Francesco Marino

Towards a Softwarized and Secure Internet of Things

Sant'Anna
Scuola Universitaria Superiore Pisa

Academic Year 2017/2018
Ph.D. Course in
Emerging Digital Technologies

Curriculum:
Embedded Systems

Towards a Softwarized and Secure Internet of Things

Author
Francesco Marino

Supervisor
Prof. Piero Castoldi

Tutor
Ing. Barbara Martini

Abstract

The Internet-of-Things (IoT) paradigm envisions the full integration of several technologies in order to enable a new class of applications relying on an unprecedented sensing and actuating infrastructure. However, traditional IoT applications are static, closed, vertically integrated solutions which do not allow to fully exploit the potentialities of this new technological platform. For this reason IoT systems are now evolving towards more open and dynamic architectures and solutions in which dynamicity is considered an added value to build context-aware and self-adapting applications. In this new continuous changing scenario, solutions for *automatic contract negotiation and management*, *service discovery* and *mutual authentication* will play a central role, since they can be considered the main building blocks to create secure and dynamic applications. The thesis contributes to this context by surveying research activities in such fields, pointing out the challenges and the open research issues that arise in such a new emerging scenario. The thesis also gives a 5G-oriented perspective of the outlined evolution of IoT systems, which will be part of the highly heterogeneous, programmable and pervasive 5G network infrastructure, and as such, will be involved in the dynamic allocation and orchestration of resources. In this regard the thesis details the techno-economic drivers which are leading to the softwarization of networks and how this will impact 5G networks management and service modeling. To cope with this new scenario the thesis presents the concept of an Operating System for managing 5G network infrastructures, and details its functionalities and building blocks.

Following the above insights, the thesis elaborates and presents technical solutions enabling a softwarized and secure Internet of Things.

The thesis describes a softwerized IoT system to monitor urban mobility in real time. The proposed system builds on a network of visual sensor nodes running a specifically developed middleware which enables in-network event composition and remote node reconfiguration at runtime. This is achieved by means of a management module which allows to dynamically allocate, manage and tune high-level tasks on IoT nodes by adopting a virtual machine approach. In order to make such component leverageable as a means to incorporate highly-programmable IoT networks into the 5G ecosystem, the thesis also proposes its integration into the standard oneM2M architecture.

With regard to security, the thesis proposes two approaches based on secure two-party computation protocols to preserve the privacy of users immersed in smart environments. In more detail, the first approach addresses usage control systems working in synergy with smart cameras to monitor users behaviour. In this context the proposed approach allows to disclose to the system only the users' attributes which are relevant for the evaluation of usage control policies, while keeping secret users' identities as long as they behave correctly. The second approach focuses on geolocation-aware applications running on mobile devices. Here the proposed technique allows to limit information leaks about the geolocation of users who use mobile applications requiring access to users' position to deliver the intended services. In conclusion, the thesis proposes an

architecture to support certificate-based mutual authentication in constrained IoT scenarios. The proposed architecture allows IoT nodes to flexibly delegate to powerful nodes the most burdensome cryptographic tasks involved in certificate validation and to reduce bandwidth consumption by introducing a new type of compact certificate. The thesis also highlights the role that the proposed approach can play in the context of the oneM2M security framework in enabling standard secure machine-2-machine interactions.

TABLE OF CONTENTS

List of Tables

List of Figures

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Thesis reference scenario

Distributed systems are going through a profound process of change which can be observed in several trends, such as: the ever deepening differentiation of service elements, i.e. the increasingly marked boundaries between the application logics, service components, and execution infrastructures domains; the increasing heterogeneity of the execution infrastructure, ranging from the Cloud to the Fog and passing through the Geo-distributed Cloud and the Edge-cloud, that is leading to the coexistence of several infrastructures which are greatly diversified in terms of technology and decentralized in terms of administration; the scattered allocation of the service elements, no longer hosted on PC and Data Centers only, but also by mobile personal devices, smart objects, household appliances, etc. The distributed systems which arise at the intersection of those evolutionary paths are characterized by a high level of decentralization, with no single control points and no global states, and a high level of dynamicity, due to the mobility of the involved entities and the resulting ever evolving execution contexts where assuring the continuity of the delivered services is not a trivial task.

