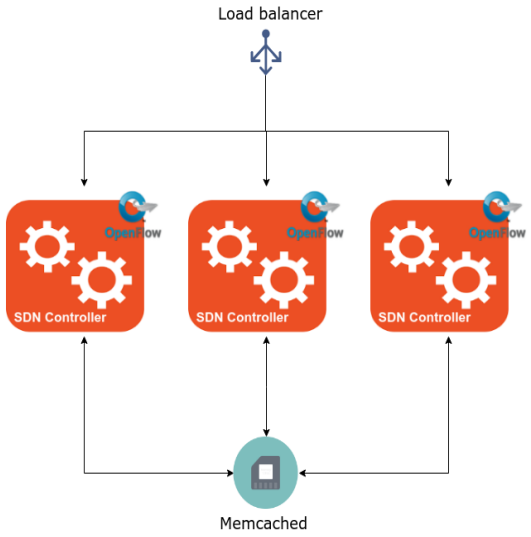


Figure 5.31 – Scalability management of the SDN controller

5.9.2 Performance evaluation, Security and Privacy

An important step should be taken into account for the future works, is the improvement of the performance of such architecture, particularly regarding the SDN controller in order to be able to manage huge amount of requests, and to be able to deliver a resilient and reliable services for IoT networks. First, these functionalities need to be implemented in another open source controller, which for sure provides very optimized and efficient functions rather than a test version of a close source SDN. Secondly, by solving the scalability and the single point of failure issues, we argue that we can have better performances.

Secondly, a deeper security analysis of the architecture needs to be undertaken in order to guarantee the robustness of the system against the attacks and to be able to mitigate them in the worst case. Several attacks have been identified by the researchers that targets the SDN controller, mainly the DDoS attacks [148, 149], and all the attacks that may target the northbound interface of the controller, since it exposes a RESTful API, and in particular the web related ones, as we identified in the previous section. Hence, the architecture needs to undertake stricter tests using more advanced security tools.

Finally, a privacy and risk analysis needs to be considered, in particular in the light of the new GRPR law which requires such process in any system that process personal data. Hence, the system should be able to protect the user's data, including the IoT related data gathered from the different sensors, all along its life cycle, during their processing. Additionally, it should respect all the privacy principles, to mention, data minimization, limited storage, integrity and confidentiality, and so on and so forth.

5.10 Conclusion

Managing the security of multiple smart spaces containing numerous and heterogeneous connected devices is the main concern of this chapter. A solution to solve this challenge by using an SDN controller as a central security check point is proposed and elaborated all along this chapter. The SDN controller interacts with several security, monitoring, and QoS third parties, using its northbound interface, in order to manage the smart spaces. A special focus was given to the security mechanisms, such as the confidentiality, the integrity, the authentication and the access control. Furthermore, the proposition can be also used for a larger scale management of multiple smart infrastructures, for instance in a smart city. Moreover, the performance analysis provides the limits of our architecture regarding the number of users and the load on the system in the extreme case. Unfortunately, the results show an important disadvantage linked to the scalability and the availability of the controller, which need to be discussed and solved in future works. Thus, and as discussed earlier, in order to be able to manage more users, solution including multiple instantiations of the controller, load balancers and cache sharing (in order to manage the shared databases and memories between the different instances of the controllers, and to manage their different states) can be discussed. Also, since the current state of the architecture create what is called a Single

Point of Failure (SPOF), we argue that this issue can be also mitigated using one of the solutions proposed in the literature, such as adding redundancy to the SDN controller with synchronization mechanisms, or by using a master/slave solution [146], which can be also analyzed in future works.

Currently, we are investigating the implementation of the previously mentioned modules in an open source controller called Floodlight [150], in order to compare their performances and to improve the proposition. We argue that the latter one is more mature and provides more functionalities and robustness since it is maintained by a larger community.

Chapter 6 | Business model and market analysis

Contents

As a part of EIT digital doctoral school programs [10], which provides a training to acquire a mindset for Innovation and Entrepreneurship (I&E), this chapter was introduced. The main aims of this program, is to link the academic field to the market, and to avoid keeping the result under the research shelves.

This chapter presents a business and market analysis of one of the use cases presented in chapter 3. More precisely, it concerns the remote consultation use case in the section 3.4.2, where a patient can remotely interact with a doctor using a WebRTC session enhanced with contextual health information coming from IoT medical sensors. After several interviews with doctors in the CHU ("Centre Hospitalier Universitaire"/"University Hospital") of Rennes, we observed that there is raising need regarding the telemedicine services, in particular regarding the remote follow-up of the patient. This issues becomes more convoluted in rural areas, and for the dependent,

elderly and persons with chronic conditions.

Hence, since the proposed use cases are, indeed, in the heart of the needs, particularly in France, a business and market analysis of the remote consultation use case was conducted, with a particular focus on the regulation and the needs in French health system. For this purpose, a project was created called "SIRONA" (which refers to a goddess, in Celtic polytheism, a healing deity and associated with healing springs).

In this chapter, the value proposition, the business and market analysis, the different scenarios, and the regulations regarding the data protection (with a special focus on the health related ones) of the project, are provided. The technical details of the use case can be found in the chapter 3.

6.1 Introduction

Rural areas usually have several problems regarding the access to medical services such as hospitals, emergency relief centers, specialized doctors, and even ambulances in some areas. is also due to the medically deserted areas, where most of practitioners goes to big cities since there is more opportunities, more convenience and more benefits. Moreover, Europe faces the problem of an aging population. As a consequence, more and more dependent persons need medical assistance, and especially to have medical experties from specialized doctors. Furthermore, their reduced mobility adds more challenges. In facts, practitioners confirmed that for medical follow-up, several cases do not need to be physically present, and a simple visio-call can be sufficient.

Currently, there are several diagnosis done using tele-consultation system for persons with chronic conditions, in particular, based on the interviews, in dermatology and psychiatry. The efficiency of such system proved it success through several applications in rural areas such as with Saint Pierre-et-Miquelon (Canada), Belle-île (France), etc. Traditionally and currently, medical data, such are patient records, ECG, echography, scanner, picture, etc., are shared through a specific platform owned by the specific hospital or a set of collaborating medical entities. Based on these data, experts can provide answers to the requesting peer which can be the patient itself or a doctor requesting tele-expertise.

Moreover, different type of interactions between the patient and the doctor were identified:

- Patient-to-doctor interaction without any external help.
- (Patient & nurse)-to-doctor interaction where a nurse (or even another doctor) can be present on the patient side and communicate remotely with a doctor.
- Doctor-to-doctor interaction in order to discuss or ask for advice concerning certain a patient without the presence of the patient.

Hence, the main pain points targeted by this project are presented. First for the patients:

- Facilitate the access to the medical services, particularly in the deserted areas.
- Simplify the medical follow-up for the patients, and particularly for the elderly, the dependent persons and the patients with chronic conditions.
- Reduce the transportation, costs, time, and tiredness.
- Encouraging patient to respect their follow-up.
- Being always in contact with the family doctor.

And for the medical staff and society:

- Reduced mobility for the medical staff, since the tele-consultation may be done instead of physically visiting an elderly house or a patient. However, this case replaces only the normal cases. The solution does not replace the necessity of a physical presence of a doctor for more complicated cases, and particularly the urgent ones.
- Encourage the patients to respect their follow-up, helps the doctors providing better healthcare, and to fulfill their duties.
- Following their patients efficiently.
- More frequent appointments (i.e. for patients with chronic conditions).
- Covering medical desert.
- Reduce healthcare services costs.

6.2 Value proposition description and validation

Based on the analysis of the health market and on personal experience with doctors, especially with specialized doctors, the project proposes mainly to bring the basic medical services "simple consultation" to the patients. The targeted users segment is wide, since, anyone with a need to see a doctor without having to be physically present is concerned. However, the elderly and dependent persons are the ones who need more attention. Hence, the segment is narrowed down of these types of patients.

The main idea is to provide remote medical support for elderly and dependent persons in order to solve difficulties to access medical services in particular in rural areas. Moreover, one of the main values of our project is to facilitate the communication between the patients and the medical support (doctor/nurse/etc.), and also to be able to detect anomalies and call/notify the emergency service as soon as possible. Moreover, the link between the patient and the regular doctor (or family doctor) is very valuable. With our proposition, this relationship of trust with the regular doctor is strengthened, by offering possibility of remote consultation, for instance in case where the patient is

moving or traveling.

The main characteristics of the proposition are:

- The newness based on a literature analysis and a preliminary market analysis.
- The design of a new architecture specially for these needs.
- Cost reduction by reducing the mobility of the patients as far as possible.
- Time saving for both the patients and the doctors.
- Personal/physical energy saving.
- Ecosystem saving (less need to use transportation).
- Social and emotional value proposition, in particular for patients that suffers from chronic and complicated diseases, or for the elderly that wants to be as close as possible to their home and family.

6.3 Business model description and validation

6.3.1 Solution description

As explained in the section 3.8.7, the project is based the tele-consultation. The solution proposes the possibility for the medical practitioners to remotely consult their patient. Upon doctor recommendation, the patient can use additional medical IoT devices in order to collect health related data such as the vital signs. Hence, the strong points of the solution are mainly:

- The use of medical IoT sensors.
- The visio-conferencing system.
- The transfer of these medical data in real-time using WebRTC.

On the other side, the doctors are provided with a platform which aggregate all their patients data, together with a facilitation to access their medical records. Hence, from the doctor side, other functionalities can be found such as:

- Visio-conferencing system.
- Analysis of the medical data (potentially through machine learning as future work).
- Patients management system.
- Secure storage and access to the medical files.
- Prescription generation.

6.3.2 Business model and lean model canvas

The business analysis of the proposed solution is based on two main strategies: the Business Model Canvas, and the Lean Model Canvas. In this section, only a summary of the commune parts of the two models is provided, after a brainstorming of all the possible ideas and suggestions. Mainly, the analysis concerns the customer segment targeted by the solution, which is presented in subsection 6.3.2.1. The customer relationship, which defines how the service provider should maintain a good relationship with the clients, in order to ensure the client satisfaction, is presented in subsection 6.3.2.2. And finally, the channels, which defines the delivery methods of the solution to the client. Both business and technological sides are analyzed in subsection 6.3.2.3.

6.3.2.1 Customer segments

- Patients: Elderly; Dependent; Persons with chronic conditions; Persons in rural areas; Isolated patients; Patients having health issues but still wants to appear valid to other persons; Valid people but with antecedent health issues (heart attacks in the past); Children recovering (that don't want to skip a whole day and skip classes just to see a doctor for a 15 minutes).
- Medical Third parties: Healthcare systems; Health insurances.
- Medical support: Doctors in rural areas; Doctors specialists; Nurses; Hospitals; Emergency relief centers; Dependent persons.

6.3.2.2 Customer relationship

- Hot-line for emergency assistance for the proposed application/platform.
- Maintenance of the availability of the product.
- Getting feedbacks from the users and improve the product base on that.

6.3.2.3 Channels

- Business side:
  - Proposing the solution directly to medical practitioners and hospitals.
  - Participating in seminars, conferences, workshops, and so on, in order to introduce the product.
  - Proposing trainings to the potential customers.
- Technological side:
  - Web/mobile easy to use application.
  - Based on open source solutions.
  - Through the Internet.

6.3.3 Potential stream of revenue

The revenue will be mainly based on a subscription model to our services. Three customers of different size (insurance, hospital, liberal practitioner) were considered for this subscription model. The solution should be then tailored and adapted to the special need of each type of customers. Moreover, several additional services/features that our solution could provide, can impact the subscription cost, such as:

- Planning/agenda service.
- Storage of the records service.
- Proposition of the available doctors in the surrounding areas.
- Shipping medicine (with health regulation certification).
- Storage of medical/personal data (with health regulation certification).

6.3.3.1 Revenue stream: optimistic PnL forecast

In this forecast, an optimistic estimation of the best case situation is performed, where big contracts with hospitals are made. An even better case can be having a contract with a new hospital each year. A successful contract is represented by a peaks in Fig. 6.1. Hence, the following graph represents the estimated revenues forecast for the first 3 years:

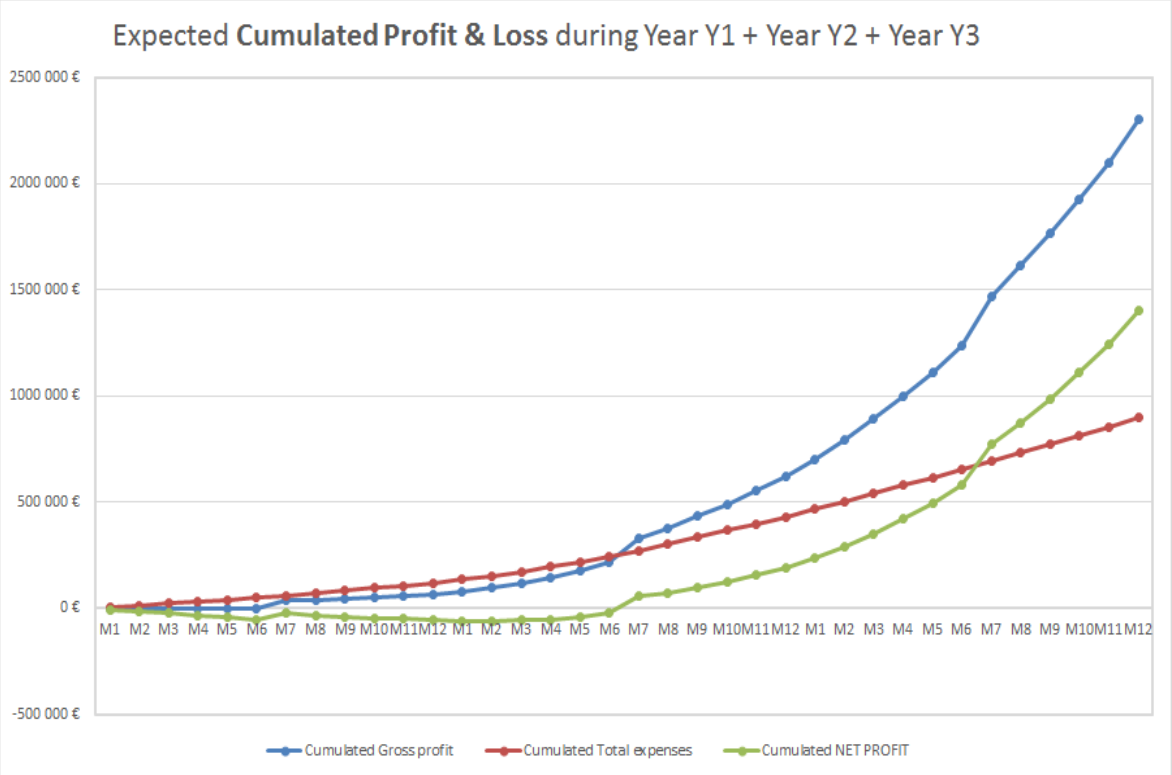


Figure 6.1 – Revenue stream: optimistic PnL forecast

6.3.3.2 Revenue stream: pessimistic PnL forecast

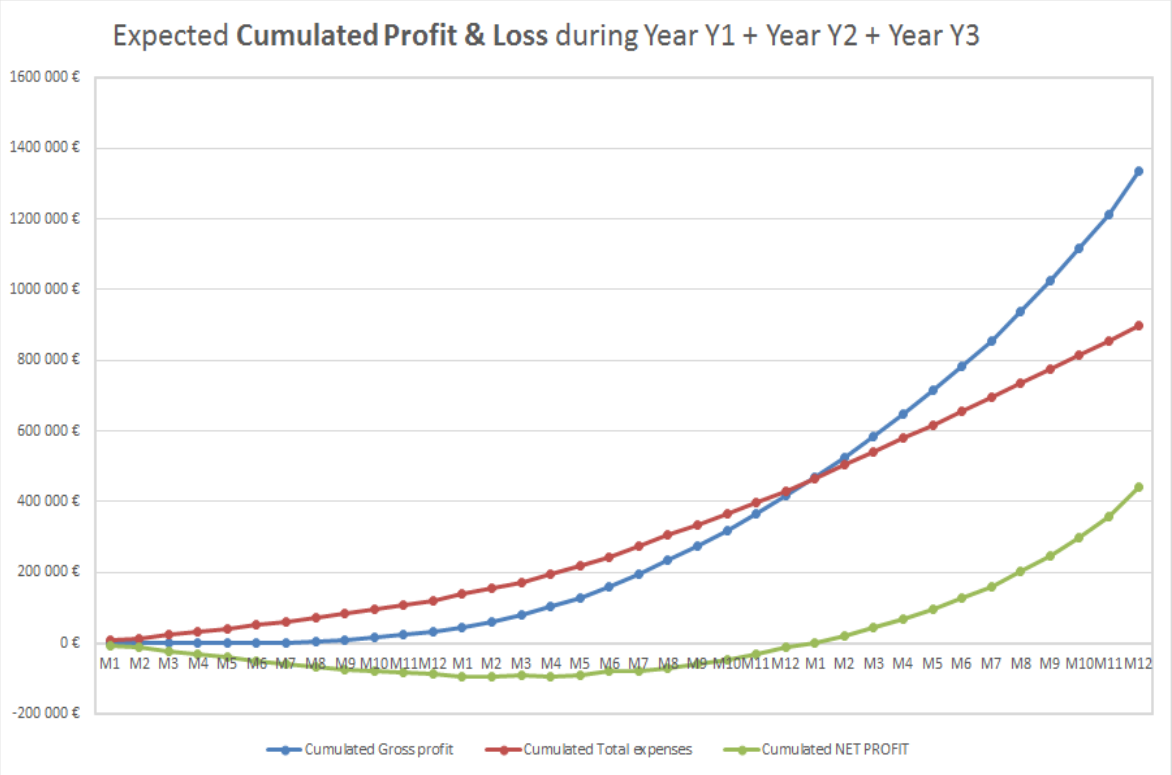


Figure 6.2 – Revenue stream: pessimistic PnL forecast

In this forecast, only contracts with private practitioners are made. The target is to have at least contracts with 1% of the total number of French practitioners in France (around 1000 doctor). Hence, the Fig. 6.2 represents the estimated revenues forecast for the first 3 years.

6.3.3.3 Revenue stream: pricing policy

The main pricing policies that can be targeted with our project are:

- Subscription:
  - Hospital subscription: annual subscription.
  - Practitioners subscription: monthly subscription.
- Behavioral
  - Selling anonymized data to third parties (pharmaceutical enterprises, sensors enterprises, etc.).
- Other
  - Percentage of the consultation.

6.4 Market analysis

In this study, a market analysis regarding the main actors/companies that provide tele-consultation services, was done. The results show a large number of companies in US with huge turnover and large offer of tele-consultation services. Meanwhile in France, there are very few companies and there recent one. The next table shows the data collected regarding some of the competitors around the globe:

Competitor name	Turnover	Social Capital
Médecin Direct	850 300 € (2015)	87 100€
Qare	N/A	225 000€
Linkéos	N/A	96 000 €
Teladoc	493 M$	N/A
Mdlive	13 M$	N/A

Table 6.1 – Market analysis

6.4.1 Competitors

For the main competitors, the study is narrowed down to only the French ones. Considering their already implemented solution, different distinctions from our approach were detected, as explained in the following table:

Competitor name	Services	Strength	Weakness	Differentiation
Médecin Direct	Pool of practitioner available for remote consultation	Insurance partnership	Only customer with partnership insurance can access	Insurance oriented
Qare	Monthly subscription to access remote consultation	No partnership with insurance	Android app. Visio embedded in navigator	Customer oriented
Linkéos	Offer remote devices and kit for different case of medical follow-up	Medical sensor certified by health regulation		Hospital oriented

Table 6.2 – Competitors analysis

6.4.2 Risks management

Identifying the main risks regarding the provided solution is the main purpose of this section. Following several discussions, tutoring and interviews, several limitations were raised and which should be taken into account in order to achieve the project's goals. The main purpose is to assure the uncertainty raised by several factors, which can be economical, social acceptance, technical, and so one, in order to maximize the realization of the opportunities. Thus, the main identified gaps and barriers are:

1. The desire to use technology and to substitute (which is from the point of view of the patients, whereas the proper word of our proposition is "complete") the direct interaction with the doctor/patient. Since, indeed, the best way to do a consultation is by physically be present in front of the doctor. Hence, one of the main risk that needs to be managed is to target a specific cases and customers segments that will adopt this technology.

2. The cultural and social aspects, since most of the patient are used to see the doctor physically. Moreover, for the elderly, physically seeing and discussing with a doctor is also a way to interact, discuss and socialize with other persons.

3. Security, trust and privacy are one of the main concerns of both the patients and the doctors. The provided solution needs to be totally secure since it uses very sensitive data directly linked to the health of the patient, and which also can threaten its own life. Additionally, the privacy of these data needs to be guaranteed. The European committee requires the respect of the privacy by default and the privacy by design principles.

4. And finally, the regulation issue. In particular regarding the use of the IoT medical devices. This can be due to the fact that there is still no precise consensus on the difference between a medical device and a well-being device. Moreover, most of the current medical devices present on the market are not regulated. And finally, there are questions about the trust issues regarding the collected medical data.

6.4.3 Partnership model

In order to define the partnership model of a company, several questions need to be answered:

- What is it you are looking for? Money? Clients? Talent?
- How much control do you want?
- How much risk can you handle?
- Can you manage it?
- Will it add value?

Some of the main models are: Acquisition; Minority Interest (ownership); Joint Venture; Alliance; Franchise.

6.4.3.1 Selected model for our project

After answering these questions:

- What is it you are looking for? Money? Clients? Talent?: For our project what we are mainly looking for is money and clients.

- How much control do you want?: The different parts of our platform: the communication channel, the management of the sensors, the analysis and the computation of the medical data, the storage of data, and finally the security and privacy within the different parts of the platform. Hence, we want to have full control over the value chain. However, outsourcing one or several parts of our system is also possible.
- How much risk can you handle?: Since we are dealing with sensitive medical data, the risks need to be as minimal as possible.
- Will it add value ?: The added value is the expertise of the partner, for instance a better management of the security, or a better management of the storage.

Hence, for this project, the best partnership strategy can be "joining ventures" with other companies or experts.

6.4.4 Entry point to the market

Identifying the early adapters of a technology is a very important step in order to push the product into the market. Hence, since the project targets mainly elderly and dependent persons, a possible entry point can be the EHPAD, which is a French system helping elderly people suffering from the lack of autonomy. Since, they have more or less a financial autonomy, and they are always looking for innovative solutions in order to help their patients. In addition of having their own medical staff, which can be benefit from the proposed solution to have better communication with other doctors such as for remote expertise.

An interviewed doctor mentioned that in these facilities, elderly were transported often to hospitals, using ambulances, some time for long distance, just for a simple medical control, or follow-up. Some of these checks can be performed via tele-consultation, with the help of the local medical staff (i.e. a nurse).

On the other side, targeting directly the liberal practitioner would be more difficult. Indeed, they have their own affinity to technologies, however, the use of tele-consultation may not be the first interest/priority for them. Nevertheless, with a support from social security and being involved in a "Plan Régional de Santé" (Regional Health Plan) could help the project gain more visibility and acceptance. Also, with the movement in the health regulation regarding the medical IoT devices, and with the possibility to cover their cost with the social security, the project may become more attractive since it is positioned at the heart of the future needs.

6.4.5 User scenario schema

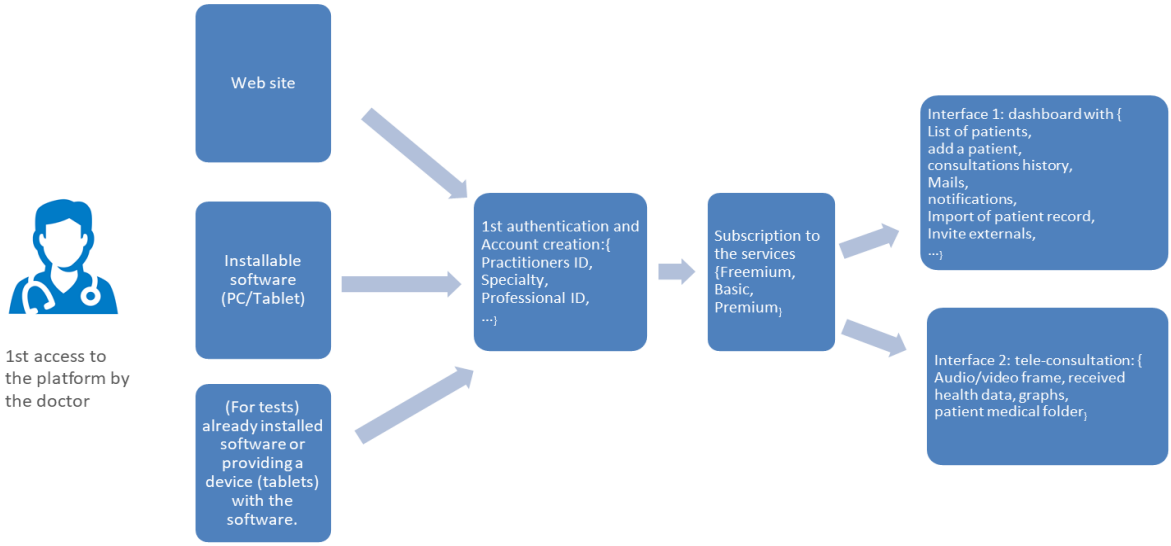


Figure 6.3 – first authentication of the doctor

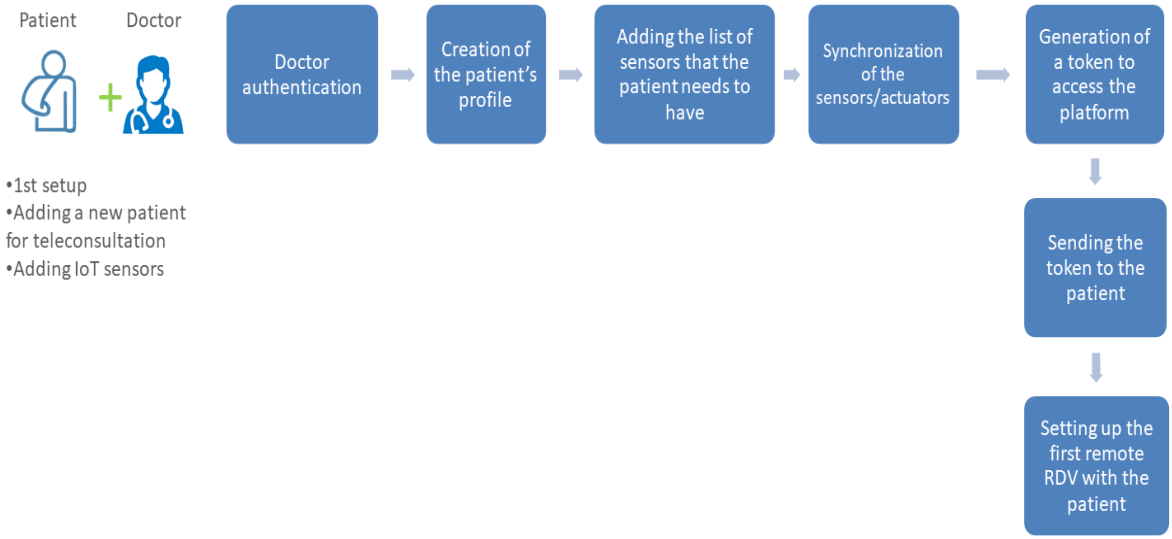


Figure 6.4 – first setup of the patient

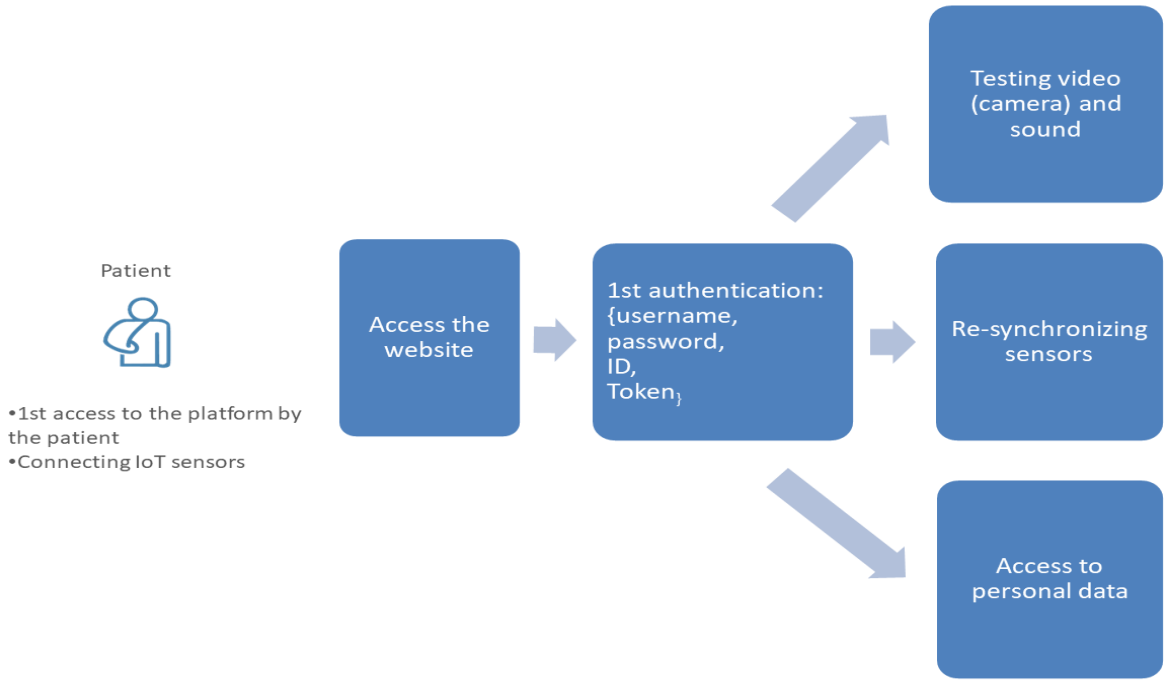


Figure 6.5 – first authentication of the patient

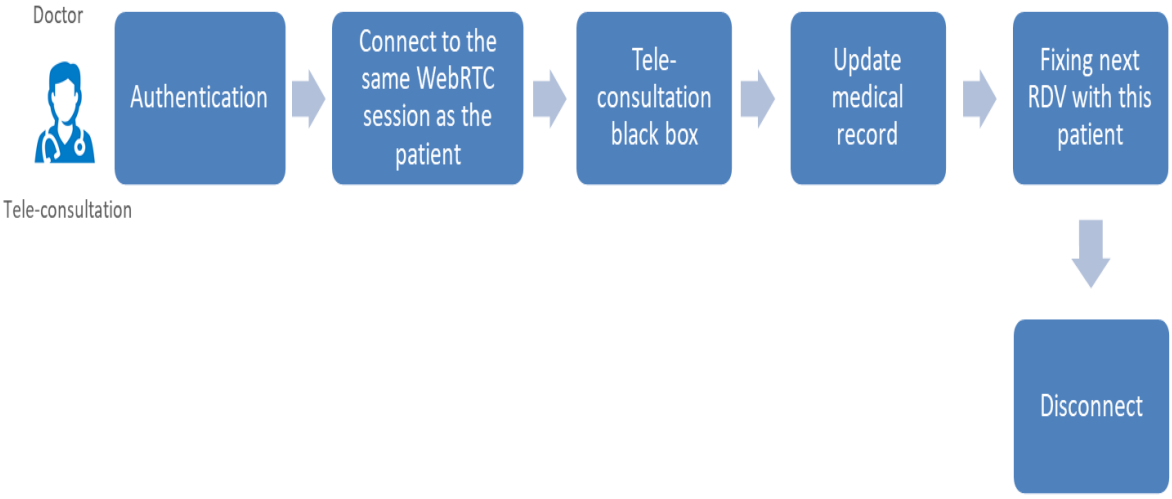


Figure 6.6 – Teleconsultation from the doctor side

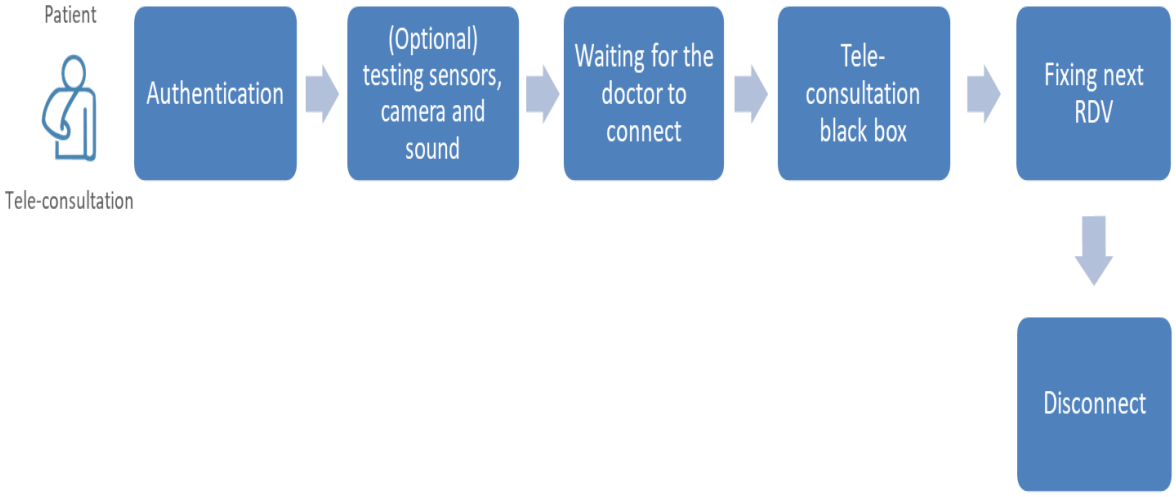


Figure 6.7 – teleconsultation from the patient side

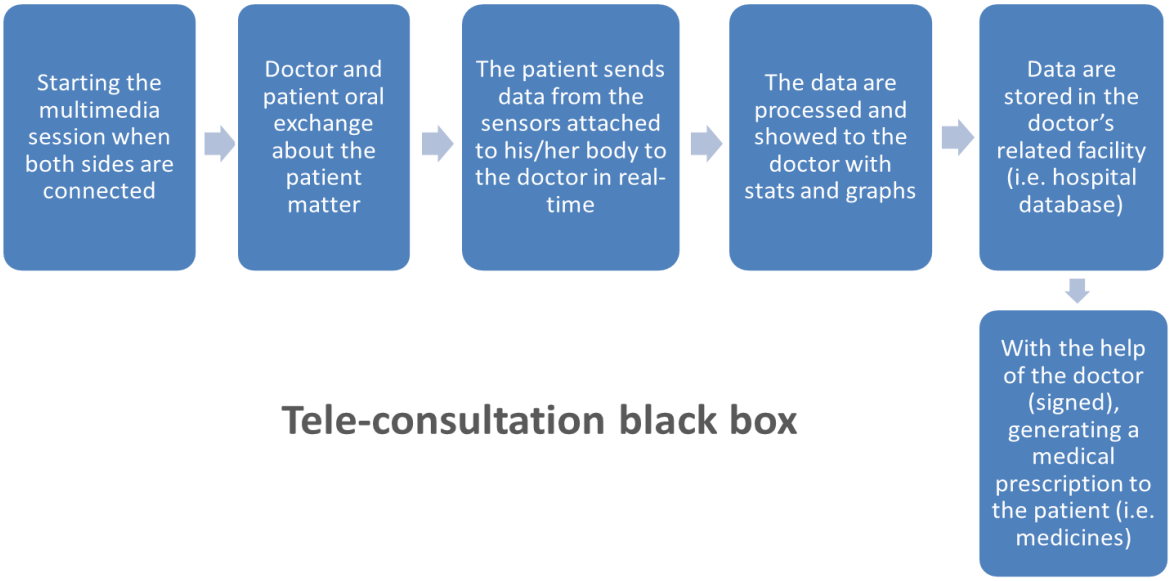


Figure 6.8 – teleconsultation black box

6.5 Data protection

6.5.1 GDPR main articles dealing with health data

6.5.1.1 Concerned parties

According to the GDPR, anyone in the EU who controls data and/or undertakes data processing is affected. The health sector will, hence, need to undertake a more holistic approach to data management in order to reduce the risk of this sensitive domain. The main principles that has to be applied to the personal data, and in particular to health related data are defined in [151] (Article5). They need to be always respected by the data controllers, and they are mainly:

1. Lawfulness, fairness and transparency
2. Purpose limitation

3. Data minimization
4. Accuracy
5. Limited storage
6. Integrity and confidentiality
7. Accountability

6.5.1.2 Processing

Hence, health data needs higher protection standards. As mentioned earlier, data concerning health has special mention under the new regulation of the GDPR. However, it is important to highlight that these kinds of data are subject to higher standards of protection than the general notion of personal data. Generally according to [151] (Article9), the processing of these forms of health data is prohibited unless one of several exceptions is present, which are:

1. "Explicit consent" of the concerned person (data subject). And in case of inability, other measurements can be applied. For healthcare systems, explicit consent requires always the strongest forms of agreement which needs to be clear and documented [151] (Article9).
2. If the data subject (the patient) makes his/her health data public.
3. If it is for the patient vital interest, and in particular if the patient is unable to provide explicit consent.
4. If it is for healthcare purpose, for instance for preventive or occupational medicine, for the assessment of the working capacity of the patient, for medical diagnosis, for the provision of health or social care or treatment, or the management of health or social care systems and services.
5. If it is for public interest, in particular in the area of public health, for instance in order to protect the population against a serious cross-border threat to health, or ensuring high quality standards and safety of healthcare, medicinal products, or medical devices.
6. And finally if one of the EU member States introduces new laws/conditions into the GDPR regarding this kind of data.

Hence, when processing health related data, the data processors have to implement their data processing operation with respect to these regulations and to be very careful with these data in particular regarding the storage, the confidentiality and integrity and if there is consent or not to perform other operations.

6.5.1.3 Rights

The main significant rights introduced by the GDPR are:

1. The "right to data portability" in [151] (Article20): transferring health data from one controller to another.

2. The "right to be forgotten" [151] (Article17): Where personal health data can be erased, in particular in case of withdrawing consent, or when there is no longer a purpose for processing it in accordance with the principle of limited storage and data minimization.

3. Some significant changes in the "Subject Access Right" (SAR) [151] (Article15). The, healthcare organizations need to revise their SAR procedures and policies, since, under the new GDPR a SAR can be made free of charge and must be addressed quickly (i.e. can be within one month of receipt of the request).

4. Rights in case of a breach [151] (Article34): under the new GDPR law, reporting a security breach becomes mandatory. Breaches must be reported to a data protector regulator within 72 hours at most. Moreover, those affected by the breach must be informed. The health sector needs, however, to be more clear and effective procedures since they are dealing with sensitive personal health data. Additionally, the health actors should consider implementing efficient and strong breaches prevention systems, with strong security and privacy mechanisms.

5. Right to be informed/transparency [151] (Article13-14): data controllers must provide some information to the patients in a concise, transparent, intelligible and easily accessible form, using clear and plain language.

6.5.2 Security of health data regarding Working Party 29

With the expected benefits of the IoT, special focus needs to be given to the privacy and security challenges, which is the main concern of the WP29. Organizations which place privacy and data protection at the heart of product development are better placed to ensure that their services respect the principles of privacy by design, which is expected by EU citizens under the new laws [78].

Obviously, the rapid development of the IoT and the high adoption of this solution by individuals and by industrials raises new and significant risks regarding the personal data, which are usually collected by these IoT devices. These risks may be for instance the disclosure of these data to unauthorized ones, the unauthorized surveillance of individuals, or the use of these devices as a platform of attack, as recently seen with Mirai attack. Hence, the WP issued several opinions and recommendations in order to contribute to mitigate the risks which may be raised by IoT. These recommendations are addressed to the different stakeholder concerned such as the devices manufacturers, the application developers, the social platforms, the further data recipients, data platforms and the standardization bodies, in order to help them to integrate the privacy by design and data protection principles in their products and services.

In this analysis, a special focus is given to these point in order to protect the users of our platform:

- The platform must remain in complete control of the collected personal data throughout the processed data lifecycle.
- The consent of the users is mandatory.
- The use of the data must be clearly defined, freely available for the owner upon demand, and very specific to the targeted use in order to respect the data minimization principle.

Therefore it is important for SIRONA, as an IoT stakeholders, to make sure that these special points, together with the previously mentioned principles, are protected and that all the rights of the users are respected. Moreover, not only the security of the devices has to be considered but also the communication links, storage infrastructure and other inputs in the ecosystem. Based on the document provided by the WP29, some of the recommendation to all stakeholders can be summarized to:

- An analysis of the privacy of the system, via a Privacy Impact Assessment (PIA) should be performed before launching any new IoT applications to the market.
- Raw data should be deleted as soon as they have extracted the data required for their data processing.
- The product should apply the privacy by design and the privacy by default principles.
- Data subjects and users must always stay in control of their data.
- To provide a user-friendly interface to the users in order to express their explicit consent.
- Depending on the capabilities of the IoT devices, they should be able to inform the uses when their data are being collected/used/etc.
- OS and device manufacturers should inform the users on what type of data are being collected, what processing are done, and so on.
- The device should inform the stakeholder if a user withdraw his/her consent.
- A notice should be designed, by the application developers, to frequently remind users that sensors are collecting data.
- The application should be able to provide facilitation of access, modification and deletion of personal information collected by IoT devices.
- The developers should use a standard and usable format in order to export raw and/or aggregated data.
- Data minimization principle : only the minimum needed data collected.

6.5.3 ASIP Guideline

ASIP Santé (Agence des Systèmes d'Information Partagés de Santé / The shared healthcare information systems agency) [79], recommends that the interoperability of our proposed information system should follow the rules of "Système d'Information de Santé (CI-SISI)" regarding the health data, these system needs mainly to be able to:

- Technical interoperability of data and services with regards of security and confidentiality.
- Interoperability with the medical sectors standards and to make them understandable for every users.

Remote consultation, or tele-consultation, is regulated as a medical service and has to follow ethical rule of medicine. It cannot be presented as only a technical solution of substitution of the direct interaction of a patient and a doctor. For CNOM, telemedicine is not a way to repopulate deserted rural area but a mean to bring back proximity of medical service without moving. Telemedicine should also helps in coordinating all the medical services offered at different levels (nurse, doctors, hospitals and so on).

CNOM indicates that telemedicine could enhance the healthcare quality for the patients, by reducing unnecessary mobility, if a good coordination and communication with the medical expert and staff exists, and should follow these principle guidelines:

- Ethics in remote medicine:
  - User's right are a priority: information, consent, discretion of medical case. In case of the non consent for the remote service, patient has to be able to access real medical service.
- Remote consultation:
  - Patient should be able to know the identity, the location and qualification of the doctor. They should also know the consultation time and how they will receive the results of the consultations.
  - Doctors should not be professionally dependent on the remote services and has right to stop if he judge data or information not sufficient to perform their diagnostic.
- Insurance:
  - Doctors has to be covered by the insurance for this specific practice of consultation.
- Telemedicine contract:
  - It has to be explicit that SIRONA does not provide any medical services. The platform provides only the technological solution needed for the teleconsultation at the disposition of practitioners.

6.5.4 Privacy Impact Assessment for SIRONA

6.5.4.1 Definition

The PIA process is designed to identify and address the privacy issues of a particular project. By considering the possible consequences of a current or proposed action, by identifying any potential privacy risks and then examining ways to mitigate or avoid those risks. A PIA is best undertaken at the beginning of a project before any significant investment. The outcome of the PIA can influence the final design of the project. A PIA should also be carried out when a change to a project is proposed. Throughout the project, the PIA will need to be revisited, reviewed and updated when necessary to incorporate changes to the project as it progresses. The term 'Data Protection Impact Assessment (DPIA)' is used in the GDPR, where data protection is only one of the aspects of privacy [76][152].

There is a mandatory obligation to conduct a DPIA when a processing is 'likely to result in a high risk' to impact the rights and the freedoms of data subjects. Data concerning health is categorized as 'sensitive data' requiring a DPIA, which includes genetic, biometric or unique identifier data. A DPIA is required for each 'processing activity' [76] [152].

6.5.4.2 Objective and benefits

The primary objective of the PIA is to protect the privacy of the individuals whose personal health data are being collected and processed. Some of the main benefits of providing a PIA in health sector are:

1. Ensuring the protection of privacy rights of the patients.
2. Ensuring that the health organization complies with the regulations.
3. Building a trust relationship between the health organizations and the patients.
4. Facilitating the integration of the "privacy by design" into the implementation and the design process of the final product.
5. Reducing the cost, in particular the cost of the security and privacy since they will be considered by design.

6.5.4.3 Who/when should conduct a PIA?

It should be undertaken by the service provider and by the entities/persons having the right expertise and knowledge about the project and the regulation. In particular, the project manager and his/her team under the supervision of the DPO (Data Protection Officer). It should be done before any real processing of personal health related data.

6.5.4.4 The PIA process

According to [76], the PIA process aims at evaluating the privacy implication of projects and relevant legislative compliance. This means that, using this process the privacy risks can be evaluated in order to try to solve them prior to any incident. And according to the same guidance, there are five stages in the PIA process:

Stage 1. Threshold assessment

Stage 2. Identify the privacy risk

Stage 3. Address privacy risks and evaluate solutions

Stage 4. Produce a PIA report

Stage 5. Incorporate the PIA outcomes into the project plan

6.5.4.5 PIA results

The results of the PIA are presented in section 3.8.7, since the business model and analysis concerns one of the use case of the first contribution.

6.6 Conclusion

This chapter provided a business and market analysis of the SIRONA project, which is about the remote consultation for the dependent and elderly persons in rural areas. It first presents the added value of this project, followed by a business model together with a market. The added value of this project follows real needs in the hospitals, in particular in the CHU of Rennes, extracted from interviews with doctors from different specialties. The results show that:

- Solutions currently exist in the hospitals for remote expertise with other doctors, but most of the time remote consultation is done just by phone calls.
- These services do not exist in rural areas.
- There is a need for this solution, in particular for the fragile persons in rural areas, where frequently, they do not come for their medical follow up because of the distance, tiredness and transportation costs.
- Thus, the medical follow-up is the real need, since in most of the time persons do not need to be physically present in front of the doctor, and where contextual data can help provide better diagnosis and remotely.

Moreover, from a technical point of view:

- In the hospitals, monitoring sensors exists, and widely used. However, they are mainly used just in the range of the hospital, with a very limited transmission range, and they are mainly embedded sensors attached directly to the body of the patient, such as holters, pacemakers and pomps, and that they trigger alerts in case of an abnormality.

- Generally, practitioners do not trust the medical devices widely used in the market, but they trust the regulated ones.
- These devices can help the doctors providing better diagnosis, in case of remote consultations.

Finally, a data protection evaluation, regarding the privacy and the eventual risks, is conducted in order to push the final product to the market, which is mandatory in this kind of projects.

Chapter 7 | Conclusion & perspectives

Contents

In this chapter, we summarize the major contributions of this thesis together with a discussion of the future research directions.

7.1 Major contributions

In this thesis, we focused on the articulation between two different technologies, WebRTC which represents the new real-time and peer-to-peer communication technology, and a browser-based one, and the emerging Web of Things, which enables the abstraction of all the connectivity complexity of the Internet of Things, while aiming at having an interoperable and silo-free architecture for all the smart objects. Thus, the objective is to enhance the capabilities of WebRTC by the contextual data surrounding the communicating peers, which can be collected from the incrementally deployed sensors. These data can be then used to support the experts and allowing them to serve better expertises of a given situation. For instance, allowing an agriculture engineer to serve better diagnosis and advices to a farmer regarding the state of the soil of a farm, by collecting data from soil sensors and sending them securely in peer-to-peer and real-time through an ongoing WebRTC multimedia session between them. As a consequence, analyzing this issue guided us to provide three main contributions, which can be summarized as follows:

7.1.1 Coupling WebRTC and the Web of Things

In the first contribution, we focused on the previously mentioned issues, by providing first a set of use cases illustrating the need of such articulation. These use cases are mainly related to the telemedicine domain, and the need for such use cases has been supported by, first, the recent efforts of the French system to solve the issues related to the medical deserts using telemedicine services, and by several interviews with doctors.

Based on these use cases, an architecture linking these two technologies was implemented. The architecture, provides in addition to the multimedia capabilities of WebRTC, the ability to access the smart objects surrounding a user in a secure way, using authentication, confidentiality, integrity, and access control mechanisms. Additionally, and following the European regulation, the architecture need to be compliant with the privacy requirements set by the new GDPR law. Hence, a privacy and risk analysis was conducted on one of the telemedicine use cases, which is the tele-consultation one, in order to protect the personal sensitive data of the use, and to go toward the privacy by design notion. As an additional step, also an analysis of a business model regarding the tele-consultation use case, has been analyzed as a step to link the academic result to the market and to avoid keeping them on the shelf.

7.1.2 Toward next generation smart houses

Next generation smart homes require managing all the appliance and the facilities in a simple, intelligent, and efficient way, together with providing the users with wellbeing and comfort services. Hence, a better user experience. In this contribution, and through a collaboration with colleagues in Aalborg university, we tried to integrate the previously mentioned telemedicine services to the next generation smart homes. These services provide both wellbeing services, by allowing the users to monitor their vital signs at will using wearable medical sensors, and healthcare services, such as a remote consultation with a doctor or remote monitoring of the patient in his/her own home. Moreover, the work in this real smart home, also included the management of the different present heterogeneous sensors, and the efficient energy management. The energy part consists in monitoring and visualizing the energy, gas, and water consumption using smart meters and smart plugs together with the management of the renewable energy (PV panels and wind turbine). Furthermore, an important step of this collaboration, and contribution, consists in adding a security layer to the system, including authentication, confidentiality, integrity, access control and reliability.

7.1.3 Access to multiple smart spaces using SDN

In the previous contributions, we dealt just with the smart objects in the environment of the user, such as in the smart home. We called such environment a “smart space“. However, positioning our approach in the context of communication services operating in smart cities requires the ability to support a multiplicity of SSs, each with its own network and security policy. Hence, in order to allow a participant to access one of his own SSs or one of other participants, it becomes necessary to dynamically identify, select, deploy, and enforce the SSs specific routing and security rules, so as to have an effective, fast and secure access. Therefore, this contribution consists in defining an efficient management of the routing and security issues for multiple SSs, which can be distributed all over the network of an ISP. On the other side, Software Defined Networking (SDN) controllers proved their efficiency for managing, dynamically, large networks. And due to the fact that routing concerns are intimately intertwined with those of security, the SDN clearly appears as a promising tool to solve these issues.

Hence, the proposed solution consists in centralizing the decision concerning the management of the various SSs using the SDN controller together with third party security related entities. These entities interact with the northbound interface of the controller. We, therefore, developed an original architecture allowing the end-users to dynamically manage the access to their SSs in an effective and simple way. A prototype illustrating our approach was implemented together with a set of experiments to evaluate the performance and security of the system. This approach was finally illustrated in the e-Health domain by demonstrating the possibility of managing a health infrastructure such as a Hospital.

7.2 Perspectives for future work

Furthermore, the obtained results can be extended through short and long term future works.

7.2.1 Short term perspectives

It concerns several works directly linked to the proposed contributions of this thesis. In this section, they will be discussed and briefly analyzed.

7.2.1.1 Machine learning for telemedicine

In addition to the real-time and the multimedia capability that are proposed using the coupling of WebRTC and the WoT, we believe that adding machine learning and advanced algorithms based on the behavior of the patient can be also discussed as future work, in order to detect early syndromes of the well-known diseases such as early detection of diabetes, AIDS, Alzheimer, and so on. Moreover, adding a machine learning from the doctor side during a WebRTC session may add more possibilities and facilitate the task for the doctor in order to provide a better diagnosis to the remote patient. This machine learning may for instance analyze the data received from the medical devices attached to the body of the patient and provide a first diagnosis to the doctor based on health related ontologies.

7.2.1.2 Other application domains for the first contribution

The strength of the architecture coupling WebRTC and the WoT, introduced in chapter 3, is that it can be used in several domains, and not only for the e-Health. For instance, another application domain can be for the Intelligent Transportation Systems (ITS), and in particular for the connected cars. Where the main idea is to take advantage of the sensors located in a smart vehicle in order to send information to a remote entity (i.e. an emergency center) via WebRTC. For instance, in case of detection of a serious crash, the car will automatically call an emergency center for help. The car will try to access some of the still functional sensors (cameras, GPS, and so on) and then automatically establish a WebRTC session with the emergency center. Furthermore, this proposal can be a complementary to the mechanisms proposed in the eCall project [153], which is a European initiative to bring rapid assistance to motorists involved in a collision anywhere in the European Union.

7.2.1.3 Using an Open Source SDN controller and improving the performances

In the current implementation of the SDN solution to manage multiple smart space, we used a closed source SDN controller together with external security blocks, due to the limitation imposed by the controller. Moreover, the performance evaluations showed important issues regarding the single point of failure and the scalability. Hence, we argue that first by using a more developed and well known open source SDN controller, we may be able to integrate the security blocks to the core of the controller, and hence, avoiding the use of third party entities, having better control over the security of the network and having better performances since at least we avoid several interactions with the third parties. Additionally, using redundancy together with load balancing and shared memories allow to tackle the single point of failure issue, and can provide even better scalability, which may be beneficial for the management of the smart spaces. Open source controllers such as OpenDayLight, Floodlight and ONOS are the favorite candidates for these tasks due to their huge communities.

7.2.1.4 Privacy analysis regarding the SDN

In the first contribution, a privacy analysis was conducted in order to identify the privacy risks regarding the user's personal data, and to try to mitigate them. Similar work need to be done for the third contribution, regarding the use of SDN in order to manage the access to the smart spaces of the users. Since the architecture manages personal data, mainly collected using the smart objects in the smart spaces, analyzing the privacy of such architecture becomes a must under the new European GDPR law. Moreover, it becomes more serious when applied to the health domain in order to manage critical infrastructures such as the hospitals. We also argue that similar privacy analysis methodology can be applied to this architecture.

7.2.2 Long term perspectives

7.2.2.1 Smart contracts using Block chains

One major drawback regarding the conducted business model of the tele-consultation use case is the guarantee that the patients pay their consultations and that the doctors receives them. Moreover, and in particular in France, the social security needs also to be in the loop of the transactions between the health professionals and their patients. Hence, a possible solution for such issue is the use of Block chain in order to create smart contracts between the patient and the health professional. Thus, allowing the performance of credible, trackable and irreversible transactions without third parties. However, the analysis of such solution requires, first, better regulations regarding the use of smart medical objects for healthcare in general, and for telemedicine in particular. Secondly, it requires a good understanding of the health system in the targeted country, and in particular regarding the insurance part. And finally, a good knowledge regarding the use of Block chain and smart contracts.

7.2.2.2 5G: the healthcare slice

The current legacy and traditional infrastructures are attached to their traditional methods of healthcare, which is the main obstacle of the healthcare organizations toward the new connected world, with an exponential growth of connected devices, especially the health and wellbeing related ones. Hence, a world going toward an era of softwarization in healthcare. Thus, the main challenges facing the current healthcare industry are mainly related to data management, the lack of IT infrastructure, low operating margins, unbalanced distribution of medical resources, and the IoT related medical devices. For this reason, healthcare organizations need to adapt advanced and new techniques to address these challenges. The new vision of 5G technology can assist healthcare service providers to improve these processes further, and telcos may have a role to play in this value chain. Additionally, the current trend is to exploit the new network capabilities and performances enabled by 5G at the application layer, and in particular with the Web application such as WebRTC, and with the IoT world in order to support many more devices with very low latency than previous wireless networks. Moreover, coupling the 5G with the SDN and NFV seems to be a very promising solution toward the next generation network, in particular for the telco operators. Hence, our previous works seem to be at the heart of these new trends, in particular regarding the telemedicine services using WebRTC and the WoT, and the use of the SDN controller to control the access to the different smart spaces.

Bibliography

[1] K. Kajimoto and al., "Web of things (wot) architecture," *Unofficial IETF Draft*, June 2018, last visited: 16 Sept 2018. Copyright, 2017-2018 W3C (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and permissive document license rules apply. Status: "Draft". http://www.w3.org/Consortium/Legal/2015/doc-license. [Online]. Available: https://w3c.github.io/wot-architecture

[2] F. F. C. C. NGNI, "OpenSDNCore – Research and Testbed for the carrier-grade NFV/SDN environment." [Online]. Available: https://www.opensdncore.org/

[3] J. Granjal, E.Monteiro, and J. S. Silva, "Application-layer security for the wot: Extending coap to support end-to-end message security for internet-integrated sensing applications," in *Wired/Wireless Internet Communication - 11th International Conference, WWIC, St. Petersburg, Russia, June 5-7. Proceedings*, 2013, pp. 140–153.

[4] S. Gerdes, L. Seitz, G. Selander, and C. Bormann, "An architecture for authorization in constrained environments," *IETF, Internet-Draft*, Published: 2015 (Expires: September 2, 2016).

[5] S. Lucero *et al.*, "Iot platforms: enabling the internet of things," *White paper*, 2016.

[6] J. Mander, "Understanding the mobile-only internet user," in *Trends 17, the trends to watch in 2017, Globalwebindex*, 2017.