A distributed system paradigm which has in essence the natural tendency to evolve according to the outlined trends is the Internet of Things (IoT). Indeed the IoT scenario involves highly heterogeneous smart objects, i.e. devices possessing very different computational and energetic capabilities which are deployed in the environment in order to provide low-latency and context-aware services to users. However, the IoT vision is still far from being fully achieved. Classic IoT applications are vertical silos, very static both in terms of offered functionalities and in the nodes participating to their implementation, that must be known at design time. Such systems are targeted to very specific tasks, and they are isolated, not requesting nor offering services

from the outside.

Nowadays, however, IoT systems are striving to evolve towards an open model in order to improve their own flexibility and efficiency. In this new scenario they would be allowed to interact with each other, enabling this way horizontal applications based on dynamically established relations between nodes. So, besides being service providers, nodes would also become consumers of services provided by other nodes, conveniently selected and orchestrated to offer higher-level services which could be even completely different from the ones envisaged at deployment time.

1.1.1 Challenges in the reference scenario

In the realization of the envisioned scenario there are a number of enabling technologies which are expected to play a major role, namely automatic contract management, service discovery and security technologies.

Indeed, contracts will be of utmost importance in a context where service providers and consumers need to frequently establish, change and interrupt business relationships. However their definition, negotiation, and monitoring raise several issues in the considered scenario. Those tasks in fact are in fact quite demanding for IoT systems, especially for their constrained segments.

Service discovery solutions, on the other hand, need to be leveraged to guide the selection of the most suitable service providers to start a negotiation with, depending on the characteristics of the desired services. The challenges that arise in this context are introduced by the necessity of reaching a trade-off between the overheads introduced by service discovery architectures and the freshness and completeness of the information they deliver. Moreover, the openness of the considered scenario enables interactions between entities belonging to different administration domains, and this may lead to semantic disalignment of services descriptions resulting in ineffective service selection procedures. In the end it is worth noting that, since the interactions in the reference scenarios will not occur between humans but between an increasing number of autonomous machines, the whole service discovery, negotiation and monitoring process should be fully automated.

In the end, security solutions will provide a fundamental, transversal substratum for a wide range of technologies, including the aforementioned ones. In fact, mutual authentication must be assured between the entities signing a contract in order to make possible the enforcement of the related obligations, i.e. fees and penalties. On the other hand, the implementation of confidentiality and privacy preserving mechanisms will be paramount to enable secure and reliable contract management and service discovery systems, and make people accept the deployment of pervasive IoT technologies in the environments they live. Also in this regard, overheads caused by security technologies need to be addressed.

It is clear that service discovery, negotiation and security will be paramount for the fully incorporate IoT systems into the 5G platform. To achieve this, however, it will be important to

bring the softwarization trend down to the IoT segment of the network. Moreover, the design and adoption of standard IoT architectures will be required to automate the integration of the most disparate IoT systems into the all-encompassing 5G framework.

1.2 Thesis contributions

The main contributions of the thesis can be summarized as follows:

- **Discussion of the trends which are affecting nowadays IoT ecosystems, and identification and analysis of the enabling technologies of tomorrow's IoT ecosystems:** the thesis depicts a comprehensive picture of the transformations IoT systems are going through. The requirements of the IoT systems which arise at the culmination of the pinpointed evolution processes are analyzed and, on this basis, their enabling technologies are identified and extensively discussed. The discussion includes a broad survey of the state of the art of the enabling technologies and a reasoned analysis of the gaps that need to be filled to allow their employment in the IoT context;