[7] H. Mshali, T. Lemlouma, M. Moloney, and D. Magoni, "A survey on health monitoring systems for health smart homes," *International Journal of Industrial Ergonomics*, vol. 66, pp. 26 – 56, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169814117300082

[8] M. Vega-Barbas, D. Casado-Mansilla, M. A. Valero, D. L. de Ipiña, J. Bravo, and F. Flórez, "Smart spaces and smart objects interoperability architecture (s3oia)," in *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, July 2012, pp. 725–730.

[9] U. DeSA *et al.*, "World population prospects: the 2017 revision," *Population division of the department of economic and social affairs of the United Nations Secretariat, New York*, 2017, "last visited: 28 juin 2018". [Online]. Available: "https://esa.un.org/unpd/wpp/publications/Files/WPP2017_KeyFindings.pdf"

[10] "EIT digital Doctoral School program." [Online]. Available: https://www.eitdigital.eu/eit-digital-academy/doctoral-school/

[11] S. E. Jaouhari, A. Bouabdallah, and J.-M. Bonnin, "La sécurité des objets connectés," in *MISC: multi-system & internet security cookbook*, no. 88, November 2016, pp. 54–59.

[12] ——, "Chapter 14 - security issues of the web of things," in *Managing the Web of Things*, Q. Z. Sheng, Y. Qin, L. Yao, and B. Benatallah, Eds. Boston: Morgan Kaufmann, 2017, pp. 389 – 424. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128097649000184

[13] A. Sergiienko, *WebRTC Blueprints*. Packt Publishing Ltd, 2014.

[14] R. Raymond, B. Aboaba, and J. Uberti, "Object rtc (ortc) api for webrtc," *W3C ORTC Community Group, W3C Community Draft, Dec,*, 2016.

[15] S. Loreto and S. P. Romano, *Real-Time Communication with WebRTC: Peer-to-Peer in the Browser*. " O'Reilly Media, Inc.", 2014.

[16] S. Nandakumar and C. Jennings, "Annotated Example SDP for WebRTC," Internet Engineering Task Force, 2018.

[17] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550 (INTERNET STANDARD), Internet Engineering Task Force, Jul. 2003, updated by RFCs 5506, 5761, 6051, 6222, 7022, 7160, 7164. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[18] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security," RFC 4347 (Proposed Standard), Internet Engineering Task Force, Apr. 2006, obsoleted by RFC 6347, updated by RFCs 5746, 7507. [Online]. Available: http://www.ietf.org/rfc/rfc4347.txt

[19] ——, "Datagram Transport Layer Security Version 1.2," RFC 6347 (Proposed Standard), Internet Engineering Task Force, Jan. 2012, updated by RFCs 7507, 7905. [Online]. Available: http://www.ietf.org/rfc/rfc6347.txt

[20] R. Stewart, "Stream Control Transmission Protocol," RFC 4960 (Proposed Standard), Internet Engineering Task Force, Sep. 2007, updated by RFCs 6096, 6335, 7053. [Online]. Available: http://www.ietf.org/rfc/rfc4960.txt

[21] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing, "Session Traversal Utilities for NAT (STUN)," RFC 5389 (Proposed Standard), Internet Engineering Task Force, Oct. 2008, updated by RFC 7350. [Online]. Available: http://www.ietf.org/rfc/rfc5389.txt

[22] R. Mahy, P. Matthews, and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)," RFC 5766 (Proposed Standard), Internet Engineering Task Force, Apr. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5766.txt

[23] J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," RFC 5245 (Proposed Standard), Internet Engineering Task Force, Apr. 2010, updated by RFC 6336. [Online]. Available: http://www.ietf.org/rfc/rfc5245.txt

[24] "WebRTC Architecture." [Online]. Available: https://webrtc.org/architecture

[25] E. Rescorla, "WebRTC security architecture," Internet Engineering Task Force, 2018.

[26] ——, "Security Considerations for WebRTC," Internet Engineering Task Force, 2018.

[27] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song, "Toward a standardized common m2m service layer platform: Introduction to onem2m," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 20–26, June 2014.

[28] A. Nettsträter and al., "Internet of things-architecture iot-a deliverable d1. 3–updated reference model for iot v1. 5," 2012.

[29] AIOTI, "High level architecture (hla), release 2.0, aioti wg03 – lot standardisation," 2015.

[30] D. Guinard and V. Trifa, *Building the Web of Things*. Manning Publications Co, 2016.

[31] W. H. Organization, *Global diffusion of eHealth-making universal health coverage achievable: report of the third global survey on eHealth, Geneva*. World Health Organization, 2016.

[32] ——, "Resolution wha58.33. sustainable health fnancing, universal coverage and social health insurance, in the fifty-eighth world health assembly, geneva," 2005, [last visited: 28 juin 2018]. [Online]. Available: http://www.who.int/healthacademy/media/WHA58-28-en.pdf

[33] W. H. Organization *et al.*, *National eHealth strategy toolkit*. International Telecommunication Union, 2012.

[34] M. Kay, J. Santos, and M. Takane, "mhealth: New horizons for health through mobile technologies," *World Health Organization*, vol. 64, no. 7, pp. 66–71, 2011.

[35] F. Magrabi and al., "Home telecare: system architecture to support chronic disease management," in *Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 4, 2001.

[36] C. Lau and al., "Asynchronous web-based patient-centered home telemedicine system," vol. 49, 12 2002.

[37] H. Zheng, R. J. Davies, and N. D. Black, "Web-based monitoring system for home-based rehabilitation with stroke patients," in *18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*, June 2005, pp. 419–424.

[38] C. Y. Chiang and al., "An efficient component-based framework for intelligent home-care system design with video and physiological monitoring machineries," in *Fifth International Conference on Genetic and Evolutionary Computing*, Aug 2011, pp. 33–36.

[39] P. Pierleoni and al., "An innovative webrtc solution for e-health services," in *IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, Sept 2016, pp. 1–6.

[40] M. A. Al-Taee and al., "Web-of-things inspired e-health platform for integrated diabetes care management," in *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, Dec 2013, pp. 1–6.

[41] J. a. Azevedo, R. L. Pereira, and P. Chainho, "An api proposal for integrating sensor data into web apps and webrtc," in *Proceedings of the 1st Workshop on All-Web Real-Time Systems*, ser. AWeS '15. New York, NY, USA: ACM, 2015, pp. 8:1–8:5. [Online]. Available: http://doi.acm.org/10.1145/2749215.2749221

[42] J. Jang-Jaccard and al., "Webrtc-based video conferencing service for telehealth," *Computing*, vol. 98, no. 1, pp. 169–193, 2016.

[43] H. Moustafa, E. M. Schooler, G. Shen, and S. Kamath, "Remote monitoring and medical devices control in ehealth," *IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–8, Oct 2016.

[44] T. H. Luan, L. Gao, Z. Li, Y. Xiang, and L. Sun, "Fog computing: Focusing on mobile users at the edge," *CoRR*, vol. abs/1502.01815, 2015. [Online]. Available: http://arxiv.org/abs/1502.01815

[45] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC '12. New York, NY, USA: ACM, 2012, pp. 13–16. [Online]. Available: http://doi.acm.org/10.1145/2342509.2342513

[46] P. Verma and S. K. Sood, "Fog assisted-iot enabled patient health monitoring in smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1789–1796, June 2018.

[47] P. Wang, S. Liu, F. Ye, and X. Chen, "A fog-based architecture and programming model for iot applications in the smart grid," *CoRR*, vol. abs/1804.01239, 2018. [Online]. Available: http://arxiv.org/abs/1804.01239

[48] J. Granjal, E. Monteiro, and J. S. Silva, "On the effectiveness of end-to-end security for internet-integrated sensing applications," in *2012 IEEE International Conference on Green Computing and Communications*, Nov 2012, pp. 87–93.

[49] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," RFC 7252 (Proposed Standard), Internet Engineering Task Force, Jun. 2014. [Online]. Available: http://www.ietf.org/rfc/rfc7252.txt

[50] S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt, "Lithe: Lightweight secure coap for the internet of things," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3711–3720, Oct 2013.

[51] S. El Jaouhari, A. Bouabdallah, J.-M. Bonnin, and T. Lemlouma, "Securing the communications in a wot/webrtc-based smart healthcare architecture," in *Second International Workshop on Mobile Technology for Healthcare (MT4H) in conjunction with the 11th International Conference on Frontier of Computer Science and Technology (FCST) (accepted)*, 2017.

[52] O. Connect, "Openid connect." [Online]. Available: http://openid.net/connect/

[53] C. Pfleeger, *Security in Computing*, ser. Prentice-Hall International editions. Prentice Hall PTR, 1997.

[54] B. W. Lampson, "Protection," *SIGOPS Oper. Syst. Rev.*, pp. 18–24, 1974.

[55] A. M. C. and B. Liskovs, "A decentralized model for information flow control," *ACM Symposium on Operating Systems Principles (SOSP)*, pp. 129–142, Oct. 1997.

[56] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, Feb 1996.

[57] M. A. Al-Kahtani and R. Sandhu, "A model for attribute-based user-role assignment," in *Proceedings. 18th Annual Computer Security Applications Conference.* IEEE, 2002, pp. 353–362.

[58] J. Liu, C. Liu, D. Jiao, and J. Chen, "The research of a multi-factor dynamic authorization model," in *IEEE Ninth International Conference on e-Business Engineering, Hangzhou, China*, 2012, pp. 201–205.

[59] D. F. Ferraiolo and D. R. Kuhn, "Role-based access control," 1992, pp. 554 – 563.

[60] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, Feb 1996.

[61] ——, "Role-based access control models," vol. 29, no. 2. IEEE, 1996, pp. 38–47.

[62] R. Yavatkar, D. Pendarakis, and R. Guerin, "A Framework for Policy-based Admission Control," RFC 2753 (Informational), Internet Engineering Task Force, Jan. 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2753.txt

[63] W. Jansen, "Inheritance properties of role hierarchies," in *21st National Information Systems Security Conference*, 1998, pp. 6–9.

[64] S. E. Jaouhari and A. Bouabdallah, "A privacy safeguard framework for a webrtc/wot-based healthcare architecture," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan*, vol. 01, July 2018, pp. 468–473.

[65] J. Mohammed, C. H. Lung, A. Ocneanu, A. Thakral, C. Jones, and A. Adler, "Internet of things: Remote patient monitoring using web services and cloud computing," in *2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*, Sept 2014, pp. 256–263.

[66] S. J. Wu, R. D. Chiang, S. H. Chang, and W. T. Chang, "An interactive telecare system enhanced with iot technology," *IEEE Pervasive Computing*, vol. 16, no. 3, pp. 62–69, 2017.

[67] F. Jimenez and R. Torres, "Building an iot-aware healthcare monitoring system," in *34th International Conference of the Chilean Computer Science Society (SCCC)*, Nov 2015.

[68] A. M. Ghosh, D. Halder, and S. K. A. Hossain, "Remote health monitoring system through iot," in *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, May 2016, pp. 921–926.

[69] M. S. D. Gupta, V. Patchava, and V. Menezes, "Healthcare based on iot using raspberry pi," in *International Conference on Green Computing and Internet of Things (ICGCIoT)*, Oct 2015.

[70] H. N. Saha, S. Auddy, S. Pal, S. Kumar, S. Pandey, R. Singh, A. K. Singh, P. Sharan, D. Ghosh, and S. Saha, "Health monitoring using internet of things (iot)," in *8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON)*, Aug 2017, pp. 69–73.

[71] M. S. Mahmud, H. Wang, A. M. Esfar-E-Alam, and H. Fang, "A wireless health monitoring system using mobile phone accessories," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2009–2018, Dec 2017.

[72] S. S. Al-Majeed, I. S. Al-Mejibli, and J. Karam, "Home telehealth by internet of things (iot)," in *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2015, pp. 609–613.

[73] A. A. Alghanim, S. M. M. Rahman, and M. A. Hossain, "Privacy analysis of smart city healthcare services," in *IEEE International Symposium on Multimedia (ISM)*, Dec 2017.

[74] S. E. Jaouhari, A. Bouabdallah, J. M. Bonnin, and T. Lemlouma, "Toward a smart health-care architecture using webrtc and wot (accepted paper)," *World-Cist'17 - 5th World Conference on Information Systems and Technologies, Porto Santo Island, Madeira, Portugal*, April 2017.

[75] S. El Jaouhari, A. Bouabdallah, and J.-M. Bonnin, "A secure webrtc/wot-based health-care architecture enhanced with access control," in *The 32nd International Conference on Information Networking (ICOIN) 2018, Chiang Mai, Thailand (accepted).*

[76] CNIL, "Privacy impact assessment (pia) methodology," Febrary 2018.

[77] "General data protection regulation (gdpr)," April 2016. [Online]. Available: https://www.eugdpr.org

[78] W. 29, "Opinion 8/2014 on the on recent developments on the internet of things," 2014.

[79] ASIP, "l'agence française de la santé numérique." [Online]. Available: esante.gouv.fr

[80] M. Protection, "A guide to medical records in ireland," *Medical Protection Society*, 2012.

[81] A. Act, "Health insurance portability and accountability act of 1996," *Public law*, vol. 104, p. 191, 1996.

[82] "Yeelight," 2018, https://xiaomi-mi.com/smart-lighting/; last visited: 24 November 2018. [Online]. Available: xiaomi-mi.com/smart-lighting/

[83] "Philips Hue," 2018, https://www2.meethue.com/; last visited: 24 November 2018. [Online]. Available: meethue.com

[84] "Nest," 2018, https://nest.com/; last visited: 24 November 2018. [Online]. Available: nest.com/fr/

[85] L. Pescosolido, R. Berta, L. Scalise, G. M. Revel, A. D. Gloria, and G. Orlandi, "An iot-inspired cloud-based web service architecture for e-health applications," in *2016 IEEE International Smart Cities Conference (ISC2)*, Sept 2016, pp. 1–4.

[86] N. Zhu, T. Diethe, M. Camplani, L. Tao, A. Burrows, N. Twomey, D. Kaleshi, M. Mirmehdi, P. Flach, and I. Craddock, "Bridging e-health and the internet of things: The sphere project," *IEEE Intelligent Systems*, vol. 30, no. 4, pp. 39–46, July 2015.

[87] A. M. Rahmani, N. K. Thanigaivelan, T. N. Gia, J. Granados, B. Negash, P. Liljeberg, and H. Tenhunen, "Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2015, pp. 826–834.

[88] N. M. Khoi, S. Saguna, K. Mitra, and C. Ahlund, "Irehmo: An efficient iot-based remote health monitoring system for smart regions," in *2015 17th International Conference on E-health Networking, Application Services (HealthCom)*, Oct 2015, pp. 563–568.

[89] P. Verma and S. K. Sood, "Fog assisted-iot enabled patient health monitoring in smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1789–1796, June 2018.

[90] M. S. Hossain and G. Muhammad, "Emotion-aware connected healthcare big data towards 5g," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2399–2406, Aug 2018.

[91] M. Killian, M. Zauner, and M. Kozek, "Comprehensive smart home energy management system using mixed-integer quadratic-programming," *Applied Energy*, vol. 222, pp. 662 – 672, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261918305282

[92] A. Najafi-Ghalelou, S. Nojavan, and K. Zare, "Robust thermal and electrical management of smart home using information gap decision theory," *Applied Thermal Engineering*, vol. 132, pp. 221 – 232, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359431117365493

[93] X. Fan, B. Qiu, Y. Liu, H. Zhu, and B. Han, "Energy visualization for smart home," *Energy Procedia*, vol. 105, pp. 2545 – 2548, 2017, 8th International Conference on Applied Energy, ICAE2016, 8-11 October 2016, Beijing, China. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1876610217307956

[94] A. Singaravelan and M. Kowsalya, "A novel minimum cost maximum power algorithm for future smart home energy management," *Journal of Advanced Research*, vol. 8, no. 6, pp. 731 – 741, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2090123217301054

[95] A. Jindal, N. Kumar, and M. Singh, "Internet of energy-based demand response management scheme for smart homes and phevs using svm," *Future Generation Computer Systems*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17320976

[96] H. Shareef, M. S. Ahmed, A. Mohamed, and E. A. Hassan, "Review on home energy management system considering demand responses, smart technologies, and intelligent controllers," *IEEE Access*, vol. 6, pp. 24 498–24 509, 2018.

[97] L. Yu, T. Jiang, and Y. Zou, "Online energy management for a sustainable smart home with an hvac load and random occupancy," *IEEE Transactions on Smart Grid*, pp. 1–1, 2018.

[98] N. Javaid, I. Ullah, M. Akbar, Z. Iqbal, F. A. Khan, N. Alrajeh, and M. S. Alabed, "An intelligent load management system with renewable energy integration for smart homes," *IEEE Access*, vol. 5, pp. 13 587–13 600, 2017.

[99] A. Basit, G. A. S. Sidhu, A. Mahmood, and F. Gao, "Efficient and autonomous energy management techniques for the future smart homes," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 917–926, March 2017.

[100] W. Li, T. Logenthiran, V. Phan, and W. L. Woo, "Implemented iot-based self-learning home management system (shms) for singapore," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2212–2219, June 2018.

[101] E. J. Palacios-Garcia, E. Rodriguez-Diaz, A. Anvari-Moghaddam, M. Savaghebi, J. C. Vasquez, J. M. Guerrero, and A. Moreno-Munoz, "Using smart meters data for energy management operations and power quality monitoring in a microgrid,"

in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, June 2017, pp. 1725–1731.

[102] E. Rodriguez-Diaz, E. J. Palacios-García, M. Savaghebi, J. C. Vasquez, J. M. Guerrero, and A. Moreno-Munoz, "Advanced smart metering infrastructure for future smart homes," in *2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Sept 2015, pp. 29–31.

[103] F. J. Bellido-outeirino, J. M. Flores-arias, M. Linan-Reyes, E. J. Palacios-garcia, and J. J. Luna-rodriguez, "Wireless sensor network and stochastic models for household power management," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 3, pp. 483–491, August 2013.

[104] E. J. Palacios-Garcia, J. M. Flores-Arias, A. Chen, F. J. Quiles-Latorre, and F. J. Bellido-Outeirino, "Home energy management system based on daily demand prediction and zigbee network," in *2015 IEEE International Conference on Consumer Electronics (ICCE)*, Jan 2015, pp. 315–316.

[105] F. J. B. Outeiriño, J. F. Arias, M. Liñán-Reyes, and E. Palacios-Garcia, "In-home power management system based on wsn," in *2013 IEEE International Conference on Consumer Electronics (ICCE)*, Jan 2013, pp. 546–547.

[106] C. Lévy-Bencheton, E. Darra, G. Tétu, G. Dufay, and M. Alattar, "Security and resilience of smart home environments good practices and recommendations," *The European Union Agency for Network and Information Security (ENISA)*, 2015.

[107] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, March 2017, pp. 618–623.

[108] M. Al-Shaboti, I. Welch, A. Chen, and M. A. Mahmood, "Towards secure smart home iot: Manufacturer and user network access control framework," in *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, May 2018, pp. 892–899.

[109] S. Bandara, T. Yashiro, N. Koshizuka, and K. Sakamura, "Access control framework for api-enabled devices in smart buildings," in *2016 22nd Asia-Pacific Conference on Communications (APCC)*, Aug 2016, pp. 210–217.

[110] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo, "Design of secure user authenticated key management protocol for generic iot networks," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 269–282, Feb 2018.

[111] Y.-P. Kim, S. Yoo, and C. Yoo, "Daot: Dynamic and energy-aware authentication for smart home appliances in internet of things," in *2015 IEEE International Conference on Consumer Electronics (ICCE)*, Jan 2015, pp. 196–197.

[112] Y. Ashibani, D. Kauling, and Q. H. Mahmoud, "A context-aware authentication service for smart homes," in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2017, pp. 588–589.

[113] H. Ren, Y. Song, S. Yang, and F. Situ, "Secure smart home: A voiceprint and internet based authentication system for remote accessing," in *2016 11th International Conference on Computer Science Education (ICCSE)*, Aug 2016, pp. 247–251.

[114] B. Ahlgren, M. Hidell, and E. C. H. Ngai, "Internet of things for smart cities: Interoperability and open data," *IEEE Internet Computing*, vol. 20, no. 6, pp. 52–56, Nov 2016.

[115] J. Schaad, "Use of the Advanced Encryption Standard (AES) Encryption Algorithm in Cryptographic Message Syntax (CMS)," RFC 3565 (Proposed Standard), Internet Engineering Task Force, Jul. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3565.txt

[116] "Passport." [Online]. Available: http://www.passportjs.org/

[117] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, "A survey on activity detection and classification using wearable sensors," *IEEE Sensors Journal*, vol. 17, no. 2, pp. 386–403, Jan 2017.

[118] A. Wang, G. Chen, J. Yang, S. Zhao, and C. Y. Chang, "A comparative study on human activity recognition using inertial sensors in a smartphone," *IEEE Sensors Journal*, vol. 16, no. 11, pp. 4566–4578, June 2016.

[119] N. Chirakanphaisarn, T. Thongkanluang, and Y. Chiwpreechar, "Heart rate measurement and electrical pulse signal analysis for subjects span of 20–80 years," *Journal of Electrical Systems and Information Technology*, vol. 5, no. 1, pp. 112 – 120, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2314717216300265

[120] A. Anvari-Moghaddam, J. M. Guerrero, J. C. Vasquez, H. Monsef, and A. Rahimi-Kian, "Efficient energy management for a grid-tied residential microgrid," *IET Generation, Transmission Distribution*, vol. 11, no. 11, pp. 2752–2761, 2017.

[121] A. Anvari-Moghaddam, H. Monsef, and A. Rahimi-Kian, "Optimal smart home energy management considering energy saving and a comfortable lifestyle," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 324–332, Jan 2015.

[122] ——, "Cost-effective and comfort-aware residential energy management under different pricing schemes and weather conditions," *Energy and Buildings*, vol. 86, pp. 782 – 793, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778814008573

[123] "Mongodb: Replication." [Online]. Available: https://docs.mongodb.com/manual/replication/

[124] N. Feamster, J. Rexford, and E. Zegura, "The road to sdn: An intellectual history of programmable networks," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 87–98, Apr. 2014. [Online]. Available: http://doi.acm.org/10.1145/2602204.2602219

[125] Y. Xu and Y. Liu, "Ddos attack detection under sdn context," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.

[126] K. L. A. Lim, "Computer security event management system," Aug. 14 2012, uS Patent 8,245,297.

[127] M. Boussard, D. T. Bui, L. Ciavaglia, R. Douville, M. L. Pallec, N. L. Sauze, L. Noirie, S. Papillon, P. Peloso, and F. Santoro, "Software-defined lans for interconnected smart environment," in *2015 27th International Teletraffic Congress*, Sept 2015, pp. 219–227.

[128] Q. Xiaofeng, L. Wenmao, G. Teng, H. Xinxin, W. Xutao, and C. Pengcheng, "Wot/sdn : web of things architecture using sdn," *China Communications*, vol. 12, no. 11, pp. 1–11, November 2015.

[129] O. Flauzac, C. González, A. Hachani, and F. Nolot, "Sdn based architecture for iot and improvement of the security," March 2015, pp. 688–693.

[130] P. Bull, R. Austin, E. Popov, M. Sharma, and R. Watson, "Flow based security for iot devices using an sdn gateway," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, Aug 2016, pp. 157–163.

[131] S. Chakrabarty and D. W. Engels, "A secure iot architecture for smart cities," in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2016, pp. 812–813.

[132] V. Sivaraman, H. H. Gharakheili, A. Vishwanath, R. Boreli, and O. Mehani, "Network-level security and privacy control for smart-home iot devices," in *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2015, pp. 163–167.

[133] M. A. Salahuddin, A. Al-Fuqaha, M. Guizani, K. Shuaib, and F. Sallabi, "Softwarization of internet of things infrastructure for secure and smart healthcare," *Computer*, vol. 50, no. 7, pp. 74–79, 2017.

[134] C. Thuemmler, J. Mueller, S. Covaci, T. Magedanz, S. de Panfilis, T. Jell, and A. Gavras, "Applying the software-to-data paradigm in next generation e-health hybrid clouds," in *2013 10th International Conference on Information Technology: New Generations*, April 2013, pp. 459–463.

[135] H. Silva and A. Neto, "A holistic sdn-capable session-plane tailored for efficient iomt smart surveillance applications," in *2016 IEEE Globecom Workshops (GC Wkshps)*, Dec 2016, pp. 1–6.

[136] M. A. Shayokh, A. Abeshu, G. B. Satrya, and M. A. Nugroho, "Efficient and secure data delivery in software defined wban for virtual hospital," in *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, Sept 2016, pp. 12–16.

[137] M. Suh, S. H. Park, B. Lee, and S. Yang, "Building firewall over the software-defined network controller," in *16th International Conference on Advanced Communication Technology*, Feb 2014, pp. 744–748.

[138] S. T. Yakasai and C. G. Guy, "Flowidentity: Software-defined network access control," in *2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, Nov 2015, pp. 115–120.

[139] J. Matias, J. Garay, A. Mendiola, N. Toledo, and E. Jacob, "Flownac: Flow-based network access control," in *2014 Third European Workshop on Software Defined Networks*, Sept 2014, pp. 79–84.

[140] S. Sciancalepore, G. Piro, D. Caldarola, G. Boggia, and G. Bianchi, "Oauth-iot: An access control framework for the internet of things based on open standards," in *2017 IEEE Symposium on Computers and Communications (ISCC)*, July 2017, pp. 676–681.

[141] A. Hesham, F. Sardis, S. Wong, T. Mahmoodi, and M. Tatipamula, "A simplified network access control design and implementation for m2m communication using sdn," in *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, March 2017, pp. 1–5.

[142] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519 (Proposed Standard), Internet Engineering Task Force, May 2015, updated by RFC 7797. [Online]. Available: http://www.ietf.org/rfc/rfc7519.txt

[143] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[144] J. H. H. H. J. Heyman, C. Byström, "Locust: An open source load testing tool." [Online]. Available: https://locust.io/

[145] "OWASP Zed Attack Proxy Project." [Online]. Available: https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

[146] L. Sidki, Y. Ben-Shimol, and A. Sadovski, "Fault tolerant mechanisms for sdn controllers," in *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov 2016, pp. 173–178.

[147] B. Fitzpatrick, "Distributed caching with memcached," *Linux J.*, vol. 2004, no. 124, pp. 5–, Aug. 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1012889.1012894

[148] Q. Yan and F. R. Yu, "Distributed denial of service attacks in software-defined networking with cloud computing," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 52–59, April 2015.

[149] S. M. Mousavi and M. St-Hilaire, "Early detection of ddos attacks against sdn controllers," in *2015 International Conference on Computing, Networking and Communications (ICNC)*, Feb 2015, pp. 77–81.

[150] "Project floodlight." [Online]. Available: http://www.projectfloodlight.org/

[151] G. D. P. Regulation, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46," *Official Journal of the European Union (OJ)*, vol. 59, no. 1-88, p. 294, 2016.

[152] H. Information and Q. A. (HIQA), "Guidance on privacy impact assessment in health and social care," December 2010.

[153] "ecall: Time saved = lives saved," 2016. [Online]. Available: https://ec.europa.eu/digital-single-market/en/ecall-time-saved-lives-saved

[154] G. Nibaldi, "Specification of a trusted computing base (tcb)," MITRE CORP BEDFORD MA, Tech. Rep., 1979.

[155] V. Beltran, E. Bertin, and N. Crespi, "User identity for webrtc services: A matter of trust," *IEEE Internet Computing*, vol. 18, no. 6, pp. 18–25, Nov 2014.

[156] I. T. Javed, K. Toumi, and N. Crespi, "Protectcall: Call protection based on user reputation," in *2017 IEEE Trustcom/BigDataSE/ICESS*, Aug 2017, pp. 660–667.

[157] M. Perumal, D. Wing, R. Ravindranath, T. Reddy, and M. Thomson, "Session Traversal Utilities for NAT (STUN) Usage for Consent Freshness," RFC 7675 (Proposed Standard), Internet Engineering Task Force, Oct. 2015. [Online]. Available: http://www.ietf.org/rfc/rfc7675.txt

[158] L.-S. Huang, E. Y. Chen, A. Barth, E. Rescorla, and C. Jackson, "Talking to yourself for fun and profit," *Proceedings of W2SP*, pp. 1–11, 2011.

[159] M. Hazhirpasand and M. Ghafari, "One leak is enough to expose them all," in *Engineering Secure Software and Systems*, M. Payer, A. Rashid, and J. M. Such, Eds. Cham: Springer International Publishing, 2018, pp. 61–76.

[160] S. Englehardt and A. Narayanan, "Online tracking: A 1-million-site measurement and analysis," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 1388–1401. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978313

[161] N. M. Al-Fannah, "One leak will sink a ship: Webrtc ip address leaks," in *2017 International Carnahan Conference on Security Technology (ICCST)*, Oct 2017, pp. 1–5.

[162] A. Reiter and A. Marsalek, "Webrtc: Your privacy is at risk," in *Proceedings of the Symposium on Applied Computing*, ser. SAC '17. New York, NY, USA: ACM, 2017, pp. 664–669. [Online]. Available: http://doi.acm.org/10.1145/3019612.3019844

[163] D. Wagner, B. Schneier *et al.*, "Analysis of the ssl 3.0 protocol," in *The Second USENIX Workshop on Electronic Commerce Proceedings*, vol. 1, no. 1, 1996, pp. 29–40.

[164] N. Mavrogiannopoulos, F. Vercauteren, V. Velichkov, and B. Preneel, "A cross-protocol attack on the tls protocol," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 62–72.

[165] Y. Tian, Y. C. Liu, A. Bhosale, L. S. Huang, P. Tague, and C. Jackson, "All your screens are belong to us: Attacks exploiting the html5 screen sharing api," in *2014 IEEE Symposium on Security and Privacy*, May 2014, pp. 34–48.

[166] C. Kruegel, "Internet security," *The Industrial Communication Technology Handbook*, 2005.

[167] C. Metz, "Aaa protocols: authentication, authorization, and accounting for the internet," *IEEE Internet Computing*, vol. 3, no. 6, pp. 75–79, Nov 1999.

[168] A. Skarmeta, J. L. Hernández-Ramos, and J. B. Bernabe, "A required security and privacy framework for smart objects," in *ITU Kaleidoscope: Trust in the Information Society, Barcelona,Spain, December 9-11*, Dec 2015, pp. 1–7.

[169] S. Lee, J. P. Jeong, and J. S. Park, "Dnsna: Dns name autoconfiguration for internet of things devices," in *18th International Conference on Advanced Communication Technology (ICACT), Pyeongchang, South Korea*, Jan 2016, pp. 410–416.

[170] Z. Yan, N. Kong, Y. Tian, and Y. J. Park, "A universal object name resolution scheme for iot," in *IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber,Physical and Social Computing*, Aug 2013, pp. 1120–1124.

[171] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280 (Proposed Standard), Internet Engineering Task Force, May 2008, updated by RFC 6818. [Online]. Available: http://www.ietf.org/rfc/rfc5280.txt

[172] W. Yeong, T. Howes, and S. Kille, "Lightweight Directory Access Protocol," RFC 1777 (Historic), Internet Engineering Task Force, Mar. 1995, obsoleted by RFC 3494. [Online]. Available: http://www.ietf.org/rfc/rfc1777.txt

[173] S. Sun, S. Reilly, L. Lannom, and J. Petrone, "Handle System Protocol (ver 2.1) Specification," RFC 3652 (Informational), Internet Engineering Task Force, Nov. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3652.txt

[174] A. Bassi and al., "Enabling things to talk," *Designing IoT Solutions With the IoT Architectural Reference Model*, pp. 163–211, 2013.

[175] R. Hummen and R. Moskowitz, "Hip diet exchange (dex)," *IETF, Internet-Draft*, Published: 2015 (Expires: September 22, 2016).

[176] D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig, and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)," RFC 5191 (Proposed Standard), Internet Engineering Task Force, May 2008, updated by RFC 5872. [Online]. Available: http://www.ietf.org/rfc/rfc5191.txt

[177] S. Pack and Y. Choi, "Pre-authenticated fast handoff in a public wireless lan based on ieee 802.1 x model," in *Mobile and Wireless Communications*. Springer, 2003, pp. 175–182.

[178] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "Extensible Authentication Protocol (EAP)," RFC 3748 (Proposed Standard), Internet Engineering Task Force, Jun. 2004, updated by RFCs 5247, 7057. [Online]. Available: http://www.ietf.org/rfc/rfc3748.txt

[179] J. L. H. Ramos, A. J. Jara, L. Marin, and A. F. Skarmeta-Gómez, "Dcapbac: Embedding authorization logic into smart things through ecc optimizations," *Int. J. Comput. Math.*, pp. 345–366, 2016.

[180] R. Sinnema and E. Wilde, "eXtensible Access Control Markup Language (XACML) XML Media Type," RFC 7061 (Informational), Internet Engineering Task Force, Nov. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc7061.txt

[181] S. Raza and al., "Lithe: Lightweight secure coap for the internet of things," *IEEE Sensors, Volume 13*, 2013.

[182] A. jit A.Chavan and M. K. Nighot, "Securing coap using enhanced dtls for the internet of things," *International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 12, December 2014*.

[183] X. Cao, D. Shila, Y. Cheng, Z. Yang, Y. Zhou, and J. Chen, "Ghost-in-zigbee: Energy depletion attack on zigbee based wireless networks," *IEEE Internet of Things Journal*, 2016.

[184] P. Pongle and G. Chavan, "A survey: Attacks on rpl and 6lowpan in iot," in *International Conference on Pervasive Computing (ICPC), Pune, India*, Jan 2015, pp. 1–6.

[185] "Internet of things research study 2015 report," *Hewlett Packard*, Nov 2015. [Online]. Available: http://www8.hp.com/h20195/V2/GetPDF.aspx/4AA5-4759ENW.pdf

[186] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-service detection in 6lowpan based internet of things," in *IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Lyon, France*, 2013, pp. 600–607.

[187] K. Sonar and H. Upadhyay, "A survey: Ddos attack on internet of things," *International Journal of Engineering Research and Development*, 2014.

[188] M. B. Shemaili, C. Y. Yeun, K. Mubarak, and M. J. Zemerly, "A new lightweight hybrid cryptographic algorithm for the internet of things," in *International Conference for Internet Technology And Secured Transactions, London United Kingdom*, Dec 2012, pp. 87–92.

[189] W. R. Company, "Security in the internet of things lessons from the past for the connected future," *White Paper*, 2015.

[190] E. Vasilomanolakis, J. Daubert, M. Luthra, V. Gazis, A. Wiesmaier, and P. Kikiras, "On the security and privacy of internet of things architectures and systems," in *International Workshop on Secure Internet of Things, SIoT, Vienna, Austria, September 21-25*, 2015, pp. 49–57.

[191] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, and M. Richardson, "A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs)," RFC 7416 (Informational), Internet Engineering Task Force, Jan. 2015. [Online]. Available: http://www.ietf.org/rfc/rfc7416.txt

[192] A. Cavoukian, "Operationalizing privacy by design: A guide to implementing strong privacy practices," *Information and Privacy Commissioner, Ontario, Canada*, 2012.

[193] A. Iliev and S. W. Smith, "Protecting client privacy with trusted computing at the server," *IEEE Security & Privacy*, vol. 3, no. 2, pp. 20–28, 2005.

[194] J. Audun, I. Roslan, and B. Colin, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[195] S. Cirani and al., "Iot-oas: An oauth-based authorization service architecture for secure services in iot scenarios," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 1224–1234, Feb 2015.

[196] A. Bhargav-Spantzel, J. Camenisch, T. Gross, and D. Sommer, "User centricity: A taxonomy and open issues," *Journal of Computer Security*, vol. 15, no. 5, pp. 493–527, 2007.

[197] D. van Thuan, P. Butkus, and D. van Thanh, "A user centric identity management for internet of things," in *International Conference on IT Convergence and Security (ICITCS), Beijing, China*. IEEE, Oct 2014.

[198] J. Granjal, E. Monteiro, and J. S. Silva, "On the feasibility of secure application-layer communications on the web of things," in *2012 IEEE 37th Conference on Local Computer Networks (LCN), Clearwater, Florida*, Oct 2012, pp. 228–231.

[199] V. Gupta, M. Wurm, Y. Zhu, M. Millard, S. Fung, N. Gura, H. Eberle, and S. C. Shantz, "Sizzle: A standards-based end-to-end security architecture for the embedded internet," *Pervasive and Mobile Computing*, vol. 1, no. 4, pp. 425–445, March 2005.

[200] W. Jung and al., "Ssl-based lightweight security of ip-based wireless sensor networks," in *23rd International Conference on Advanced Information Networking and Applications, AINA, Workshops Proceedings, Bradford, United Kingdom, May 26-29*, 2009, pp. 1112–1117.

[201] J. Jonsson and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1," RFC 3447 (Informational), Internet Engineering Task Force, Feb. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3447.txt

[202] A. Yegin and Z. Shelby, "Coap security options," *expired IETF Internet-Draft*, 2011.

[203] D. Hardt, "The OAuth 2.0 Authorization Framework," RFC 6749 (Proposed Standard), Internet Engineering Task Force, Oct. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6749.txt

[204] S. Gerdes, O. Bergmann, and C. Bormann, "Delegated authenticated authorization for constrained environments," in *IEEE 22nd International Conference on Network Protocols, Raleigh, North Carolina*, Oct 2014.

[205] S. Gerdes and al., "Delegated coap authentication and authorization framework (dcaf)," in *IETF, Internet-Draft*, Published: 2014 (Expires: April 21, 2016).

[206] E. Maler, D. Catalano, M. Machulak, and T. Hardjono, "User-managed access (uma) profile of oauth 2.0," *IETF, Internet-Draft*, Published: 2015 (Expires: July 29, 2016).

[207] S. Cirani and M. Picone, "Effective authorization for the web of things," in *IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, Italy*, Dec 2015, pp. 316–320.

[208] S. W. Oh and H. S. Kim, "Study on access permission control for the web of things," in *17th International Conference on Advanced Communication Technology, Seoul Korea*, 2015, pp. 574–580.

[209] S. W. OH and H. S. KIM, "Decentralized access permission control using resource-oriented architecture for the web of things," in *16th International Conference on Advanced Communication Technology (ICACT), Pyeongchang, South Korea*, 2014, pp. 749–753.

[210] E. Barka, S. S. Mathew, and Y. Atif, "Securing the web of things with role-based access control," in *Codes, Cryptology, and Information Security - First International Conference, C2SI , Rabat, Morocco, May 26-28, Proceedings - In Honor of Thierry Berger*, 2015, pp. 14–26.

[211] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed nist standard for role-based access control," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 3, pp. 224–274, Aug. 2001.

[212] T. Müldner, G. Leighton, and J. K. Miziołek, "Parameterized role-based access control policies for xml documents," *Information Security Journal: A Global Perspective*, vol. 18, no. 6, pp. 282–296, 2009.

Appendix

Chapter A | Achievements

A.1 List of contributions

International journal:

Journal 1. Ibrahim Tariq Javed, Rebecca Copeland, Noel Crespi, Marc Emmelmann, Ancuta Corici, Ahmed Bouabdallah, Tuo Zhang, Saad El Jaouhari, Felix Beierle, Sebastian Göndör, Axel Küpper, Kevin Corre, Jean-Michel Crom, Frank Oberle, Ingo Friese, Ana Caldeira, Gil Dias, Nuno Santos, Ricardo Chaves, Ricardo Lopes Pereira, "Cross-domain identity and discovery framework for web calling services" Annals of Telecommunications, volume 72, .459-468, Springer, published august 2017.

National journal:

National review 1. S. E. Jaouhari, A. Bouabdallah, and J.-M. Bonnin, "La sécurité des objets connectés", in MISC: multi-system & internet security cookbook, no. 88, November 2016, pp. 54–59.

Book chapter:

Book chapter 1. S. E. Jaouhari, A. Bouabdallah and J. M. Bonnin, "Chapter 14 - security issues of the web of things", in Managing the Web of Things, Q. Z. Sheng, Y. Qin, L. Yao, and B. Benatallah, Eds. Boston: Morgan Kaufmann, 2017, pp. 389 – 424.

IETF Draft:

IETF Draft 1. R Copeland, K Corre, I Friese, S El Jaouhari, "Requirements for trust and privacy in WebRTC peer-to-peer authentication", IETF draft September 2016.

International conferences:

Conference 1. El Jaouhari S., Bouabdallah A., Bonnin JM., Lemlouma T. (2017) Toward a Smart Health-Care Architecture Using WebRTC and WoT,. WorldCIST 2017, Porto Santo, Portugal, in Advances in Intelligent Systems and Computing, vol 571.

Conference 2. S. E. Jaouhari, A. Bouabdallah and J. M. Bonnin, "A secure WebRTC/WoT-based health-care architecture enhanced with access control," 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 2018, pp. 182-187.

Conference 3. S. El Jaouhari and A. Bouabdallah, "Dynamic security management of smart WoT infrastructures using SDN," 2018 IEEE 88th Vehicular Technology (VTC), presented, 27–30 August 2018 in Chicago, USA.

Conference 4. A. Bouabdallah and S. El Jaouhari, "Certified multimedia statement with WebRTC and microservices," 2016 Asia Pacific Conference on Multimedia and Broadcasting (APMediaCast), Bali, Indonesia, 2016, pp. 47-52.

International workshops:

Workshop 1. S. E. Jaouhari, A. Bouabdallah, J. M. Bonnin and T. Lemlouma, "Securing the Communications in a WoT/WebRTC-based Smart Healthcare Architecture," 2017, MT4H workshops 14th International Symposium on Pervasive Systems, Algorithms and Networks & 2017 11th International Conference on Frontier of Computer Science and Technology & 2017 Third International Symposium of Creative Computing (ISPAN-FCST-ISCC), Exeter, UK, 2017, pp. 403-408.

Workshop 2. S. El Jaouhari and A. Bouabdallah, "A Privacy Safeguard Framework for a WebRTC/WoT-Based Healthcare Architecture," The 13th IEEE International Workshop on Security,Trust, and Privacy for Software Applications (STPSA) in IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 2018, pp. 468-473.

Posters:

Poster 1. "Innovative usages of WebRTC for the Web of Things", Rescom summer school, 2016, Guidel, France.

Poster 2. "Application of WebRTC enhanced with the Web of Things for Healthcare services", Journée Objets Connectés de Santé – 5 October 2017 - Paris, France.

Chapter B | Security in WebRTC

Communication protocols, in particular the ones dedicated to the real time exchanges, are required to guarantee the security of the system and their users. Especially regarding the authentication, the integrity and the confidentiality.

During the development of WebRTC, the IETF and the W3C identified several security challenges, in particular:

- A well know issue in the web, and WebRTC is not an exception, is that the JavaScript application can be downloaded from any website, and most of the time without the consent of the user.
- The user must be able to control the access to his/her input-output devices (in particular the multimedia related ones such as the camera and the microphone).
- The user should not receive an important signaling traffic as long as he/she did not accept the incoming call.
- The confidentiality and the authentication must be guaranteed during all the communication.
- The private information of the user such as the identity, geographic location, and so on, must not be revealed without his agreement, (and also to provide the possibility of making anonymous calls).

Thus, the security of WebRTC articulates around five fundamental points [25][26]:

- The security depends on the level of trust granted/accorded to the browser and to the supplier of the WebRTC services.
- The JavaScript application must be executed inside the Sandbox of the browser.
- The whole communication must be encrypted in order to protect it from the external threats.
- Getting the explicit consent of the users is a requirement, and it becomes even an obligation with the new GDPR law.
- The authentication, the identity verification and the origin checks are also required.

Hence, following this logic, the security in WebRTC can be divided into four parts representing the main aspects to be secured in order to guarantee the security of the overall system, and which will be explained in details in the next sections:

1. The browser.
2. The web server providing the WebRTC application, or the service provider.
3. The signaling.
4. The security of the peer-to-peer exchanges.

B.1 Security of the browser

The security of the browser is critical for the global security of WebRTC, since all the WebRTC system is based on it. Hence, a strong WebRTC security, should be build over a strong security basis, thus the browser. The security of the browser is one of the main concerns of several researches and in particular for the organization and the companies maintaining these browsers. The security of the browser is out of the scope of the thesis, and the used browser is supposed to be is a secure one, and behaves as a Trusted Computing Base (TCB) [154] (TCB refers to a set of components offering a secure environment to the system, WebRTC in this case). The main goal is to get a maximum of users' authentication and a minimum of trust in the service providers (calling services).

In WebRTC the entities interacting with the system can be divided into two main categories:

1. The authenticated entities: which includes the calling services, with a verifiable origin (i.e. valid certificates, etc.), as well as the users whose origin can be verified cryptographically. However, the fact that an entity is authenticated, does not systematically means that it is a trusted one. Several models were developed for WebRTC in order to provide a level of trust in an entity, such as in [155, 156].
2. The non-authenticated entities: which represents the other entities, the behavior of which is supposed suspicious/malicious.

Hence, an authentication method must be implemented in WebRTC. The W3C and IETF standards requires in this case the use of identity providers, which will be discussed next.

B.1.1 Identity providers (IdP)

The Identity Provider (IdP) allows the verification of the identity of a given user. In the Fig. B.1, an example of the establishment of a WebRTC session including an identity verification via an IdP is shown. Prior the establishment of the WebRTC session, Alice and Bob are supposed to be already registered with their respective IdP. They, then, request an identity assertion, which affirms their identity to whom concerned.

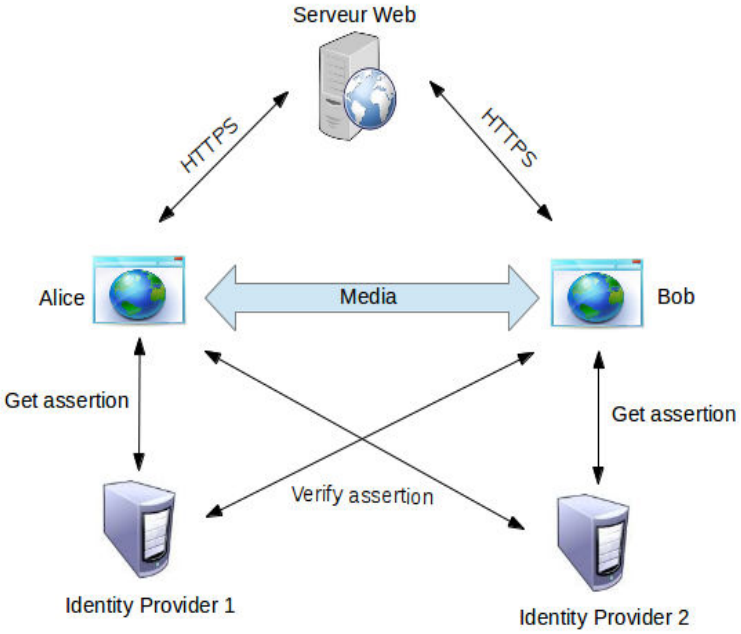


Figure B.1 – Identity verification with an identity providers (IdP)

When Alice wants to communicate with Bob, during the signaling process, one of the exchanged messages contains the identity assertion of Alice. Upon reception of this assertion by Bob, he contacts Alice's IdP (Identity Provider 1), to verify the received assertion, and in other words to verify the identity of Alice. An identical operation is done from Alice side when she receives Bob's assertion (with the Identity Provider 2).

Several well-known identity provider exists today such as Google, Facebook, etc. The Fig. B.2, summarizes the main exchanges between Alice and Bob in order to perform a mutual verification of their identity:

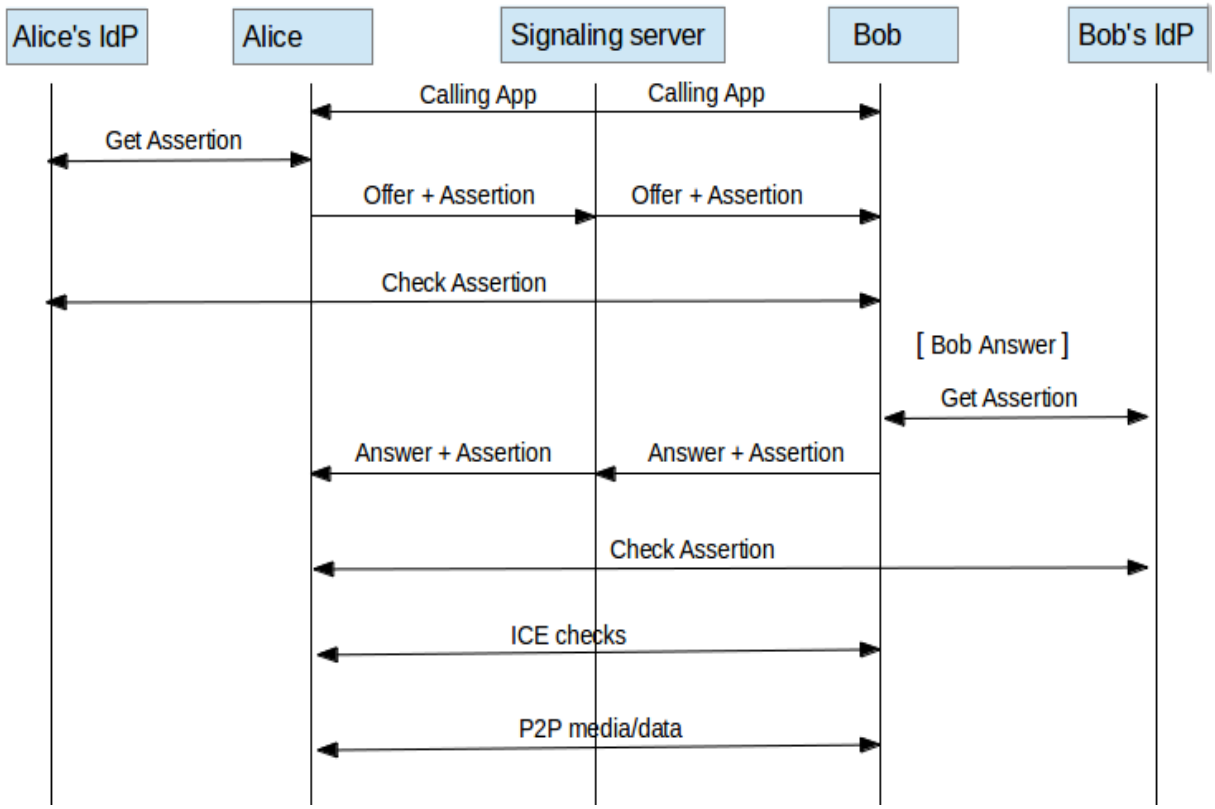


Figure B.2 – Prior authentication to a WebRTC session

B.1.2 Authorization

WebRTC requires the management of several types of authorizations in order to offer it services. The main authorizations concern: 1) the authorization to access the local multimedia equipments of the user such as the camera and the microphone, 2) the authorization to receive traffic from the remote communicating peer and also to continue receiving traffic during the call (which is also called consent freshness). These types of authorizations will be developed in the next paragraphs.

B.1.2.1 The authorization to access the local equipments of the user

In WebRTC, the access to the local equipments of the user is conditionally possible for the browser, but limited or forbidden for the web server. To obtain the access, the browser needs the explicit consent of the user via the MediaStream API. Without the user's agreement the establishment of the communication is not possible. Upon an access request to access an input/output device, a pop-up message appears in the browser asking the user to accept or to refuse the access. By default, WebRTC forbids arbitrary website from initializing calls without the explicit consent and acceptation of the concerned user.

Generally, there are two main models of permission to access these types of equipments, according to the reliability and the trust level in the calling service:

1. Short lifetime authorization: which is generally a unique access, usually applied to either a first access to the service, or to a non or less reliable service.

2. Long-lasting authorization: which is generally granted to the reliable and trusted services, the user can also give a permanent access to particular users such as family and friends.

Generally the permanent access is not recommended for the HTTP websites, but possible for the website with verified origins (HTTPS). Thus, the browser must show exactly the devices in use during all the multimedia session, using some indicators. If these ones are absent or masked, the browser must automatically stop the access and the sharing of these devices.

B.1.2.2 The authorization to receive traffic and consent freshness

Authorizing unlimited access to the network to a web application via the browser can introduce several risks. In particular, by using the browser as a platform to launch attack against other machines in the same network but not directly connected to the malicious site. The solution consists in obtaining explicit consent from the remote peer for receiving the traffic. This consent must be done during the establishment of the first WebRTC session by using the ICE Connectivity checks.

The ICE Connectivity checks is described as follow: once all the ICE candidate of Alice are gathered, ICE lists them by priority order and then sends them to Bob in the form of an SDP message in the SDP offer, which will be then sent through the signaling protocol. When Bob receives the SDP offer, he carries out an identical procedure and answers with his own ICE candidates. Then Alice and Bob create pairs with these candidates (couples of IP address and port number) and test their functioning, by exchanging STUN Binding request/response. These STUN messages are hashed using HMAC-SHA-1 in order to protect their integrity [21]. The result of these tests is then shared between both peers [23]. The Fig. B.3 summarizes in a simple way this exchanges:

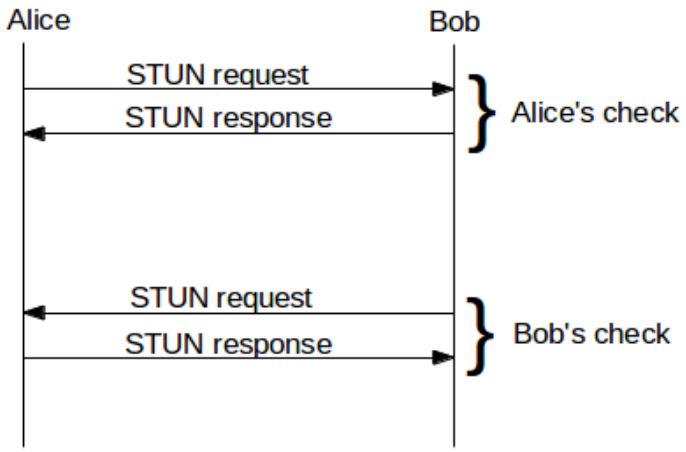


Figure B.3 – Elementary connectivity test using ICE

These connectivity tests are protected by a MAC (Authentication Code Message) using a key exchanged during the signaling process. At the end of this exchange, Alice and

Bob can then send/receive messages in a peer-to-peer fashion in both directions. Once the communication was established, it is necessary to verify that both participants are always connected. And this is done by periodically sending keep-alive messages. In case of failure of this test, and in the absence of another channel ICE (i.e. an other couple of ICE candidates), the session is terminated. WebRTC also uses STUN request/answer to guarantee the continuity of the connection (or the consent freshness). The sending and the reception of these messages is controlled by timers which is explained in [157], and which also describe the mechanisms to identify when the consent of the traffic ends, which can be either when the consent is expired or when the consent is revoked.

In the next section, another countermeasure needs to be taken into account regarding the confidentiality of the network localization of the users during a WebRTC call.

B.1.3 Confidentiality of the network localization

When two peers send their ICE candidates, the respective signaling servers establishing the communication can retrieve some information concerning the communicating peers, and in particular the network localization, which can raise some privacy issues.

As recommended in [26], in order to protect the privacy of the IP address of the remote peer or the callee, and to avoid tracking, the implementation should suppress the start of the ICE negotiation until reception of the explicit consent of the remote peer accepting the call. Thus, the identity of remote peer will be revealed only after the acceptance of the call. Both participants can also wish to mask completely their network location by forcing the routing of the traffic through a TURN server. In both cases the calling services, which can be either malicious or not, can determine the IP address of the browser. However, currently the only possible solution is by disabling the WebRTC by default in the browser.

Finally, precautions against malicious peers needs to be considered, which will be briefly presented next.

B.1.4 Malicious peers

Whatever is the confidentiality level used on WebRTC, it is always difficult to be protected against a malicious peer, in particular with the one with whom the exchange of the multimedia/data was accepted (accepted call). In this case, the malicious peer participating in the call can easily record the call for malicious reason. Also WebRTC can not forbid the peers from changing their profiles, to make false calls or to make anonymous calls. Hence, it is up to the user to judge the call and to give his authorization. The authentication mechanisms can help the use, to take the decision.

Next, the security mechanisms together with the possible attacks related to the services providers are provided.

B.2 Security of the service provider

B.2.1 Same Origin Policy (SOP)

An origin is defined by the triplet: protocol, host and port number. Same Origin Policy (SOP) restricts the way with which a document or a script loaded from an origin, can interact with another resource loaded from another origin. Thus, a browser will allow a script from a page A to reach the data of a page B, if and only if both pages have the same origin. SOP forces the scripts of each website to be executed in their own SandBox, and also prevents a server A from setting an attack against a server B from another origin via the browser of the user.

Several techniques exist allowing the interaction between scripts from various origins. They mainly lean on the consent of websites and by limiting the interactions as far as possible to only the authorized ones. Among the techniques allowing the bypass of the SOP policies:

- Cross-origin Resource sharing (CORS): which allows a script A to use the resources of a script B by contacting the target (the origin of B), in order to obtain its consent.
- WebSocket: once the connection is established between the site and the script, any further exchanges are authorized. It is advisable to subject the opening of the WebSocket connection to an authorization request.

However, controlling the access permission to the user's resources using only SOP can lead to some issues in WeBRTC. For example, if Alice gives her authorization for a website of origin X, then every time Alice visits this website, either voluntarily or through Ads, this site can reach in Alice's resources and initiate calls without her authorization. To solve this issue, there are three countermeasures based on the user's authorization:

- One time permission: where for each call, the users has to give an explicit agreement/consent to initiate the call. It is more secure, however, it can also decrease the user quality of experience, since it has to be done for each call.
- Authorized call for certain trusted persons: this solution can raise some issues, if for instance, the identity of one of authorized users, is stolen.
- Cryptographic authorization: authorization based on a prior authentication of the user, or the calling peer.

B.2.2 Mixed content

Since the authorization is based mainly on the origin, the reliability of the latter must be guaranteed by the transport protocol, HTTP or HTTPS. Hence, the user should give consent only to the websites with verified origins (i.e. the HTTPS ones with valid certificates). When an HTTPS website includes some HTTP parts, it can cause several security issues. An attacker can modify or read the HTTP parts, which may contain

sensitive data, and can also try to install malicious malware on the computer of the user, even if the origin of the website is verified (HTTPS). This issue is also called the "mixed contents".

Most of the current browsers, forbid the mixed contents in HTTPS websites. Consequently, WebRTC feature should not be accessible though pages with mixed content. Moreover, a browser allowing mixed contents, must imperatively deactivate WebRTC. Furthermore, it is possible that a page without mixed contents at the beginning of the call, integrates mixed contents during the call. Thus, a malicious JavaScript code could redirect the call towards the attacker for malicious purposes. The WebRTC implementation must in this case terminate the call and show a warning.

B.2.3 Attacks related to the service provider

Although in WebRTC the communication goes through a secure channel, it is possible that the service provider (also known as the calling server, the signaling server, or the web server) is malicious. It can either send malicious JavaScripts, which will then be executed in the browser of the user, or modify the signaling messages. Two categories of attacks can be distinguished, according to [26]:

1. Retrospective compromise of the calling service: the service in this case is not malicious during the call, but afterward it becomes compromised, in particular at the level of the encryption keys. The attacker can then cover the contents of the previously captured encrypted calls, with the hypothesis that the attacker can access to the medias and that he controls the service provider. This type of attacks is called "passive attacks". If the attacker has access to the encryption keys, then he can easily intercept all the communication. Moreover, if the key exchange is done in clear (HTTP), then it is very likely to compromise the call. As a countermeasure to these type of attacks, WebRTC has to use a key exchange mechanism guaranteeing the Perfect Forward Secrecy (PFS) (i.e. a cryptographic property which guarantees that the recovery of the session keys does not compromise the confidentiality of the previous communications).

2. During-call attack by the calling service: where the calling service can simply make a Man-In-The-Middle between the communicating peers. The solutions proposed against this type of attacks are:

    - Key continuity: a malicious calling service can present false identities to the user, but it cannot produce a private key compatible with the public key of the destination. The browser can, hence, detect this change and send a notification to the user every time this key is changed. The inconvenience of this solution is the fact that a user can have several pairs of (Private/Public) keys, if he/she uses server browsers in different computers, for instance.
    - Short Authentication String (SAS): is a version of the key agreement protocol, specially conceived so that a parameter can be compared by both parties of the communication. The confirmation of the correctness of the parameter by both peers protects against the Man-In-The-Middle attack.

- Third Party Identity: by using external trusted third parties (such as BrowserID, Federated Google Login, Facebook Connect,OAuth, OpenID, WebFinger), to prove the identity of the peers, is the most used solution in WebRTC.

B.3 Security of the signaling process

The signaling can be made in a secure way, since whatever is the used protocol, it will be encapsulated inside the WebSocket protocol. WebSocket leans on TCP, and allows the establishment of a bidirectional channel of communication between a client and the server. Thus, an HTTP website allows the establishment of a simple WebSocket channel, while an HTTPS website allows the establishment of a WSS (WebSocket Secured) channel.

B.4 Security of the peer-to-peer exchanges

After the ICE checks, both peers can establish a secure communication channel using DTLS. The WebRTC specification [26] shows that all the implementations of WebRTC should implement at least the version 1.0 of DTLS and preferably to support of the PFS. The DTLS handshake can be found in [18]. In this section, in addition to the DTLS mechanism, a brief explanation will be provided for the TURN authentication and integrity of messages mechanism, together with the main attacks and threats related to WebRTC.

B.4.1 Authentication using TURN

As explained in [21], STUN provides two mechanisms in order to authenticate the client and the server which are: 1) short-term credential mechanism, and 2) the long-term credential mechanism. However, the TURN server [22], which is based on the STUN specification, uses only the second mechanism. Nevertheless, the authentication process using long-term credentials is mandatory in TURN. The long-term credential mechanism relays on the credentials shared between the client and the server, which can be for instance the username and the password, as long as they were not compromised. Measures must be taken in this case against the replay attacks.

Hence, the TURN process, in addition to the authentication part becomes:

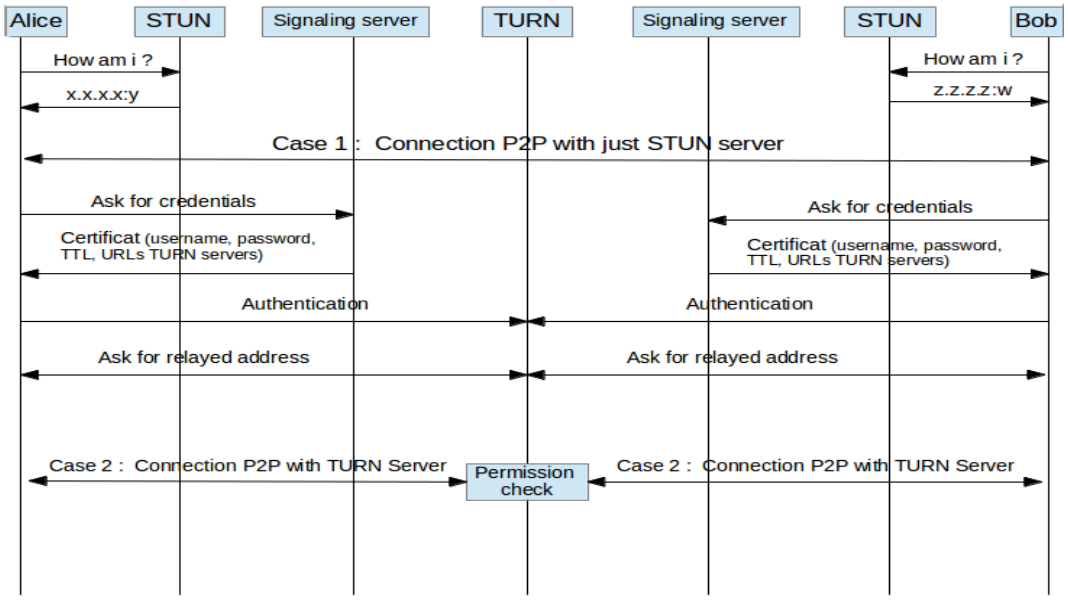


Figure B.4 – Authentication mechanism in TURN

In the Fig. B.4, a first case is explained, where we have only a simple STUN request to the STUN server, without any prior authentication. And a second use case, where, before sending a request to the TURN server to get the relayed addresses, a prior authentication is required. In this case, the authentication is based on long-term credential, which are the credentials already requested from the signaling server, more precisely from the IdP. Then, Bob and Alice can used the received credentials in order to authenticate to the TURN server and to request the relayed addresses. Finally, in order to establish the peer-to-peer channel using the relayed addresses of Alice and Bob, a permission check step is done by the TURN server.

B.4.2 Some possible attacks on WebRTC

A state of the art of the different types of attacks and the main countermeasures will be provided in this section, [26]:

- Man-In-The-Middle: as we discussed before, a malicious calling server can easily create this attack if the user gives his consent to the calling server. However, in order to be protected against this type of attacks, mechanisms including the authentication of the server and the remote peer, and the independent identification and verification of the keys, such us using IdP (section B.1.1), must be implemented.

- Downgrade attack: it is an attack where the attackers tries to lower the cryptographic level of an already encrypted connection. In this case, the use of DTLS allows the protection against this attack.

- Malicious JavaScript code: it is necessary to verify that the downloaded code does not have access rights to the private information of the users such as DTLS keys, ICE candidates and identities, passwords, etc., as well as the authorizations to manipulate them. Thus, the requirement of running them inside sandboxes.

- DoS attacks: two types were identified:
  - Attack 1: supposing having a calling server which accepts WebRTC calls, the attacker arranges so that a large number of users initiate a call towards the calling server. When a connection is established, the associated browser floods the calling server with a large amount of data. The identified countermeasure is to identify the suspicious behavior, using an Intrusion Detection System (IDS) for instance, and to be ready to close a communication channel in case there is a huge amount of data and in the absence of a reasonable audio/video flow.
  - Attack 2: the signaling server modifies the ICE candidates received by the user, and modifies the stream type field that the user wants to exchange. For instance to force the browser to send a video flow, while the victim's browser expects an audio only flow, which can overload the victim. The use of the RTP Control Protocol (RTCP) [17] messages allow countering this attack.

- Misinterpretation attack: even if the handshake of WebSocket is based on HTTP, the handshake uses the "Upgrade" mechanism to go from HTTP to WebSocket. The authors in [158] experimentally show that there are several proxies which do not implement the "Upgrade" mechanism in a good way, which sometimes means that even if the upgrade handshake is successful, the encapsulated traffic in the socket will be badly interpreted by the proxy.

- IP address leaks: [159], exploits the WebRTC IP leak, which can be triggered to fingerprint a web visitor, by scanning the user's private network ports and IP addresses from outside the local network using a web-based network scanner. The results show that, in addition to the identification of the real user identity behind a web request, an attacker can also retrieve some sensitive data related to the user's network infrastructure. Since, the origin of scanning is an internal code, it bypasses many rules on the firewall. As a countermeasure, a browser extension for Chrome and Firefox was proposed, by monitoring the network information and warns about requests related to the scanning or internal attacks to the nodes within the network. Many other works focus on WebRTC leaks in order to break the privacy of the users and to recover information related to the local IP address, such as in [160], which uses these information for tracking, and in [161], which analysis the different browsers and VPN in order determine which one are better for avoiding the leakage, and DDoS attacks in [162].

- Cross-protocol attack: where the main goal of an attacker is to position itself between the client and the server. In TLS, during the client/server exchange, the server protects only the cryptographic data related to the key exchange algorithm. The rest will not be protected, an active attacker can change the parameters of the TLS/SSL handshake, which can be used to mount the attack. As different algorithms for key exchange are possible in TLS/SSL, an attacker can deceive a client by altering the encrypting keys by the attacker keys, instead of the keys initially sent by the server. The attacker can then compute and recover the secret

key of the client, which will then compromise all the communication. There are two well known crosses-protocols attacks on TLS/SSL:

  - Attack 1: is an attacks where the malicious peer will try to deceive the client into thinking that the key exchanged used in that session is ephemeral RSA, but, in fact, the server will use Diffie-Hellman data instead. This attack is made by Wagner and Schneier, which explains a possible weakness in SSL 3.0. The attacker, in this case, will try to act as a server to the client and as a client to the server [163]. The main hypothesis is that the attacker, is an active one with the power to intercept, read and change all the exchanged messages between the client and the server.
  - Attack 2: is a similar to the previous one, where in this time, at the client side, the attacker will negotiate ephemeral Diffie-Hellman key exchange, and on the server side ephemeral Elliptic Curve Diffie-Hellman key exchange algorithm will be used. The goal, thus, remains similar, which is for the attacker to be a Man-in-the-Middle between the client and the server [164].

- Screen sharing: the visio-conference is the second most used use case in WebRTC, after the peer-to-peer communication. The participants in these visio-conferences usually need to share their screens or their applications with the other in the same session. However, guaranteeing the security of the audio/video communication is easier than security of the screen sharing, as explained in [26, 165]. Unfortunately, the security risks related to the screen sharing is mainly related to the lack of sensibility of the users regarding the security properties, for example:

  - The desktop often contains icons that the user forgot during the sharing, these icons can be for instance the names of confidential files.
  - Desktop notification: such as warning messages, which can be also confidential.
  - The user thinks that the sharing concerns only the windows of the communication, without paying attention the already open confidential files.

  These types of attacks is called “oversharing attacks”. Thus, an authorization policy is necessary in order to protect the users against this attack. Which includes: explicit consent for sharing with a clear presentation of the shared parts; a notification that the screen is currently shared; the non visible windows must not be shared etc.

B.5 Conclusion & Synthesis schema

In this section, a synthesis of all the previously mentioned mechanisms, together with the security related ones, is presented in the Fig. B.5.

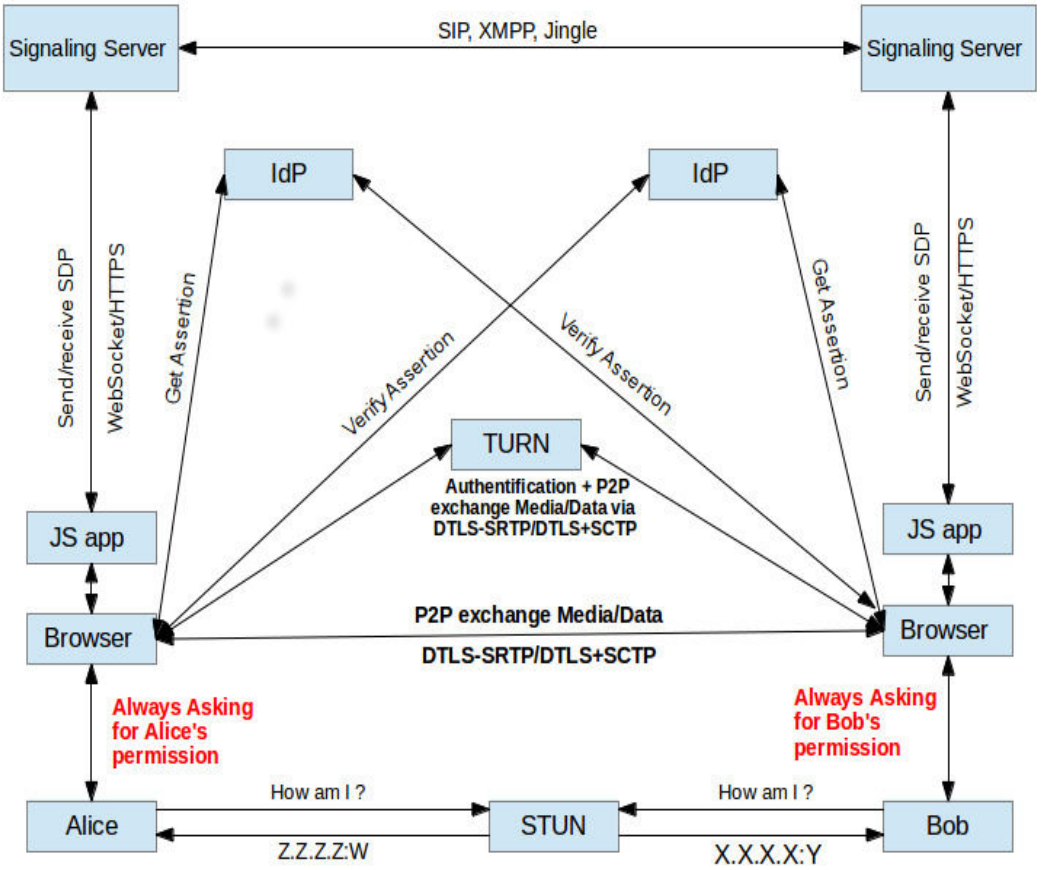


Figure B.5 – Complete example of the establishment of the WebRTC session

As a remainder, this figure can be divided into three main parts:

1. The Signaling part, where first the client (Alice and Bob) communicates with the signaling server, through their browsers, in order to get the JavaScript application with the WebRTC logic, and to start the signaling process. As mentioned before, the clients and the signaling server exchanges SDP messages, which contains the information related to the establishment of the session, securely transported using WSS/HTTPS. This part includes also the identity verification using the respective IdPs of the each user.

2. The NAT traversal: in order to bypass the problems related to the NAT, and in order to collect the necessary addresses to interconnect the users. The main mechanisms used are STUN, TURN and ICE.

3. The WebRTC secure peer-to-peer communication directly with the browsers, or though the relayed TURN addresses, using SRTP for multimedia and SCTP with DTLS for the random data.

Chapter C | Security in the Web of Things

The progress in the area of embedded systems has favored the emergence of so called "smart objects" or "Things". These ones incorporate, in a context of low energy consumption, various wireless communication capabilities combined with a microcontroller driving sensors and/or actuators. Smartphones, connected TVs, smart watches, and so on, are concrete examples of smart objects belonging to our everyday life. The Internet of Things (IoT) conceptualizes this new environment based on traditional networks connected with objects as specific components of the real world. Building however a global ecosystem gathering the different IoT environments, where Things can communicate seamlessly is a difficult task. Since each IoT platform uses its own stack of communication protocols, they usually are not able to work across the many available networking interfaces, which creates silos of users and Things. Web of Things (WoT) has the ambition to provide a single universal application layer protocol enabling the various Things to communicate with each other in a seamless way by using the standards and the APIs of the web as a universal platform. The articulation between objects and Internet if it represents a strong point of the WoT, leads also this one to inherit all the problems of security and privacy already present in Internet. These problems rest with stronger acuity in this new environment, because of its particular characteristics. It is therefore important to analyze the way in which traditional security and privacy requirements can be declined in this new environment. In this chapter, we will try to give a global overview of the currently proposed architectures for securing the WoT. This overview covers an analysis of the different threats and vulnerabilities that an IoT, eventually a WoT, architecture can be exposed to. It covers also the solutions proposed to solve the problematics related to the identity management, data confidentiality, the authorization and the access control in a WoT system.

In this chapter we will give a survey of the current advancements in this field. The chapter is structured as follows. Section C.1 introduces the main building blocks involved in the security of the Internet, provides a list of the main security properties, investigates the security of the smart objects during their life cycle and finally exposes the threats and the security requirements in the current IoT architecture. Section C.2 begins with an introduction about the security of the WoT, and lists the currently proposed security architectures in the WoT mainly the identity management, data confidentiality and integrity, the authorization and the access control.

C.1 The existing security models

The IoT is seen as the next "industrial revolution" by many experts, where billions of connected devices are able to exchange information between them and with other computing devices. Thence, these devices will constantly generate a huge amount of data, that will be exchanged via Internet. For this reason, we have first to go through some general security properties and threats to which systems are exposed when they are connected to the Internet. Naturally, since more and more data are exchanged over the Internet, they are susceptible to undergo different kinds of attacks including hijacking, eavesdropping, tampering, etc., where the attacker doesn't have to be physically present to take control over the device, a simple corrupted package can do the work. Indeed an eventual attacker can be present in any host connected to the Internet (including the device itself), and it becomes more challenging to prevent them [166].

In this section, we will first introduce the main security properties related to the Internet. The main interest of this subsection, is (1) to give a formal definition to some keywords that will be used all over this chapter; and (2) to have a brief idea of the main security properties required in the Internet in general, and in the IoT and the WoT in particular. Next we will take an interest in the SO itself. We will analyze the different security and privacy mechanisms deployed by the SO to be able to securely communicate with the outside world. And finally we will provide a general security analysis in IoT, including the different threats and vulnerabilities that IoT is exposed to and some security requirement that should be considered while building an IoT application.

C.1.1 Security properties

- Confidentiality: guarantees that the information exchanged by the users, will not be revealed to a third unauthorized party. A restricted access to the information must be applied and allows only the authorized ones to consult them.

- Integrity: guarantees that the information exchanged between two parties will not be altered or modified by unauthorized third party. It also involves maintaining the consistency, accuracy, and trustworthiness of data over its entire life cycle. It requires direct authorization by the owner of the data, and cryptographic mechanisms to check the integrity such as a hash function (e.g., SHA and MD5), checksums, etc.

- Availability: ensures that the data are always accessible by a legitimate user. Together with the confidentiality and the integrity they form the so called CIA triad, and they are the most crucial components of security.

- Authentication: is a way of identifying users, traditionally by requiring the user to provide valid credentials (e.g., username, password, etc.). The credentials provided are then compared to those already stored in the authentication server database. The credentials are proper to each user and must not be revealed to others. More elaborated authentication processes are required especially for

critical systems, for instance using biometric authentication (e.g., fingerprint, eye, etc.), or requiring strong authentication (i.e., authentication with more than one identity factor).

- Authorization: usually following a successful authentication process, users are granted authorization to execute certain task or getting certain information depending on the defined policies. Mainly, who can access the resource? what resource? what action can be performed and for how much time?

- Accounting/Auditing: Accounting consists in the measures of the quantity of resources consumed by a given user during his access, including the amount of system time and the amount of data exchanged. It merely concerns system management. It serves for collecting information in order to do statistics on resource utilization, authorization control and for economical purpose such as trend analysis and billing.

  Together with the authentication and the authorization they form the **AAA** [167], which essentially defines a framework for coordinating these individual disciplines across multiple networks, technologies and platforms. Hence, allowing an intelligent and efficient access control to computer resources, enforcing policies, auditing usage and providing the information necessary for services requiring billing. These combined processes are required for any effective security and network management.

- Non-repudiation: guarantees that the author of an information cannot deny its ownership. For example, a user who signed a document with his/her private key cannot deny his/her signature.

Before going deeper into the security mechanisms defined in IoT and WoT, we need to investigate first the critical aspects related to the security of the smart objects. The next section is dedicated to the different security and privacy properties that need to be preserved during the life-cycle of a smart object.

C.1.2 Security of smart objects

An attacker who has physical access to the smart object (SO) is able to gather lots of private information. Moreover, if the attacker succeeds to recover the private keys, he/she can decrypt all the traffic flowing from and into this SO, or he/she can inject some malicious codes to other endpoints. Hence, this is a very serious problem and also a critical point in the architecture of WoT and IoT.

In this subsection we will go through the different security and privacy properties that has to be preserved in order to secure the SO, by exploring an architecture proposed in [168]

First we will expose some of the important concepts that will be used in the architecture, some are related to the security and other are for discovery. We will begin with

the notion of identity in IoT, since a lot of definitions can be found in the literature, mainly about the identity and the partial identity of objects. In the IoT, SOs are considered as independent entities, with communication and computation capabilities, and with the ability to act on behalf of the user. Traditional identity management becomes obsolete and needs to be extended in order to deal with those changes. Identity will allow in one hand to distinguish the different objects inside the network, and in the other hand to verify their origins. Same as any identity management architecture, in order to create a trust environment, those identifiers must be unique.

Another kind of identity called partial identity can be also used to authenticate objects. Partial identity is mainly used for anonymity purpose. They contain a subset of attributes of a global identity, those attributes can either be chosen by the user or by the identity provider. Pseudonym is an example of partial identity. Those partial identities can be attributed to the users and eventually the objects depending on the situation and the context. Thus we can use either global identity (or identity in general) or partial identity to identify each of the objects according to the context and the environment.

Objects discovery is also one of the concerns in the current architectural research. Objects have to be addressable, named and also discovered. This is a very complex problem especially for the mobility, the availability and the constrained nature of the objects. Many proposals have been introduced in order to address such problems, in particular: IoT Addressing, IoT Naming and IoT Discovery [169] [170]. Those problems need a suitable infrastructure such as X.509 [171] or Lightweight Directory Access Protocol (LDAP) [172]. In the solution described below they prefer to use the Handle system (HS) [173], because of its advantages such as, simplicity, search capability, interoperability with the others systems, security features and the distributed administration and service model. Handle system is a distributed information system for secure global name service on the network. HS supports secure handled resolution, enabling storing names in a distributed manner, and guarantees access control, data confidentiality, integrity and non-repudiation [173].

Now we will go through the proposed security architecture. This document adds in fact a security layer to an European project called IoT-A focusing on the design of an Architectural Reference Model (ARM) [174], aiming at bringing interoperability between the different IoT domains. This proposal intends to extend the security functions in ARM, particularly the security and privacy in the different stages of the smart object life cycle.

In this architecture, the life cycle of a smart object is divided into three phases: the first one is the *Bootstrapping and Registration* phase where the smart object is installed, commissioned and connected to the network. The second phase is the *Discovery and Provisioning* process, where a smart object tries to access the resources of another one. And finally the *Operation* phase, where the smart object is able to communicate with the destination in a secure manner.

Every step in the previous life cycle needs to provide security and privacy of the users. The information and resources of the smart object must also be protected. Before going deeper into the security aspects of the different phases, a security hypothesis was assumed. Every smart object needs to be statically configured with a cryptographic material such as a X.509 certificate or equivalent, called *root identity*, in order to execute some security computation later in the different phases. Those materials could be provided either by the manufacturer or by its owner. Next we will inspect the security and privacy analysis of the different phases and also the proposed mechanisms.

The first step in the life cycle of a smart object is the *Bootstrapping*, where the smart object is installed and needs to be connected to the network. First the smart object needs to be authenticated and authorized before deployed. Naturally not every smart object is allowed to act in the network, otherwise malicious frameworks starting to deploy infected objects in the network may be problematic. This authentication and authorization can be performed using a lightweight protocol respecting the constraints related to the computation capabilities and energy consumption of the objects. Different protocols were mentioned such as Host Identity Protocol Diet EXchange (HIP-DEX) [175],Protocol for carrying Authentication for Network Access (PANA)[176], and 802.1X an IEEE standard for port-based network access control [177]. The proposed solution in [168] can use either PANA or EAP methods [178]. Once this operation done, the smart object can be registered and ready to be discovered, this is done using the resolution infrastructure Handle system discussed before. Additional privacy aspects can be done at this level, in case of successful authentication and authorization. For example, the smart object can compute other cryptographic materials, for anonymity purpose for instance by computing a partial identity. One more thing before going into the next step, is that the credentials are mainly exchanged using a Key Encapsulation Mechanism (usually a public key algorithm), in order to provide a symmetric key material, that will be used later to encrypt the future messages exchanges.

The second step is *Discovery and Provisioning* where the requester object wants to get resources from an other Object. Hence service discovery should check the authentication and the authorization of the requester object. And the final step is the *Operation*, where an Object A tries to communicate with an Object B. Figure C.1 shows the messages exchanged between the different objects to create a secure communication channel. It regroups the two previous steps:

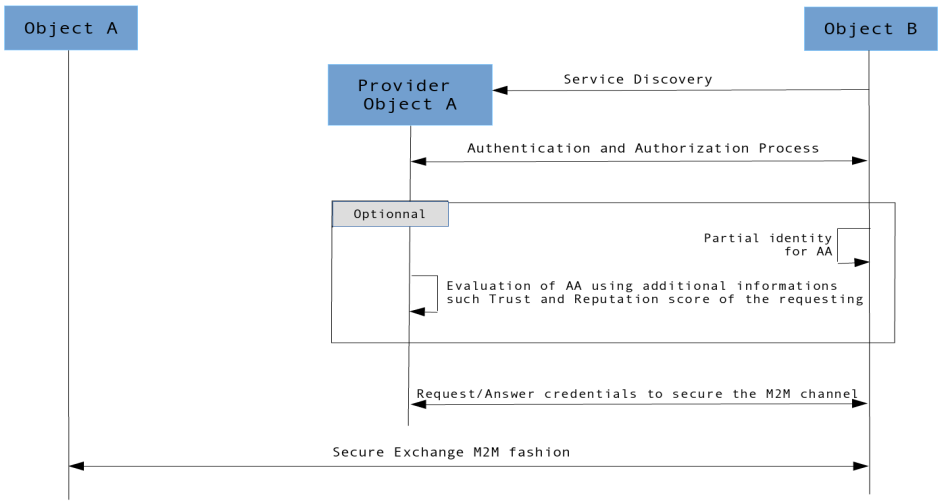


Figure C.1 – Discovery and Operation phases

As shown, first B has to discover if the provider has such kind of service. If it is the case, B will have to authenticate itself, either by providing its full identity, or using a partial identity for privacy-preserving purposes or anonymity. The selection of the identity is done according to the Object B policies and also to the contextual data. Once the authorization and the authentication are successfully done, B requests credentials from the provider to create a secure communication with A. The credentials are formated as a DCapBAC token [179]. The authentication, the authorization, and token exchanges parts can be done using PANA, and the exchanges between the object B and the provider can be secured using HTTPS or CoAP/DTLS [49]. As for the access control, it can be done via XACML technology [180]. Finally, with this DCapBAC token, a secure CoAP-DTLS [181, 182] channel can be created between the Object A and the Object B, hence providing a secure M2M communication.

Now that we analyzed the security of the smart object, in what follows we will analyze briefly some of security and privacy properties in the IoT. First we will go through the different threats and vulnerabilities that the IoT is exposed to, then the main security requirements for an IoT system, and finally we will present an example of an authorization framework.

C.1.3 The security in the Internet of Things

The huge number of smart objects that can be integrated into the Internet thanks to the IoT, and the number of users becoming more reliant on these interconnected devices, raises several security and privacy issues. In order to fully understand those issues, we structured this subsection as follows: (1) We will go through the main threats and vulnerabilities encountered in the IoT. To do so, we firstly present a risk analysis and the origins of those risks. And then, we present the most important attacks surface area, and the top ten IoT's vulnerabilities according to OWASP[1]. (2) We will present the main security requirements that should be fulfilled by an IoT application. (3) And we will conclude with an example of security model dedicated to the IoT.

[1]https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project

C.1.3.1 Threats and vulnerabilities

In order to secure an IoT application, several points need to be considered:

- Firstly, with the device itself, since those devices apply relatively weak security mechanisms due to their constrained nature, and also they can be either physically accessible or they can simply be malicious. Hence all these possibilities need to be treated, as explained in C.1.2.
- Then, the different communication protocols used to communicate with the smart devices. Several vulnerabilities appear (such as Ghost attack in ZigBee [183], usurpation, Sybil attack and Sinkhole attack in 6LoWPAN [184], etc), which can compromise the devices and the infrastructure.
- Next, the threats coming from the external entities. Attacks such eavesdropping, tampering, DoS attacks, phishing attacks or code injection attacks can occur.
- And finally, problems related to the privacy and trust from the device owner perspectives. Naturally private information needs to be confidential, protected and also guaranteed to be destined only to the legitimate person or device. Hence security and privacy requirements need to be set and applied to protect IoT.

Related works in this field considered the analysis of IoT's specific properties, to be able to analyze the security and privacy challenges, and this is the objective of this section.

A research study in HP [185] conducted on the security risks on the IoT devices, by viewing 10 of the most popular devices in some of the most common IoT architectures, shows that:

- 90 % of devices collected at least one piece of personal information via the device, the cloud, or its mobile application such as name, address, data of birth, health information, and even credit card numbers.
- Six out of 10 devices that provide user interfaces raised security concerns and were vulnerable to a range of issues such as persistent XSS and weak credentials.
- 70 % of devices do not encrypt communications to the Internet and local networks.
- 70 % of devices along with their cloud and mobile application enable an attacker to identify valid user accounts through account enumeration.
- 80 % of devices along with their cloud and mobile application components failed to require passwords of a sufficient complexity and length, hence weak passwords.

According to HP and OWASP those problems are mainly due to insufficient authentication and authorization, lack of transport encryption, insecure web interface and insecure software and firmware. Deeper analysis of those problems will be threatened in what follows. OWASP Internet of Things (IoT) Project [1], an Open Web Application Security Project explores and exposes the security risks associated with the IoT

in order to help developers, manufacturers or any entity of interest to understand and make better decision when dealing with IoT technologies. This project is structured into sub-projects mainly the IoT attack surface area and the top IoT vulnerabilities.

The IoT attack surface[2]:

The IoT attack surface is pretty wide and exposes an exhaustive list of attacks surfaces and their correspondent vulnerabilities. In this section, we will go through the most important ones which will give us the essential background to continue the exploration of the advancements in the field of IoT security researches:

- Ecosystem Access Control: This area focuses on the problems related to control the access to the device, mainly the problems related to the implicit trust between the components, the enrollment security and the lost access procedures.
- Device Web Interface: it covers all the web attacks that the device's web interface may be exposed to such as SQL injection attacks, Cross-site scripting, username enumeration and weak passwords etc.
- Device Network Services: covering the network attacks that may threat the device, such as Denial of Service, poorly implemented encryption, unencrypted services, buffer overflow, relay attacks, etc.
- Local Data Storage: since the resources in the device need to be protected, several vulnerabilities may occur mainly in case of unencrypted data or data encrypted with discovered keys, lack of data integrity checks or the use of same static encryption/decryption keys.
- Mobile Application: vulnerabilities such as lack of transport encryption, lack of two-factors authentication, weak passwords, etc.
- Network Traffic: covers all the problems related to the communication protocols specially the wireless ones (WiFi, Zigbee, Bluetooth) and the application of fuzzing protocols to test the IoT applications.
- Authentication/Authorization: one of the most important problems, not only in the IoT but also in the WoT. The vulnerabilities are mainly related to the authentication/authorization related values (credentials, session keys, tokens, etc.), device to device authentication, device to mobile application authentication and the lack of dynamic authentication.
- Privacy: user data disclosure, user/device disclosure and differential privacy are the main vulnerabilities that an IoT user may be exposed to.
- Hardware (Sensors): and last but not least all the problems related to the electronic devices itself such as sensing environment manipulation, tampering and damaging the device physically.

[2]https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Attack_Surface_Areas

Covering all the vulnerabilities provided by OWASP needs a lot of time. In the discussion that follows we will focus on the top ten vulnerabilities provided by OWASP. They are present in many attacks surface areas, and they form also the basis of the previous statistics made by HP.

IoT's top ten vulnerabilities[3]:

1. Insecure Web Interface: it's a threat that can be initiated either by an internal or an external attacker, exploiting the problems related to weak credentials or by the enumeration of users accounts. In term of exploitability and detectability, it is rated as EASY, which means that attacks of this type can be discovered just by manually examining the interface or by using automated testing tools. Also other issues can be identified using those tools such as cross-site scripting. The impact of an unsecured web interface may lead to data loss or corruption, denial of access and even complete device takeover. This is why it is rated as SEVERE regarding the impact.

2. Insufficient Authentication/Authorization : since access to the device's resource must be prohibited for unauthenticated or unauthorized entities, insufficient authentication or authorization mechanism is rated as SEVERE concerning the impact on the data and the device itself (data loss and corruption and even complete compromise of the device and/or the user accounts). The attacker can take advantage of the lack of granular access control and weak credentials, since authenticating entities with weak credentials may not be sufficient. In term of exploitability it is rated as AVERAGE and EASY in term of detectability.

3. Insecure Network Services: checking vulnerabilities related to open ports such as DoS, buffer overflow, and fuzzing attacks are very commune, since anyone who has access to the device via a network connection may attack the device, thus only the necessary ports need to be exposed and protected. Also, abnormal request traffic need to be blocked. Several DoS attacks have been proven to be efficient on the IoT devices [186, 187] specially when unsecured network services are available, rendering the device unavailable or inaccessible to the user. In term of exploitability and detectability it is rated as AVERAGE. However, the impact is rated as MODERATE.

4. Lack of Transport Encryption: eavesdropping and tampering can be easily set by an attacker in case of unencrypted data sent over the network, since it is prevalent that the traffic in the local networks is assumed to be not widely visible hence the lack of transport encryption. However, traffic can be visible in case of a mis-configured local wireless network which may result in data leak or loss. Many propositions for end-to-end encryption using DTLS or a lightweight cryptography have seen the light [188]. In term of exploitability it is rated as AVERAGE and EASY in term of detectability. The impact is rated as SEVERE.

[3]https://www.owasp.org/index.php/Top_IoT_Vulnerabilities

5. Privacy Concerns: protecting the user's private information is an important requirement in any system. In IoT case, the attacker can take advantage of a weak authentication, lack of transport encryption or unsecured network to gather personal non-protected information. This scenario can be crucial in case of confidential and sensitive information such as credit cards information or health information. Thus data anonymization, authorization and encryption need to be guaranteed in such environments. In term of exploitability it is rated as AVERAGE and EASY in term of detectability. As for the impact, it is rated as SEVERE.

6. Insecure Cloud Interface, **7.** Insecure Mobile Interface, **8.** Insufficient Security Configurability are also critical aspects that must be taken into account while dealing with such environment [2].

9. Insecure Software/Firmware: even if the exploitability of this kind of vulnerabilities is DIFFICULT, its impact is considered as SEVERE, since it can lead to compromission of user's data and even to take control over the device. The device needs to be able to perform updates regularly, specially when vulnerabilities are discovered. The updates need also to be protected, since software/firmware updates files delivered on insecure network connection are susceptible to altering attacks. Consequently, encryption and integrity checks need to be performed. This vulnerability can be EASILY detected by inspecting the network traffic during the update and check the encryption.

10. Poor Physical Security: finally, physical attacks are widely used to access the operating system and the sensitive data stored in the device such as the encryption keys and the credentials, by disassembling it and accessing the storage medium. Precautions can be made by encrypting the stored data and ensuring that the USB ports can not be used maliciously. In term of exploitability and detectability it is rated as AVERAGE.

C.1.3.2 Security requirements for the IoT

The exponential growth of the number of deployed devices and the size of data that will be exchanged on the network raised several challenges in order to achieve a global architecture. Those challenges concern not only the operational part, but also, and most importantly, the security and the privacy in such environment[189]. What makes the uniqueness of the IoT are the properties that need to be treated in order to define the security and the privacy challenges. Such properties, as explained in [190], are the result of the analysis of many related researches in the IoT field, they are mainly four caractéristiques: 1) *Uncontrolled environment* which is natural for an environment such as the Web especially when dealing with mobile Things (from one domain to another), physically accessible and requiring the establishment of trust relationships in order to exchange sensible information. 2) *Heterogeneity* since the IoT environment may integrate various types of entities coming from different origins. 3) *Scalability* related to the plethora of Things that need to be interconnected, hence a highly scalable protocols need to be applied and 4) *Constrained resources* in term of energy, computation

capabilities and storage space. The same analysis [190] shows that the security requirements can be grouped into five main sections: Network Security, Identity Management, Privacy, Trust, and Resilience.

Network Security:

Preventing eavesdropping, tampering, spoofing, denial of service, and so on, of sensitive information when they are sent via the Internet, either from a Thing to another or from a Thing to human, is an important requirement for network security. *Confidentiality* requires the establishment of a secure communication for the IoT's smart objects, specially when they communicate through the Internet. Traditionally, several technologies such as IPSec and TLS have been proven to fulfill the requirement, however they require significant cryptographic computations that exceed the capacities of the current IoT devices. Thus, dedicated secure network stack for the IoT needs to provide strong and lightweight encryption, so that the constrained devices can benefit from the same security functionalities that are typical of unconstrained domains. Most of the solutions trend to use a trusted unconstrained node to offload the computationally intensive tasks such as the calculation of the master session key. Another property guaranteed by the encryption is the *Integrity* of the data to ensure they are not altered during their way to the destination. While *Authenticity* provides proof that a connection is established with an authenticated entity, it can also include the integrity. And finally, the Things need always to be available meaning that the connectivity of a Thing should persist even under link failure, referring to the *Availability* property. *Secure routing* is also one of the issues that can occur in the network in order to ensure the delivery of the packets, such as the detection of malicious nodes, the confidentiality of network topology and the stability against attacks. It can be instantiated through the implementation of secure routing with a strong protocol such as RPL [191].

Identity Management:

First of all, each object needs to be aware of its own resources such as identity, constraints, security requirement, etc. Due to the enormous number of devices deployed in the Internet, and the complex relationships that they can have with each other, appropriate identity management mechanisms need to be present. Still identity management alone is not sufficient showing the importance of authentication, authorization, accountability and revocation mechanisms. *Authentication* is very important to IoT and is likely to be the first operation carried out by a node when it joins a new network, which appears in first deployment or mobility cases as examples. Usually authentication is performed using an authentication server with a network access protocol such as PANA or Extensible Authentication Protocol (EAP) [178]. As for the management of the access *authorization* and the ownership of resources, federated authorization such as Kerberos and OAuth have the possibility to provide delegation of access across domains and provide quick revocation. A presentation of an authorization framework is presented in C.1.3.3.

Privacy:

Objects dealing directly with the private information of individuals and organizations raise a challenging privacy issue in IoT. The environment needs to provide data privacy for the data transmitted in the Internet, in the sense that traffic sniffed containing those data will not reveal/expose its content. Moreover, the emergence of the new concept of the "Privacy by Design", introduced in [192], defines guidelines for every new technology that process personal data, and in particular the sensitive ones, in order to guarantee their security from the early stages of the design. In particular for IoT devices, since, they continuously collect and process personal data, while being connected to other entities and other devices and to the Internet. These guidelines implies mechanisms such as data minimization, anonymity, unlinkability, purpose limitation, and so on and so forth.

Trust:

Giving a proper definition of the Trust specially in a distributed architecture such IoT is still a challenge, since any trusted entity can become malicious either intentionally or after being compromised. However, the Trust in this case can be separated into three parts. The first part is the *Device Trust*, since a prior trust cannot always be established due to the mobility and the distributions properties of IoT. However, approaches such as trusted computing [193] and computational trust [194] can solve the problem. The second part is the *Entity Trust* referring to the expected behavior of the different entities. This part presents more challenges. Solutions based on the behavior analysis and the application of proper policies need to be investigated. The last part is the *Data Trust*, which can use the previous established trust relationships to judge the trustworthiness of the data e.g., data originating from a trusted entity might be also trusted.

Resilience:

And finally the Resilience requirement, where the IoT applications need to ensure the availability of the resources in case of system failure, and also to have robustness against the different attacks.

C.1.3.3 Example of security model in IoT: OAuth-Based Authorization Service

Authorization as we defined it previously, is a critical aspect in accessing the objects in a secure way, and preventing unauthorized users from extracting information, in particular the private ones. Many innovative architectures have been proposed in the last years. One of the main problems related to the authorization process (for example with OAuth) is the limited computational power and the communication constraints of the smart objects, since some cryptographic primitives such as computing checksums or digital signatures require both processing power and energy consumption. Moreover, if the access policies for the services provided by the Smart Object reside on the

Smart Object itself, it could be extremely hard to dynamically update them once they have been deployed. The proposed architecture, described below, solves this issue by deploying an external authorization service based on OAuth.

OAuth-Based Authorization Service for IoT (IoT-OAS), is a framework based on HTTP/CoAP services to provide an authorization framework, in order to protect the privacy of the personal information. This authorization is based on the protocol OAuth allowing a secure authorization from third-party applications, and benefiting from the advantages of using such delegated authorization protocol such as lower processing load, fine-grained customization of access policies and scalability. The proposed framework uses the access token in order to access the IoT application resources, and also a delegation approach [195].

All along this thesis, guaranteeing most of these requirements was our objective for all our contributions, starting with the implementation of the security properties such as the authentication of the users when requesting the access to an IoT resource. In this case using strong credentials (eventually strong authentication), reduces the attack surface. Then, by guaranteeing the confidentiality and the integrity by encrypting all the traffic between the different components of each architecture. And finally, by controlling the access to the IoT resources so that only the authorized ones have a granted access. Other properties such as the availability, the accounting and the non-repudiation need more efforts and time to be enforced, and which can be subject to future work such as with the non-repudiation by using block chain to create smart contracts, and so on and so forth.

As for the security requirements, during this thesis, the network security was managed only at the application layer by applying encryption , the other aspect of the network security such as the secure routing are out of the scope of this thesis. Since the identity management, the trust and the resilience of the IoT devices are considered as huge issues by them self, in addition to their constrained nature, it was considered also out of the scope of this thesis. However, we argue also that the privacy is a big concerns and needs particular attention, since the user's of such architectures are very concerned regarding their personal data. Thus, in our first contribution, a privacy and risk analysis is provided.

Now that we have a global overview of the threats and the security requirements in the IoT, we will introduce the security of the WoT by listing the currently proposed models, mainly the identity management, data confidentiality and integrity, the authorization and the access control.

C.2 Security in the Web of Things

In this section we will first go through the problematics related to the identity management, we then focus on data confidentiality and integrity, and finally we explore the

authorization and the access control models.

Generally, openness and sharing in any ecosystem come always with security and privacy issues, same applies to the WoT. Things' shared resources and data need to be protected against vulnerabilities raised from malicious intervention and inadvertent errors. In the last few decades, several Web services based solutions have been proposed to address those privacy and security issues. However, those solutions in most of the time are not compatible with the constrained environment such as the WoT. Moreover, it does introduce new dimensions of risk due to its heterogeneous nature. Some of the main threats related to WoT can be summarized in the following list:

- **Impersonating a Server or a Thing**: if a WoT architecture relays on a Web server that acts as a proxy to deliver requests to the right destination, Things discovery or other purposes. An attacker can take control of the server and present itself as a valid server. Hence, all the traffic going through it will be compromised by the attacker including credentials and users/objects identities. Furthermore, an impersonated Thing can reveal personal information, or send malicious code that can be injected in the requester side.
- **Tampering attacks** against Things' resources.
- **Unauthorized access** to the Things' resources.
- **Eavesdropping** of the traffic flowing between the different entities in the WoT environment, hence compromising the confidentiality property.
- **Denial of Services attacks**, which aim the unavailability of the Objects in the Web, by submerging it with excessive amount of network traffic.

To face such threats, WoT needs to provide some security and privacy guarantees while taking into account the mobility and the size of the Network. It is necessary to protect the Things private resources and critical information from being accessed, modified or inserted by unauthorized entity. Thus, Authentication, Authorization and Access control are indispensable requirements for the WoT, combined with efficient Identity management and policies can provide a strong security and privacy architecture. Other security properties need to be also taken into account such data integrity and confidentiality which can be provided through secure communication through encrypted channels inside the WoT ecosystem. The rest of the section provides an overview of the different security mechanisms that are actually proposed or that are under development.

We will start with how the identity is managed in the WoT architectures, the different identity management models and we will conclude with an example in section C.2.1. Then in section C.2.2, a study on providing confidentiality and integrity of data through securing the channels between the communicating devices. Then in section C.2.3, we introduce the authorization approach that can be applied to a WoT architecture. We conclude in section C.2.4 with the analysis of the access control mechanisms that can be deployed in a WoT environment.

C.2.1 Identity Management in WoT

Controlling users identity is an important process in any application and system. Such control includes authenticating users, identifying them and life cycle of such identities. This process is also applied on the user's data inside an ecosystem, since personal information such as identity, credentials, social security, etc., must be protected against unauthorized access. Thus, identity management defines simply the rules to identify individuals in a given system, through their identities and depending on the circumstances.

However in our case the encountered problem is identifying the smart objects in the WoT. Undoubtedly in WoT, we will have interactions of the form object-to-object where an object A will send information to object B without human intervention, and object-to-human, where sensors will send their results to the human. So obviously for security reasons, identity control in these use cases is crucial, since we exchange and entrust more and more personal information to the smart objects, that are directly exposed to Internet.

In fact several questions raise such as: How to uniquely identify objects in WoT? And how to provide these identities in the object-to-object use case, and the object-to-human use case? Finally, how safe the identity of the users and objects in the WoT ecosystem is? We will answer these questions in the next sections of this chapter by exploring some researches in this domain and through some real-life use cases deploying such techniques.

C.2.1.1 Identity Management Models

The general architecture of the identity management (IdM) models in the WoT ecosystem is composed of an Identity Provider (IdP), a Service Provider (SP) and the user/object. In this section we will provide a brief sketch of the existing identity management models [196].

- **Centralized federation model**: where there is a unique trusted IdP responsible for collecting and provisioning users with identity information. Usually located in a secure domain, this model enables Single Sign On (SSO), and sharing the users identity information with different SP. However, the IdP suffers from the problem of single point of failure, if this part fails the entire identity management system will fail.
- **Decentralized federation model**: the IdP's functions are distributed among several IdPs, and in different secure domains. This model needs to establish a trust relationships between the SPs and the IdPs, in order to provide SSO to users affiliated to different IdPs and SPs. However in this model, the user does not have the full control of his identity information, since they are stored in the IdP, and they can be disclosed to a third party without his permission.

- **User-centric model**: this model solves the problem of controlling the user's identity, by providing the user with full control of all the transactions involving his identity. Concretely, the user needs to explicitly approve the use of his identity. The user/object can have one or more identities issued by one or more Identity Provider. Such system needs to guarantee several properties, some of the basic ones are the confidentiality, the integrity and the unlinkability. The complete taxonomy of properties related to the user control is illustrated in [196]. An example of user centric IdM is illustrated in [197].

C.2.2 Data confidentiality and integrity

Securing the communication between the different components of the Web of Thing environment is mandatory, in order to preserve data confidentiality and integrity and to prevent a third party from eavesdropping and intercepting information exchanged between the different entities of the system. However, encryption in a constrained environment such as WoT can be problematic since cryptographic computation usually requires memory and energy which are not always available in the smart devices. In what follows we will go through two examples of end-to-end encryption, the first one is based on securing the communication at the transport-layer, and the second one for securing the communication at the application layer.

C.2.2.1 End-to-end security in CoAP's Transport-layer

Several security-related issues are raised in [198], on how to allow secure communications on the WoT, either for Thing-to-Thing communication or for Human-to-Thing, since most of the applications currently envisioned for the WoT require strong security assurances while taking into consideration the constrained environment of the smart objects. Those issues need to be handled before it can realistically be considered ready for deployment.

For this objective, an experimental evaluation of the different mechanisms proposed to secure end-to-end web communications with IPv6-capable sensing applications and devices have been realized in [198]. Those experiments are based on the analysis of Constrained Application Protocol (CoAP) protocol [49] maintained by CoRE (Constrained RESTful Environments), by using the Representational State Transfer (RESTful) web services with IPv6-enabled sensing devices. In CoAP the security is not integrated at the application-layer protocol itself, but rather directly applied to all CoAP messages at the transport layer of the protocol. Thus deploying DTLS in protecting communications at the transport layer appears as a logical choice, since currently 6LoWPAN environments employ UDP, at least from the standardization's standpoint. CoAP proposes three security modes based on the usage of DTLS to secure web communications with smart objects: 1) RawPublicKey 2) PreSharedKey, and 3) Certificates all employing the CoAPs scheme when contacting a DTLS-enabled CoAP server [49]. The RawPublicKey and Certificates modes employ Elliptic Curve Cryptography (ECC) by using

the Elliptic Curve Digital Signature Algorithm (ECDSA) for devices and messages authentication, and the Elliptic Curve Diffie–Hellman (ECDH) for the key agreement. The PreSharedKey mode used in the case where the smart devices already store some predefined keys either provided by the manufacture or the developer or simply by the user. Those keys will be used to secure the communications with other devices.

The experiments done in [198] analyze the impact of the different CoAP's security modes on the network-layer payload space, the memory footprint and the computation and the energy overhead of the smart devices deploying theses security modes.

From the first analysis on the overhead on network-layer payload space, they concluded that even if the impact of CoAP security on the payload space available to applications is visible (up to 33% of the payload), it may be considered viable for applications that are thrifty with respect to payload space requirements. However, fragmentation is unavoidable in case where the applications requires a larger payload.

From the second analysis on the memory footprint of CoAP security, they concluded also that hardware-level encryption does not come without a non-negligible overhead on memory, especially the ROM memory (in the best case 78.9% will be needed in the Certificate mode). They also identified a major limitation that there is not enough ROM memory available to support the RawPublicKey mode. RAM is potentially a problem in this case, since 88.6% of memory usage in case of RawPublicKey mode may compromise the usage of other required applications in the device. Elaborated sensing devices with more memory will be required in the future to appropriately support ECC-based security modes. The remaining CoAP security modes appear to be valid in this case.

The last analysis concerns the computational and energy overhead of CoAP security, the outcome is that a significant impact of ECC on the performance and energy of current sensing platforms, inevitably influences the lifetime of sensing applications and its maximum achievable transmission rate. The advantage of hardware-based cryptography is again expressed by the values obtained from the PreSharedKey mode, therefore a good choice when pre-deployment and configuration of security-related parameters on sensing devices is desired. For scenarios where public-key cryptography is required, the Certificate mode appears again the best alternative to ECC.

The analysis concluded that the modes deploying the Elliptic Curve Cryptography consume more memory and energy, and cannot be used for high transmission rate. However in the future the ECC approach can become very promising with the evolution of the capacities of the smart devices due to the strong cryptographic properties proposed.

C.2.2.2 End-to-end security in CoAP Application-layer

Indeed, application-based security have been proved to have the capability of interpreting and interacting directly with the information contained in the payload portion of

a datagram such as the application proxies used in most firewalls for FTP transfers. These proxies have the ability to control and restrict the use of certain commands even if they are contained within the payload part of the package. Moreover, lower-layer security protocols like DTLS do not have this capability. They can encrypt the commands for confidentiality and authentication, but they cannot apply restriction and policies for their use.

Another analysis, done in [3], focusing this time on the security of the end-to-end communications between devices at the application-layer rather than the transport layer of the protocol. Respecting the limitation brought by the WoT applications, this mechanism is employable in the context of a general security architecture supporting web-enabled sensors (Things).

Historically, several approaches were proposed to address in particular the end-to-end communications security between 6LoWPAN wireless sensing devices and Internet hosts. However, most of them are not compatible with the current vision of WoT using CoAP and 6LoWPAN as standards. We mention Sizzle [199] which is one of the smallest web server providing HTTP accesses secured using SSL's 160-bit ECC keys, for key negotiation and authentication, but requiring a reliable transport-layer protocol and therefore was incompatible with CoAP and 6LoWPAN at that time. However since then researches have evolved enough so that CoAP can now provide reliable transmission of messages according to the RFC7252 (section 4.2 in [49]), by adding the *Confirmable* option to the CoAP's header. This would be a possibility to a future analysis of using Sizzle with the current CoAP. However, another issue with Sizzle is that it does not support two-way authentication which is required by many Machine-to-Machine (M2M) applications on the WoT environment. As an alternative, Sensor Networks for All-IP world (SNAIL) [200] secured using a lightweight SSL (SSNAIL) is another solution proposed to solve the two-way authentication using an ECC-enabled handshake instead of RSA[201], also requiring a reliable transport-layer protocol that was not available at that moment. This reliability can also be brought by CoAP, thus extra analysis can be done to check the compatibility of SNAIL in the WoT environment. More in line with the security in the application layer, a proposed solution for the integration of security with CoAP using options for the activation/deactivation of security contexts and for the protection of CoAP messages have been proposed and abandoned in an IETF draft [202].

The proposed mechanisms integrate and evaluate the security at the application-layer with the CoAP, by adding new security options to CoAP. The first option is *SecurityOn* which precises if the received CoAP message is protected with an application-layer security or not. This option states which security is applied (encrypted, signed or both) in the SecurityApplied field; the Destination Entity identifies the actor CoAP URI that the destination must handle, this option can be employed several times in the same CoAP messages to enable the traversal of different trust domains and possibly using different encryption keys; the timestamp to verify the legitimacy of the message; and finally the Context identifier enabling the receiver to contextualize the message in term

of security, in particular deciding the appropriate ciphers and keys.

The second option is the *SecurityToken* enabling the use of authorization and identity mechanisms. With this option, the requester may identify itself to obtain the access to a given CoAP resource. This option enables also request authorization based on a per message basis. Using the "TokenType" field, the requester can precise the authentication mechanism, that can be either a simple authentication process using its Username and Password or with a more sophisticated authentication process using its public-key, its X.509 certificate referred by a URI, or a kerberos ticket obtained from a domain server.

The last one is the *SecurityEncap* option. It transports the security related information required for the processing of the CoAP message, which depends on the content of the SecurityOn option. If only an encryption is required in the SecurityOn message (SecurityApplied set to 0), this message may transport a nonce plus the number of options followed by the encrypted data. If the SecurityOn requires a signature (SecurityApplied set to 1) this message may be used to transport an encrypted MAC plus a nonce value for freshness purpose. If the SecurityOn message requires both encryption and signature (SecurityApplied set to 2) a nonce, a MAC, a number of options and encrypted data may be transported with this option. More details on these new CoAP's security options can be found in [3].

An evaluation of the deployment of these options have been achieved to evaluate the impact of the end-to-end security on the CoAP payload, the lifetime and the communication rate of the sensing application. The test platform is composed of a CoAP sensor (TelosB), an Internet host using CoAP and a CoAP intermediary (a forward proxy) that will be used for security processing. Via the SecurityToken option, the intermediary will provide authorization of CoAP clients and access control to the resource. As explained in Figure C.2:

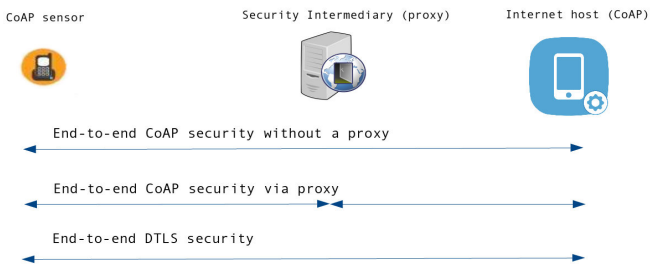


Figure C.2 – CoAP and DTLS security end-to-end usage scenarios [3]

The first evaluation is the impact of end-to-end security on the payload space of a CoAP packet, it computes the space needed to add security in the application layer. The analysis shows that the usage of the CoAP security intermediary in the encryption and the signature performs even better than using only CoAP with DTLS in the

transport-layer. Thus the usage of the intermediary allows offloading the security computation, and guaranteeing a very small impact on CoAP's payload space. The analysis also shows that in the worst case when using encryption, signature and authentication, 65% of the original 6LoPWAN payload is still available. Hence, this approach is viable from the payload space point of view.

The second evaluation is on the lifetime of sensing application, by computing the energy consumption of the device when using such security options. It shows that the usage of the CoAP security intermediary in the encryption and the signature perform even better than using only CoAP with DTLS in the transport-layer. However, end-to-end CoAP security without an intermediary has bigger impact on the expected lifetime, specially for lower communication rates where the cumulative impact of AES/CCM encryption in the default CoAP security context is lower compared to the impact of the energy required to process and transmit CoAP security options. Consequently the obtained results show that CoAP security provides acceptable lifetime values in all usage scenarios, particularly considering the WoT applications natively designed to require low or moderate wireless communications rates.

Finally concerning the impact of end-to-end encryption on the communication rate of sensing applications, the study shows that using CoAP signing and encryption of all messages (as using DTLS) provides inferior performance. However, the results are still clearly above the requirements for the number of CoAP protected messages per second of most CoAP wireless sensing applications envisioned for the WoT, even in the worst case.

The overall evaluation of the proposed mechanism shows that CoAP application-layer security may perform similarly or better than transport-layer security. This approach brings new security to CoAP messages that are not possible in the transport layer to allow secure end-to-end communication for WoT wireless sensing applications, but still needs further investigation specially in term of key management and synchronization mechanisms [3].

C.2.3 Authorization in WoT

Allowing fine-grained and flexible access control to only authorized parties is crucial to an open ecosystem such as WoT, where objects are part of the World Wide Web and easily discovered. Traditional cryptographic algorithms and protocols might not be feasible due to the constrained nature of smart objects. Most of the actual solutions aim at setting a distributed authorization architecture, where a back-end server deals with the complicated tasks, which needs processing resources while letting the constrained devices handle the minimum of messages. The server is usually located between the smart device and the requesters. However, the device also needs to be able to distinguish the different requests coming from different entities and to apply the right authorization decision.

At the same time, several lightweight cryptographic algorithms such as (SEA, PRESENT, OAuth [203]) appeared specially to satisfy this purpose. Another solution is Delegated CoAP Authentication and Authorization Framework (DCAF) [204], allowing the delegation of the complex cryptographic computation to external entities, and establishing a secure DTLS channel between resource-constrained nodes. This protocol can be used to delegate authentication of communicating peers and authorization management to a trusted third party with more computation power, memory and energy [205].

As discussed in the previous sections, constrained environment needs to be treated differently than the normal environment. Moreover, those constrained nodes are expected to be present in various aspects of everyday life, hence they will be entrusted with large amount of personal data very likely susceptible to various attacks. For this reason, authentication and authorization are required for a secure WoT.

In some cases, static configuration of the authentication and the authorization process can be efficient, just as the case of prefixed silos of users or purposes, by statistically defining the access lists and the trusted entities when first deployed or by the manufacturer. However in case of flexible access to already deployed Things available to various number of users through the Web, this options seems to be obsolete and inefficient. In addition, authorization and privileges may change depending on the circumstances and the policies of the environment such as modifying the privileges of a black listed malicious entity.

For this end, an authorization and authentication architecture is proposed in [4] exclusively for constrained environments, where complicated security tasks will be assigned to another trusted entity, or by getting help from less constrained actors in the system. In this architecture, each entity will be assigned a constrained level ("constrained level", "less-constrained level", etc.). The less constrained nodes, also called Authorization managers, will perform the complex security tasks on behalf of their respective managed nodes, such as managing keys, enforcing authorization policies, etc. Figure C.3 shows the overall authorization architecture:

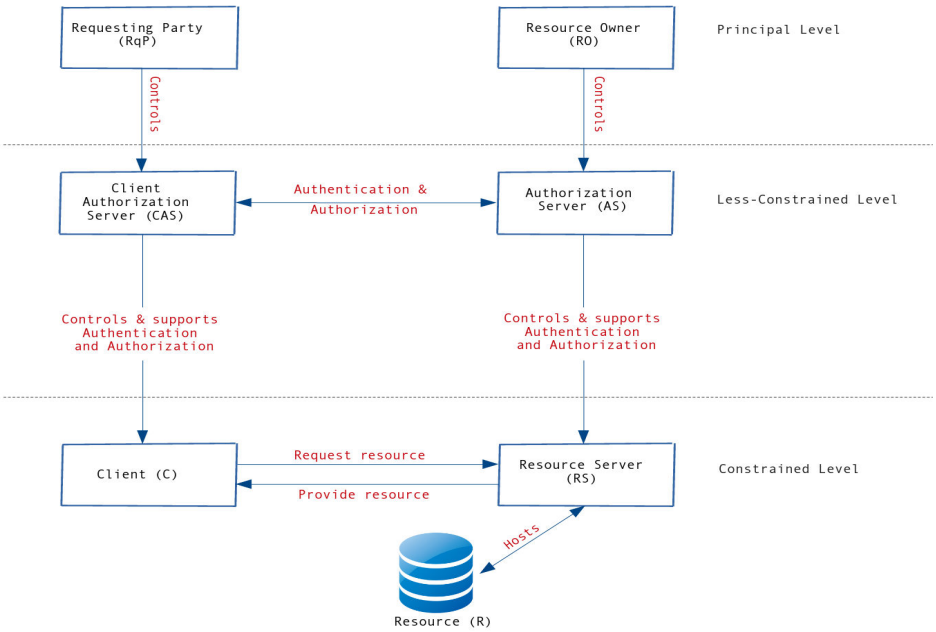


Figure C.3 – Overall authorization architecture [4]

The components deployed in this architecture with their roles are such as follow:

- The *Resource Server (RS)* is hosting and representing a resource. It can be a SO or a traditional server (less constrained device).
- The *Client (C)* is an endpoint requesting a resource on the resource server (RS). The endpoints may or may not have a trust relationship, it can also be a constrained device (Thing) or not.
- The *Authorization Server (AS)* is in charge of creating and approving the authorization and the authentication data for the RS. It's a less constrained level of the architecture (in term of memory, energy and processing). Plays the role of backup for the RO and acts on behalf of it to handle the access requests to the RS. Authorization and authentication mechanisms can be deployed also between AS and CAS in order to further relieve the constrained level.
- And finally the *Client Authorization Server (CAS)* in charge of creating and approving the authorization and the authentication data for the Client. It is also a less constrained level of the architecture and play the role of backup for the RqP and act on behalf of it to handle the access requests to the Client.
- The *Resource Owner (RO)* is the entity (principle) that owns and controls the resource and also grants permissions using mechanisms such as OAuth [203] and User-Managed Access (UMA) [206]. The RO controls and makes authorization decision for the RS. Basically if an entity is not authorized by the RO it cannot access the resource R, hence performing authorization in the RS side.
- The *Requesting Party (RqP)* is the principal in charge of the Client, it controls requests that the client makes and the acceptance of the received responses. Precisely, it controls the interactions that the Client may operate with other endpoints and makes authorization decisions on behalf of the Client. Basically the

client cannot exchange information (requests/responses) with a resource without the authorization of RqP, hence performing authorization in the requester side. Furthermore RqP may provide enough information to CAS to autonomously negotiates the access to RS with AS on behalf of the requesting Client.

- The *Principal* it can be either an RqP or a RO.

To summarize the aim of the overall architecture, the interactions between the constrained nodes must be controlled via the less-constrained level entities (AS and CAS) that act on behalf of the respective principals of the endpoints (RqP ans RO). Securing this interaction together with the control messages, by the bias of cryptographic keys, is also a requirement. The connection between the constrained nodes and the less-constrained nodes should be protected using a symmetric pre-shared keys and credentials. Also, protecting the connection between constrained nodes needs to be considered. This solution addresses also the confidentiality, the integrity and the availability problems since only the authorized entity has the right to access the resources, hence less charge on the system. Moreover, adding authentication to this architecture guarantees the accountability property and the authentication/verification of third parties.

There are other variants for this architecture depending on the scenario, for example some components can be merged in a single entity, such as combining the Client and the CAS if the Client has enough capabilities. A deeper analysis can be found in [4]. In what follows, we will present two authorization frameworks, the first one is a concrete implementation of the overall architecture presented above, and the second one is a token based mechanism.

C.2.3.1 Authorization framework example 1

An implementation of the previous architecture was proposed in [205]. It's a protocol for delegating the heavy security tasks such as authentication and authorization information to a less constrained and trusted entity. It relies on the DTLS to send authorization information and shared secrets (basically symmetric cryptographic keys) between nodes in a constrained network. It uses the notion of access token to implement the authorization architecture for constrained environment such as WoT. This protocol is called DCAF [205].

The goal of DCAF is mainly to setup a secure DTLS channel between two constrained nodes using the symmetric pre-shared key (PSK) cryptography, where the most sophisticated tasks are handled by the less constrained nodes. Moreover, DCAF ensures secure transmission of authorization tickets and enforces the authorization policies defined by the respective principal of the constrained node. Another advantage of the protocol is the support of implicit authorization where no authorization information are exchanged, hence a simplified authorization mechanism.

The actors of this implementation remain the same, with a difference in the terminology:

- Server (S) referring to the “Resource Server (RS)“.
- Client (C) remains the same.
- Server Authorization Manager (SAM) referring to the “Authorization Server (AS)“.
- Client Authorization Manager (CAM) referring to the “Client Authorization Server (CAS)“.
- Authorization Manager (AM) which can be either a SAM or a CAM
- Client Overseeing Principal (COP) referring to the “Requesting Party (RqP)“.
- Resource Overseeing Principal (ROP) referring to the “Resource Owner (RO)“.

Figure C.4 is a global overview of the proposed architecture, it summarizes the main interactions:

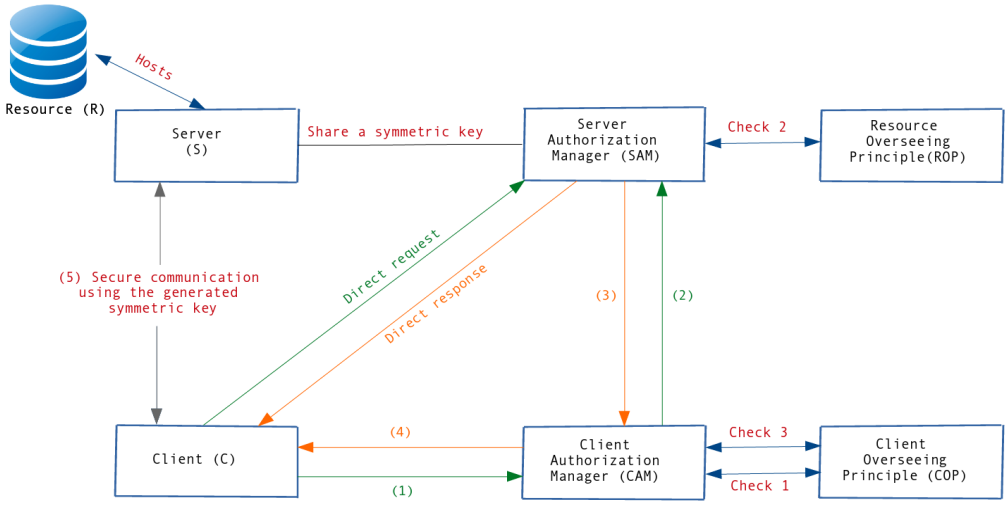


Figure C.4 – DCAF’s overall authorization architecture

As explained before in the authorization architecture, each server (S) (which is a constrained environment that hosts a resource and it can be a Thing) is controlled by a Server Authorization Manager (SAM) that will perform the authorization and authentication process on its behalf. They both have already a pre-shared symmetric key that was exchanged initially using a key exchange mechanism.

When a client requests an access to the server’s resources, he/she has first to ask the SAM for an “Access Ticket“. The client can be either a constrained device (a Thing) or a less constrained device. In the case of a less constrained client, he/she can directly request the access ticket from the SAM, that will perform the “Check 2“ to verify if the client is authorized to access the server’s resource, by consulting the Resource Overseeing Principle (ROP) policies. Once the check is done correctly, the client will get

an access ticket that contains a PSK, and also some details for the access permission in case of explicit authorization. Using this PSK, the client can directly and securely communicate with the server.

In the second case where the client is a constrained device, in order to get the access ticket he/she needs to go through a CAM that will act on his behalf, represented as step 1. The CAM will perform "Check 1" to verify if the server is an authorized source for the wanted resource, by consulting the policies defined by the COP, and then will forward the request to SAM (the step 2). Similarly, SAM will perform the "Check 2" to verify if the client is authorized to access the server's resource, by consulting ROP's policies. Once the check is done correctly, it will return an access ticket that contains a PSK and also some details for the access permission in case of explicit authorization which is represented by the step 3. Another check is done by CAM to verify if the permission in the access ticket complies with COP's authorization policies for the client ("Check 3"), then sends the ticket to the client which is shown in step 4. Finally, the client can use the PSK to directly and securely communicate with the server in step 5.

Some requirement has to be fulfilled by the authorization manager (SAM and CAM) in order to be able to provide the authorization and the authentication services. Mainly, they need to have enough storage space to store the different credentials, to be able to directly interact with the user, for example, through an interface, and of course have enough processing power to handle in one hand the different authorization requests, and in the other hand to be able to efficiently generate the PSK for a given client.

For security purposes, the channels between CAM and the Client, the channel between SAM and Server and the channel between CAM and SAM are encrypted by DTLS using the pre-shared keys. Also an authentication must be done between SAM and CAM to determine if the request is authorized or not. In this case, CoAP is used to interchange access-related data between the server and SAM in order to provide the server and the client with enough information to establish a secure channel. More details on the messages exchanged and the Ticket are presented in [205].

C.2.3.2 Authorization framework example 2

The second authorization framework complements IoT-OAS, an authorization service architecture based on OAuth for secure services in IoT scenarios [195] as we explained in C.1.3.3, by introducing access token mechanism to access the IoT resources. The analysis is also based on the delegation approach to authorization introduced by IoT-OAS, which can also manage fine-grained access to web resources in the WoT.

The proposed system is composed of four parts: the smart objects, the owner, the external user who wants to access the object and the delegation authorization service IoT-OAS. A preliminary process is the authentication of users into IoT-OAS using one of the known authentication mechanisms (Google login, Tweeter login, Facebook login, etc.), once logged the user will be granted an access token to authorize him to interact

with IoT-OAS.

The permission hence will be expressed by the tuple $< res, act, exp >$ where res is the URI of the object, act is the REST method that will be sent to the object and exp is the expiration time. The full method and messages are expressed in [207].

In fact owners prefer not only to restrict access to their object's resources, but also want to be able to grant access to authorized parties. In this case we distinguish two possible access policies:

1. Owner-to-Owner authorization: where the owner explicitly gives permission to himself to access or control an object. This type of authorization can be addressed using the OAuth 2.0 protocol [203]. The owner reads the UUID of the object, and sends a register request to IoT-OAS, which verifies if the object belongs to someone else, if not, will bind the object to the owner and grant him an all-access token allowing him to execute any operation on the object. The full message flow is explained in Figure 3.a of [207].

2. Owner-to-Any authorization: where the owner can grant authorization to access one of his/her objects to other parties. This type of authorization in the web can also be addressed by the OAuth 2.0 protocol using the User-Managed Access (UMA) profile. This type of authorization can be either *Reactive*, where the external user asks the owner for permission, as explained in Figure 3.b of [207], or *Proactive* where the owner gives directly the authorization to the external user, as explained in Figure 3.c of [207].

C.2.4 Access control in the WoT

Traditional access control focuses on the protection of data based on the identity and attributes of the users. And generally the access control is used to protect front-end and back-end data and system resources by adding restrictions on who can access the data, what users can do, which resource they have access to and what operations are allowed to be performed on the data. Ideally, access control prevents unauthorized users from viewing, modifying or copying the data. The basic model of access control can be summarized in the Figure C.5, where mainly an entity A wants to access entity's B resources, this request must go through a guard that will either allow or deny the access. The deny use case has also two branches, if the number of attempts of the same user reaches the legal one, the system will automatically drop his request and can also be black listed, else the user can resend an access request.

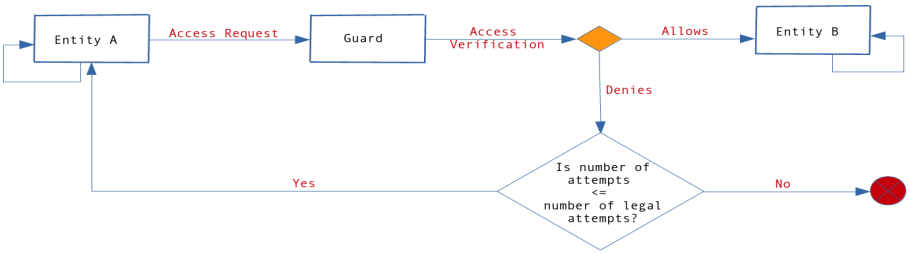
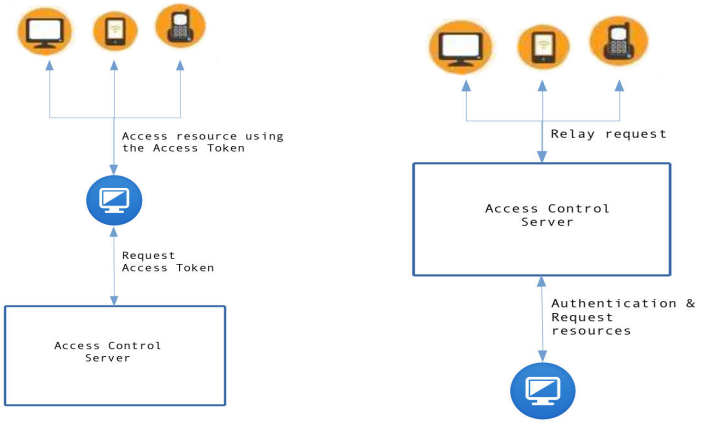


Figure C.5 – Basic access control

The WoT context enables the smart objects to publish and exchange their information over the Web. However, some security preventions needs to be taken into account, in order to deal with threats regarding the information exchanged over the Web. In particular regarding the access permission for the Things information and resources, such threats can be malicious clients, unwanted data sharing, divers attacks, etc. Hence the question of how to allow Things to grant clients secure access to their resources in such environment?

For computer network systems and in order to facilitate the access control, standard authorization models, such as Access Control List (ACL), Subject/object access control matrix [54], Multilevel security using information flow [55], Role-base access control (RBAC) [56] and Attribute-based access control (ABAC) [57], dynamic authorization model have been suggested [58] and capability-based systems must be deeply analyzed before applying them in the WoT. The selected mechanism needs to respect the constrained nature of the WoT, the autonomy (since the objects inside needs to be able to communicate with each others over the web without humans intervention) and the security requirement.

In the literature, there are two ways to implement the access control for WoT: (1) A distributed fashion, where an access control server authenticates the user and grants him the appropriate access token, allowing him to access the Thing's resources for a certain period of time or permanently depending on the deployed policy, such as shown in Figure C.6.a. And (2) a centralized architecture where all the user's requests go through an access control server that authorizes and relays them to the right destination. In this case, there is no direct interaction between the communicating parties, such as shown in Figure C.6.b. Most of the access control models can be implemented in both fashions. The centralized model is interesting in the WoT since all the complexity can be carried out by the server. However, this will create the single point of failure, impersonation and privacy problems since all the requests and eventually responses are monitored by the server. The distributed architecture provides better scalability and privacy in the system, however it can be complicated to implement models such as RBAC and ABAC in WoT constrained environment since the Things themselves needs to check the received access token.



(a) Distributed Access Control model

(b) Centralized Access Control model

Figure C.6 – Access Control models

As stated in [208], the access control for WoT should have the following requirements: each object may publish its information as one or more web resource(s) over the web, those resources can be accessed via a basic HTTP/REST request. Finally permissions assignment can have a web resource representation, according to it, a permission grant decision can be made. In what follows we will provide two access control architectures, the first one is resource-oriented architecture, and the second one is a role-based access control architecture.

C.2.4.1 Example 1: Access control through resource-oriented architecture of the WoT

A decentralized access permission control using resource-oriented architecture for the WoT has been proposed in [209]. It adopt the REST-style to allow Things and users to manage access privileges to their own web resources, which are accessible through a unique URI. They propose 5 steps to control and assure any resource access request coming from an external user. Those requests must be transported via a secure channel. As it is explained is the same paper, the first step to control the access to the resource is by filtering TCP/IP packets, especially filtering out the unallowable domain access. The second step is to parse the HTTP/REST request by filtering out the invalid requests and the abnormal parameters. The third step is to checks the HTTP header for basic authentication purpose so that the unverified users will be blocked. The fourth step is to check whether the requested resource exists or not, since requests for expired or irrelevant resource must be automatically dropped. And finally, the system needs to check the assigned access permission for the request operation. Unassigned access permission for a requested operation must be filtered.

The study focuses on the CRUD operations that can be applied on the object's resources, where a specific permission policy can be applied to each REST request. This permission policy, expressed using XML, can contain the requester information such as IP address, domain, host name, etc, and also some contextual information about the

object itself, such as the time duration, localization, hardware state, capabilities, etc. An example of access rights for each CRUD operation to an object is shown in C.1:

$$Permission\_Flag\_C[/R/U/D] = \{X\} \cap \{Y\} \quad [209] \tag{C.1}$$

$$X = Condition\_about\_subjects$$
$$Y = Condition\_about\_the\_requested\_object$$

Handling the rules that have more than one condition has to be settled by the implementation. Assuming that all the conditions have the same priority to grant permission. There are two cases, either all the conditions need to be met or at least one of them. A more precise representation can be applied by dividing the condition part into two categories: inclusive and exclusive. In this paper the chosen rule was: "Permission is granted if any of inclusive conditions meets when any of exclusive condition does not meet"[209]. Then, the expression can be represented as shown in C.2:

$$Permission\_Flag\_C[/R/U/D] = \{X\} - \{Y\} \quad [209] \tag{C.2}$$

$$X = Union\_of\_inclusive\_conditions$$
$$Y = Union\_of\_exclusive\_conditions$$

C.2.4.2 Example 2: Role-based access control

Another example of access control is presented in [210], introducing the use of the RBAC [56]. It specifies the access policies for the plethora of data published by Things in the Web, and also how to access them and how the access can continue or should terminate. Cryptographic keys are also deployed to enforce those policies. The benefits from using RBAC reside in supporting a set of important security properties such as data abstraction, least privilege and the simplicity of adding access rights to users as long as the existing roles are used.

Generally each *User* is assigned a *Role* and each role is assigned to a *Permission*, hence users acquire permission to access a particular data depending on theirs roles. The user can have many roles and a role can be assigned to many users, same for the the role which can have many permission, a permission can be assigned to different roles. And finally, each assignation needs to take into account the different constraints, for example to enforce conflict of interest policies that Thing's owner may employ to limit the number of users able to access Thing's resources. A more elaborated analysis can be found in [211].

However, RBAC suffers from the *role proliferation* problem related to the issues of dealing with a large amount of data, for instance granting permission to a big number of users to access a Thing's dataset, where each permission depends on the user's affiliation to the system. More precisely, the problem lies in handling the task of assigning a single role to each user which can be complicated in this case. *Role parameterization* have been proved to be an efficient solution in these types of scenarios [212]. The global proposed RBAC/WoT architecture, as explained in the paper, can be summarized in

Figure C.7:

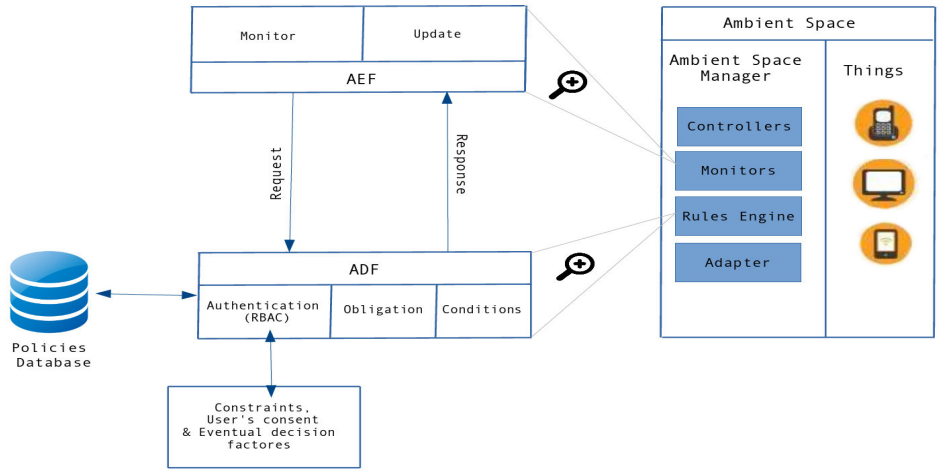


Figure C.7 – RBAC/WoT Architecture

The proposed security architecture in Figure C.7 aims at integrating RBAC model into a WoT-based environment by mapping the RBAC entities with the WoT entities. Where the Things in WoT are represented as users in the RBAC model, with a set of permission that will grant privileges to the users and eventually the Objects. Also a set of RBAC authorization rules needs to be respected in order to access any WoT entity (user/object). The access control of the object's resources is done centrally. To enforce the specified access policies in WoT, RBAC needs to use the concept of *reference monitor* (RM), a process continuously running inside a trusted computer base, referring to the core control mechanism for the access and the usage of digital information. The reference monitor, located in the ambient space manager, is composed of two main components 1) the Access Control Enforcement Facility (AEF), located in the Monitors section and the 2) Access Decision Facility (ADF), located in the Rules Engine section. The two components interact with each other in order to approve or block an access request. Mainly, the AEF intercepts the different requests, and asks the ADF for a decision. The decision is then enforced in the AEF. More details can be found in the RM standard and [210].
Other Works !!!!

C.2.5 Summary

In this section we went through the currently proposed architectures for securing the WoT in the literature. We started with, how the identity is managed in the WoT, by listing the different identity management models currently proposed. Then, how the data is secured during the communication between the different components of the WoT system, in order to guarantee the integrity and the confidentiality of the data. The majority of the proposals are related to the security of CoAP, which is considered as the ideal communication protocol for the constrained devices. Next, we presented the authorization frameworks that allow a fine-grained and flexible access control to

the SO's resources. And we concluded by presenting two models of implementing an access control mechanism in a WoT ecosystem. The first one is a centralized model, where all the access requests go through a server that decides either to authorize or to block the access. And the second one is a decentralized model where the device itself decides either to authorize or to block the access. With this summary, we concluded our overview of the security aspects in the WoT. We believe that the security and privacy in the WoT in particular, and the IoT in general, are a critical aspects that needs to be considered when developing new architectures. We noticed more and more interest by the working groups and the researchers. We hope that the future proposed solutions will be mature enough to cover most of the security and privacy properties, so that we can fully unlock the potential of the IoT/WoT.

C.3 Conclusion

Almost all the technologies are now available to implement a strong WoT and IoT infrastructure, starting from the first deployment of the Things until the step where each Thing is able to interact autonomously with other entities (either humans or objects). Indeed the revolution of the WoT will not stop here. Periodically, innovative ideas of smart objects appear, aiming at covering all the aspects of our daily life. However, the main reason why it has not been widely implemented is the lack of sufficient security and privacy protections. Users are worried about their personal data that they will share with the smart objects and more importantly who can access them. They prefer to have full control of their personal information and to have enough security mechanisms to protect their data inside and outside the infrastructure. In this chapter, we covered the advancement in the WoT security currently proposed by the researchers. We mentioned four main aspects.

The first one is about the *Identity management* that includes also the authentication of the users. As a remainder each WoT entity needs to be identifiable inside the ecosystem. Within this identity the requester can authenticate itself and ask for permissions to access resources. There is still a lot of work to do in this part, especially since the few current proposed models do not at least fulfill some privacy requirements such as the anonymity and the unlinkability. The second aspect covers the *End to end encryption* in order to guarantee the confidentiality and the integrity of the communication between the entities. Currently, the IETF working group are on the different security mechanisms related to CoAP. The third aspect is about the current *authorization* models. They are based on the use of a third entity to execute the excessive tasks in term of memory, energy and computation capabilities. This can be seen as one possible solution to the authorization. However, this solution introduces it's own security concerns such as the MITM attacks. Maybe in the future, objects will have enough capacities to perform this task autonomously without the intervention of a third party, or that new mechanisms will be proposed. Finally the existing *Access control* solutions suffer from the same restrictions as the authorization ones. The only proposed architectures are based on a third party. Moreover well defined policies needs to take place in order

to enforce this mechanism.

In our point of view, security and privacy models still needs to mature, and other aspects needs to be taken into account. Mainly the problems related to the trust relationship between the different entities of the environment. And finally, since undoubtedly vulnerabilities will appear, those systems need to be resilient, in order to prevent any threat and dysfunctionality in the future.

Chapter D | Résumé de la thèse

WebRTC est une technologie récente de communication qui permet d'établir des échanges multimédia conversationnels directement entre navigateurs. Sa richesse et sa versatilité laissent présager des opportunités inédites en termes de services de communication innovants. Nous nous intéressons dans cette thèse à des locuteurs dans un "Smart Space" (SS) défini comme un environnement centré-utilisateur instrumenté par un ensemble de capteurs et d'actionneurs connectés. Nous analysons les capacités nécessaires pour permettre à un participant d'une session WebRTC d'impliquer dans cette même session, les flux induits par les objets connectés appartenant au SS d'un utilisateur quelconque de la session. L'accès à un tel environnement, peut être soit passif via l'observation d'informations contextuelles fournies par les capteurs, soit actif par l'exécution d'une commande d'un actionneur, soit une combinaison des deux. Cette approche recèle un gisement de nombreux nouveaux usages. Nous limitons notre analyse à ceux concernant l'exercice distant d'une expertise et d'un savoir-faire.

D'un point de vue technique, il s'agit d'articuler de façon contrôlée les deux technologies différentes que sont WebRTC et l'Internet des objets (IoT)/Web des objets (WoT). Nous procédons dans un premier temps à une extension de WebRTC par WoT pour fournir à tout utilisateur d'une session WebRTC, un accès aux objets connectés du SS de tout autre participant à la session, en mettant l'accent sur la sécurisation de cet accès ainsi que sur sa conformité aux exigences de respect de la vie privée de l'utilisateur concerné. Le positionnement de notre approche dans le contexte des services de communication opérant dans les villes connectées, impose la prise en compte de SSs multiples et variés induisant chacun ses propres politiques de routage et de sécurité. Pour répondre à nos objectifs, il devient nécessaire au cours d'une session WebRTC, d'identifier, sélectionner, déployer et appliquer les règles de routage et de sécurité de façon à garantir un accès rapide et sécurisé aux différents SSs concernés. La seconde partie de notre travail consiste précisément à proposer une solution à cette problématique induite par des SSs distribués sur tout le réseau. Notre travail se décline en trois contributions principales.

D.1 Le couplage de WebRTC et du Web des Objets

D.1.1 Résumé

La première concerne la conception d'une architecture opérant la jonction entre WebRTC et WoT. Sa définition s'appuie sur l'analyse préalable de plusieurs cas d'usage

servant de fil conducteur. Cette architecture repose principalement sur une passerelle reliant ces deux environnements. De la sécurisation native de WebRTC, nous dérivons un certain nombre de mécanismes permettant de sécuriser le lien entre la passerelle et le client WebRTC ainsi que le contrôle d'accès au SS. Nous montrons la faisabilité de notre approche par le développement d'un prototype qui est mis à contribution pour explorer les cas d'usage initiaux. Nous analysons pour finir la conformité de notre système au RGPD (Règlement Général sur la Protection des Données). La pertinence des cas d'usage médicaux a été mise à l'épreuve d'un regard médical neutre grâce à des entretiens pratiqués avec des médecins et un modèle économique permettant une valorisation commerciale de la téléconsultation médicale, a été proposé.

D.1.2 Architecture

WebRTC permet la réalisation de communications multimédia temps réel entre navigateurs web. En introduisant une convergence native des services de communication synchrone / asynchrone, fixe / mobile, voix / donnée, WebRTC fait du web le support naturel et définitif de tout service de communication professionnel ou grand public. L'idée dans notre cas consiste à enrichir WebRTC en ajoutant de nouvelles fonctionnalités pour gérer les données liées aux Objets Connectés (OC) associés aux environnements respectifs de chaque participant d'une session WebRTC. Lors d'une session WebRTC, chacun des participants pourra consulter en temps-réel les données émis par les OCs liées aux autres participants.

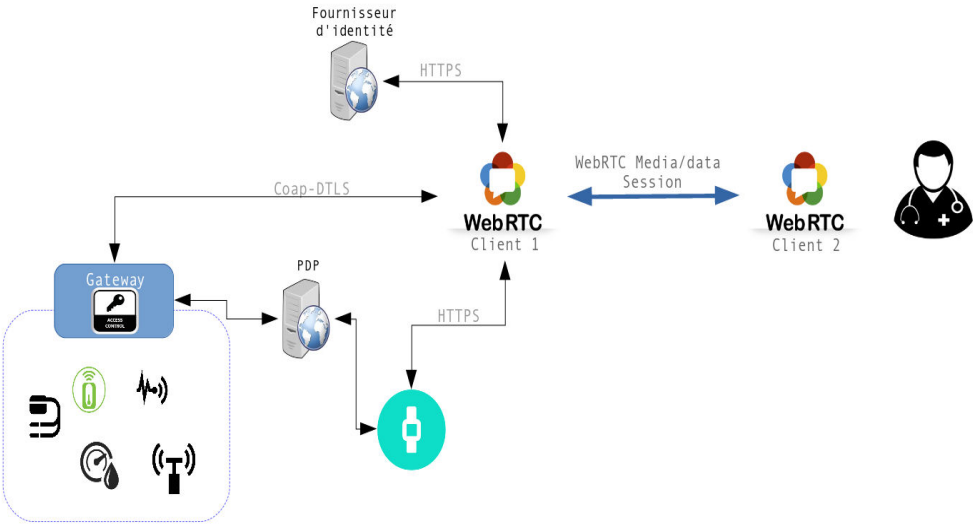


Figure D.1 – Architecture générale

Chaque client WebRTC (dérivée du navigateur à partir duquel la session s'exécute) fait office de passerelle pour ses propres OCs. Il est en outre possible de déléguer l'accès aux ressources des OCs à d'autres personnes externes, notamment celles impliquées dans la communication. La sécurité du dispositif global étant un point critique de cette famille de service, est prise en compte et analysée de façon détaillée. Des mécanismes sont introduits pour garantir la confidentialité, l'intégrité, l'authentification et le contrôle d'accès. Une vue simplifiée de l'architecture de la plateforme est présentée dans la

figure D.1. Comme indiqué précédemment, avec un simple navigateur compatible avec WebRTC (chrome, chromium, firefox, opera, etc.), un utilisateur peut, d'une part, communiquer en multimédia en temps réel et d'une autre part, accéder aux ressources des OCs de l'ensemble des participants. Nous supposons que chaque OC est identifiée via une URI et offre une API Web permettant l'accès aux ressources des OC. Dans notre cas les OC se situent dans le réseau privé de l'utilisateur (son domicile, son bureau, son garage, etc.) Nous avons illustré notre approche via plusieurs cas d'usage originaux dans le domaine de la Télésanté et qui sont présentés dans le chapitre suivant.

D.1.3 Cas d'usage

Dans cette section, une famille de cas d'usage originaux qui a été étudiée et implémentée dans le cadre de la Télésanté, est introduite. Cette famille présente un lien direct avec les services de santé à distance destinés aux patients en général et aux personnes dépendantes en particulier. Le cas d'usage le plus simple, présenté dans la figure D.2, consiste en une consultation médicale d'un patient auprès de son médecin. Le patient est supposé avoir accès à un ensemble d'objets connectés capturant certains de ses signes vitaux.

Le cas d'usage le plus simple, présenté dans la figure consiste en une consultation médicale d'un patient auprès de son médecin. Le patient est supposé avoir accès à un ensemble d'objets intelligents capturant certains de ses signes vitaux.

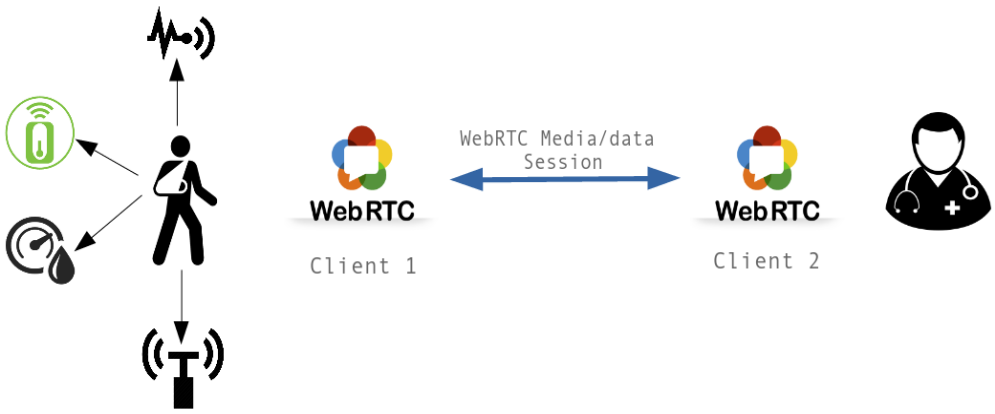


Figure D.2 – Téléconsultation

Dans ce cas, le patient peut communiquer en temps réel avec le médecin en audio/vidéo via WebRTC, parallèlement il peut transférer au médecin toutes les informations collectées à partir des capteurs physiologiques dont il est muni. Le médecin pourra ainsi évaluer l'état de santé du patient en se basant à la fois sur les échanges réalisés durant la communication audio/vidéo et en même temps en analysant les données physiologiques reçues en temps réel.

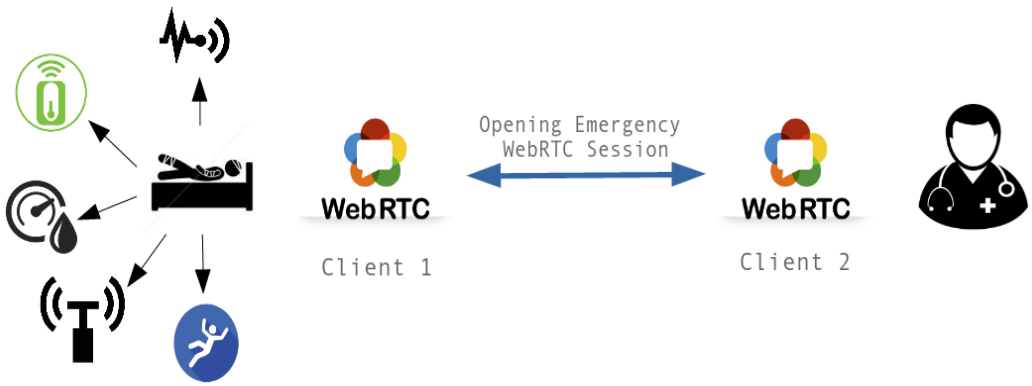


Figure D.3 – Télésurveillance

Le deuxième cas d'usage est celui de la surveillance continue de l'état de santé d'un patient comme le montre la figure D.3. Ce cas d'usage concerne principalement des personnes dépendantes ainsi que des personnes âgées ou les personnes handicapées : les capteurs utilisés sont attachés en permanence au corps du patient, et envoient en continue des données médicales à un centre médical. En cas de détection d'anomalie, par exemple en cas de chute d'une personne âgée (événement qui peut être déduit suivant l'analyse des données provenant d'un accéléromètre), une notification va être envoyée en urgence au centre médical.

Ces deux cas d'usages se positionnent principalement dans le cadre des services de santé qui peuvent être fournis au domicile du patient. D'autres cas d'usage sont détaillés dans [1].

D.1.4 Sécurité et respect de la vie privée

La sécurisation de cette architecture a été mise en œuvre afin de protéger les utilisateurs et leurs données contre les différentes menaces. Cette sécurisation permet de fournir les propriétés de sécurité suivantes:

La confidentialité et l'intégrité:s'obtiennent en chiffrant tout le trafic, d'abord entre la passerelle et le client WebRTC via CoAP-DTLS ou HTTPS, puis entre les deux clients WebRTC en utilisant SRTP pour le trafic multimédia et SCTP-DTLS pour les données [2].

L'authentification: s'appuie sur un fournisseur d'identité afin de prouver l'identité de celui qui veut accéder aux objets connectés [2].

Finalement, **le contrôle d'accès**: est basé sur RBAC (Role Based Access Control), afin de garantir que seules les personnes autorisées auront le droit d'accéder aux ressources des objets connectés. Ce contrôle d'accès est situé au niveau de la passerelle et utilise un PDP (Policy Décision Point) qui autorise ou interdit les demande d'accès [3].

De plus pour que les utilisateurs puissent accéder à ce type de services à distance, ils ont besoin de fournir et d'échanger des données à caractère personnel, en particulier des données relatives à leur santé. Ces données peuvent être ainsi exposées à la divulgation ou à la violation de la vie privée. Une attention particulière a donc été accordée au respect de la vie privée dans l'architecture proposée, notamment dans le cas d'usage de l'e-santé. Pour y parvenir il a été nécessaire de comprendre les lacunes en matière de protection de la vie privée ainsi que tout ce qui concerne la protection des données personnelles, en particulier celles relatives à la santé. En outre, une analyse des risques, a été réalisée accompagnée de l'identification de leurs origines et la proposition des contre-mesures possibles. Cette analyse a été faite via un PIA (Privacy Impact Assesement) en conformité avec le RGPD (Règlement Général sur la protection des Données) [4].

D.2 Vers la prochaine génération de maisons intelligentes

D.2.1 Résumé

La deuxième contribution propose l'intégration à une maison connectée, des services de santé à distance vus précédemment. L'idée principale est de permettre des interactions à distance entre l'utilisateur à l'intérieur de sa maison intelligente et un service médical, qui peut être un médecin par exemple. D'où la possibilité d'avoir des services de télémédecine à domicile tels que la téléconsultation et la télésurveillance. De plus, l'ensemble des travaux réalisés dans cette contribution cible le développement d'une véritable maison connectée, localisée dans les laboratoires de l'Université d'Aalborg, où en plus de la gestion des services de santé, un intérêt est également porté au système d'énergie préfigurant ainsi un premier pas vers la prochaine génération de maison connectée.

D.2.2 Architecture

Notre approche de la maison intelligente vise la possibilité, dans une certaine mesure, d'interagir avec l'utilisateur de manière indépendante et facile, afin d'améliorer la qualité de vie des résidents. Dans cette contribution, un accent particulier a été mis sur les services de santé à domicile, ainsi que sur la gestion et la rationalisation de la consommation d'énergie à l'intérieur de la maison connectée. Ce travail représente des expériences réelles réalisées dans une maison connectée à l'Université d'Aalborg, au Danemark. Ainsi, l'architecture proposée est composée de quatre couches, figure D.4:

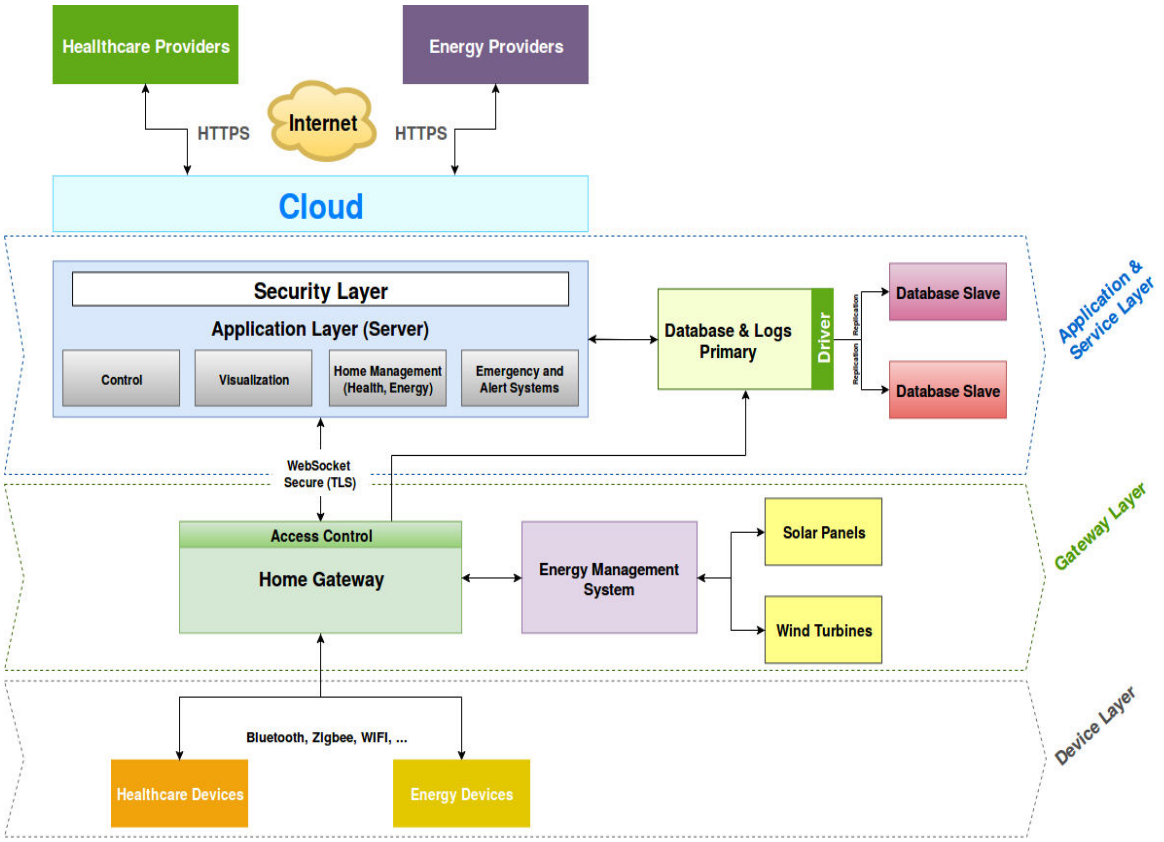


Figure D.4 – Architecture de la maison connectée

La couche des terminaux permet d'accéder aux différents objets connectés, via différents protocoles de communication concernant les différents services de gestion : santé, énergie ou compteur communicant.

La couche passerelle: permet de collecter toutes les données via une passerelle et un concentrateur de données (qui pourra être fusionné avec la passerelle dans les prochaines versions de l'architecture). Cette passerelle permet de communiquer avec des capteurs hétérogènes en implémentant un ensemble de piles protocolaires de communication.

La couche applicatif: permet de fournir plusieurs services avec les données collectées, telle la visualisation de ces données d'une manière simplifiée et compréhensible, les services de télémédecine (contribution 1), des services de gestion d'énergie (afin de permettre un meilleur contrôle de la consommation à coût bas), ainsi qu'un système d'alerte en cas de détection d'anomalie (I.e. détection de chute).

Un des buts de cette architecture est de converger vers la notion de la ville connectée. L'articulation de la maison connectée avec des fournisseurs de services externes, notamment ceux concernant la santé (pour la télémédecine) et ceux de l'énergie (I.e. information en temps réel des coûts de l'électricité) rendra cette approche possible.

D.3 L'accès sécurisé a plusieurs espaces connectés en utilisant SDN

D.3.1 Résumé

La troisième contribution introduit un contrôleur SDN pour gérer différents SSs pleuvant être impliqués dans une session WebRTC. L'idée principale consiste à permettre à un utilisateur de posséder plus d'un SS tout en conservant une gestion simple et efficace. Le principe de notre approche consiste à centraliser les décisions concernant la gestion des différents SSs. Étant donné que les préoccupations en matière de routage sont intimement liées à celles de la sécurité, le contrôleur SDN apparait clairement comme un outil prometteur pour résoudre ces problèmes. Nous avons développé sur cette base une architecture originale permettant à un utilisateur de gérer dynamiquement et de manière efficace et simple l'accès à ses différents SSs. Un prototype illustrant notre approche a été mis en œuvre avec un ensemble d'expérimentations afin d'évaluer la performance et la sécurité du système. Cette approche est finalement illustrée dans le domaine de la santé en démontrant la possibilité de gérer une infrastructure de santé telle qu'un hôpital.

D.3.2 Architecture

Il s'agit d'avoir un contrôle centralisé de la sécurité sur les différentes SS, et de permettre aux utilisateurs d'y accéder à tout moment et en tout lieu. La sécurité du système commence par les contrôles de sécurité de bas niveau en filtrant et en contrôlant tous les flux qui passent par les pare-feu et en modifiant dynamiquement les règles du pare-feu en fonction des besoins. Ensuite, en authentifiant les utilisateurs au niveau applicatif. Puis par la capacité de chiffrer le tout le trafic entre les utilisateurs et la ressource demandée dans le SS. Les clés de sécurité doivent ainsi être remises aux parties qui communiquent. Et finalement pour pouvoir contrôler les accès au différentes SSs d'une manière granuleuse.

La figure D.5 donne une vue d'ensemble de l'architecture théorique. L'architecture peut être divisée en deux parties principales: les entités qui interagissent avec l'interface nord du contrôleur. Et les entités qui interagissent avec l'interface sud du contrôleur.

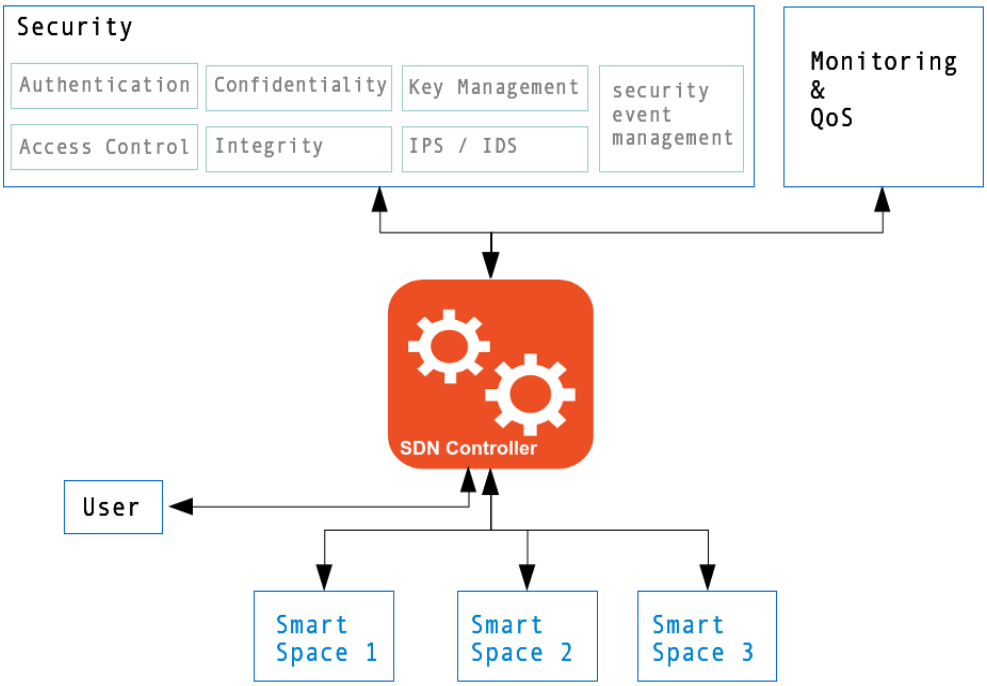


Figure D.5 – Architecture générale via SDN

La première catégorie représente les différents mécanismes complémentaires aux composants de base du SDN. Ils ajoutent principalement des mécanismes de sécurité, de surveillance et de qualité de service au système.

Les mécanismes de sécurité sont composés d'un tiers authentifiant, d'un contrôle d'accès et d'une PKI pour gérer les clés de chiffrement. Les mécanismes de surveillance pour contrôler le trafic et déployer éventuellement des mesures de détection supplémentaires, tels que un IDS, un IPS, SEM, etc.

Les mécanismes de QoS permettent de gérer la QoS des différentes SS.

Les principaux mécanismes de sécurité que nous avons proposés sont :

1. L'authentification: soit en utilisant une entité externe telle que les fournisseurs d'identité (SAML, OpenID Connect, etc.), soit en utilisant une simple authentification interne.

2. Le contrôle d'accès: afin de gérer l'accès aux différentes SS avec une granularité fine et contrer tous type d'accès non autorisé aux ressources des objets connectés que ça soit des utilisateurs illégitimes, ou des utilisateurs légitimes mais avec des intentions malicieuses.

3. Une PKI (Public Key Infrastructure): afin d'approvisionner les différentes parties communicantes avec des clés cryptographiques, afin de garantir, la confidentialité et l'intégrité des échanges [5].

D.4 References

[1] El Jaouhari S., Bouabdallah A., Bonnin JM., Lemlouma T. (2017) Toward a Smart Health-Care Architecture Using WebRTC and WoT,. WorldCIST 2017, Porto Santo, Portugal, in Advances in Intelligent Systems and Computing, vol 571.

[2] S. E. Jaouhari, A. Bouabdallah, J. M. Bonnin and T. Lemlouma, "Securing the Communications in a WoT/WebRTC-based Smart Healthcare Architecture," 2017 14th International Symposium on Pervasive Systems, Algorithms and Networks & 2017 11th International Conference on Frontier of Computer Science and Technology & 2017 Third International Symposium of Creative Computing (ISPAN- FCST-ISCC), Exeter, UK, 2017, pp. 403-408.

[3] S. E. Jaouhari, A. Bouabdallah and J. M. Bonnin, "A secure WebRTC/WoT- based health-care architecture enhanced with access control," 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 2018, pp. 182-187.

[4] S. El Jaouhari and A. Bouabdallah, "A Privacy Safeguard Framework for a WebRTC/WoT-Based Healthcare Architecture," The 13th IEEE Int. Workshop on Security, Trust, and Privacy for Software Applications (STPSA), in the IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 2018, pp. 468-473.

[5] S. El Jaouhari and A. Bouabdallah, "Dynamic security management of smart WoT infrastructures using SDN," 2018 IEEE 88th Vehicular Technology (VTC), accepted, 2730 August 2018 in Chicago, USA.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Conception sécurisée de services IoT pour les villes connectées

Mots clés : WebRTC, WoT, Sécurité, Respect de la vie privée, SDN, Maison intelligente, Santé numérique.

Résumé : WebRTC est une technologie récente de communication qui permet d'établir des échanges multimédia conversationnels directement entre navigateurs. Nous nous intéressons dans cette thèse à des locuteurs dans un *Smart Space* (SS) défini comme un environnement centré-utilisateur instrumenté par un ensemble de capteurs et d'actionneurs connectés. Nous analysons les capacités nécessaires pour permettre à un participant d'une session WebRTC d'impliquer dans cette même session, les flux induits par les objets connectés appartenant au SS d'un utilisateur quelconque de la session. Cette approche recèle un gisement de nombreux nouveaux usages. Nous limitons notre analyse à ceux concernant l'exercice distant d'une expertise et d'un savoir-faire. Techniquement, il s'agit d'articuler de façon contrôlée WebRTC et IoT/WoT. Nous procédons à une extension de WebRTC par WoT pour fournir à tout utilisateur d'une session WebRTC, un accès aux objets connectés du SS de tout autre participant à la session, en mettant l'accent sur la sécurisation de cet accès ainsi que sur sa conformité aux exigences de respect de la vie privée (RGPD) de l'utilisateur concerné. Le positionnement de notre approche dans le contexte des services de communication opérant dans les villes connectées, impose la prise en compte de SSs multiples et variés induisant chacun ses propres politiques de routage et de sécurité. Pour répondre à nos objectifs, il devient nécessaire au cours d'une session WebRTC, d'identifier, sélectionner, déployer et appliquer les règles de routage et de sécurité de façon à garantir un accès rapide et sécurisé aux différents SSs concernés et distribués sur tout le réseau. Nous développons une architecture originale répondant à ces besoins et intégrant un contrôleur SDN du fait de l'étroite imbrication entre les problématiques de routage et de sécurité. Un prototype illustrant notre approche a été mis en œuvre et testé afin d'évaluer la performance et la sécurité du système. Nous illustrons finalement notre approche dans le domaine de la santé en démontrant son apport pour gérer une infrastructure de grande taille telle qu'un hôpital.

Title : A secure design of WoT services for smart cities

Keywords : WebRTC, WoT, Security, Privacy, SDN, Smart home, e-Health.

Abstract: The richness and the versatility of WebRTC, a new peer-to-peer, real-time and browser-based communication technology, allowed the imagination of new and innovative services. In this thesis, we analyzed the capabilities required to allow a participant in a WebRTC session to access the smart Things belonging to his own environment as well as those of any other participant in the same session. The access to such environment, which we call "Smart Space (SS)", can be either passive, for example by monitoring the contextual information provided by the sensors, or active by requesting the execution of commands by the actuators, or a mixture of both. This approach deserves attention because it allows solving in an original way various issues such as allowing experts to remotely exercise and provide their expertise and/or knowing how. From a technical point of view the issue is not trivial because it requires a smooth and mastered articulation between two different technologies: WebRTC and the Internet of Things (IoT) / Web of Things (WoT). Hence, the first part of the problem studied in this thesis, consists in analyzing the possibilities of extending WebRTC capabilities with the WoT. So as to provide a secure and privacy-respectful access to the various smart objects located in the immediate environment of a participant to any other end-user involved in the same ongoing WebRTC session. This approach is then illustrated in the e-health domain and tested in a real smart home (a typical example of a smart space). Moreover, positioning our approach in the context of communication services operating in smart cities requires the ability to support a multiplicity of SSs, each with its own network and security policy. Hence, in order to allow a participant to access one of his own SSs or one of another participant (through a delegation of access process), it becomes necessary to dynamically identify, select, deploy, and enforce the SS's specific routing and security rules, so as to have an effective, fast and secure access. Therefore, the second part of the problem studied in this Ph.D. consists in defining an efficient management of the routing and security issues regarding the possibility of having multiple SSs distributed over the entire network.