- **Modelling of a 5G Operating System (5G OS) for enabling management, control and orchestration of future softwarized 5G infrastructures including IoT systems:** the thesis deals with the evolution of IoT systems also from a 5G point of view. In this respect, it elaborates concepts and modelling of an Operating System as a strategic platform to enable automated and efficient management of 5G resources, which will include more and more softwarized IoT infrastructures. The 5G OS will also act as an abstraction layer allowing users to easily access virtualized resources which could be allocated and orchestrated even on IoT nodes;

- **Proposal of a softwarized IoT system addressing a Smart City scenario, and its integration in the standard oneM2M architecture:** the thesis presents an IoT-based solution to monitor urban mobility in real time. IoT nodes, embedding vision sensors and computer vision logics, share their perceptions and build a global and comprehensive interpretation of the analyzed scenes in a cooperative and adaptive fashion. To this end nodes run a specifically developed middleware enabling in-network event composition and remote node reconfiguration at runtime. The middleware includes a management module which allows to dynamically allocate, manage and tune high-level tasks on IoT nodes. In order to make such component leverageable as a means to incorporate highly-programmable IoT networks into the 5G ecosystem, the thesis also proposes its integration into the oneM2M architecture;

- **Proposal of privacy-preserving techniques for smart cameras based video surveillance systems and geolocation-aware mobile applications:** the thesis presents two

approaches to preserve the privacy of users immersed in smart environments. The first approach addresses usage control systems working in synergy with smart cameras to monitor users behaviour. In this context the proposed technique allows to disclose to the system only the users' attributes which are relevant for the evaluation of usage control policies, while keeping secret users' identities as long as they act correctly. The second approach focuses on geolocation-aware applications running on mobile devices. Here the proposed technique allows to limit information leaks about the geolocation of users who use mobile applications requiring access to users' position to deliver the intended services. Both approaches rely on secure two-party computation protocols;

- **Proposal of an architecture to support certificate-based mutual authentication in constrained IoT scenarios, and its integration in the standard oneM2M architecture:** the thesis presents an approach to outsource the most burdensome tasks involved in certificate-based mutual authentication procedures, which often are unaffordable for constrained IoT nodes. The proposed approach allows nodes to flexibly select the tasks to outsource depending on their current execution context, and does not require both authenticating parties to rely on it, allowing this way interoperability with already deployed systems. The thesis also proposes an oneM2M compliant implementation of the presented approach.

1.3 Thesis organization

The thesis mainly addresses the enabling technologies of the forthcoming IoT scenarios, and in particular it discusses approaches paving the way to softwarized and secure IoT ecosystems. The thesis is divided in three sections: the first one, including chapters 2 and 3, outlines and analyzes characteristics, trends, requirements and enabling technologies of the emerging IoT scenarios; the second one, including chapters 4 and 5, presents IoT management solutions bringing the network softwarizion trend down to IoT nodes; the third and last one, including chapters 6, 7 and 8, reports approaches to provide security and privacy in smart environments.

Chapter 2 analyzes the current evolution of nowadays IoT systems, highlights their emerging characteristics and requirements, and identifies the technologies that need to be leveraged to meet such requirements, i.e. contract negotiation and management, service discovery and mutual authentication technologies. Then the chapter surveys research activities in those fields and points out the challenges and the open research issues that arise as a result of their employment in such a new emerging scenario.

Chapter 3 inserts in a 5G-oriented perspective the outlined evolution of IoT systems, which will be part of the highly heterogeneous, programmable and pervasive 5G network infrastructure, and as such, will be involved in the dynamic allocation and orchestration of resources. In this regard the chapter details the techno-economic drivers which are leading to the softwarization

of networks and how this will impact 5G networks management and service modeling. To cope with this new scenario the chapter presents the concept of an Operating System for managing 5G network infrastructures, and details its functionalities and building blocks.

Chapter 4 presents a softwarized visual sensor network in which each node embeds computer vision logics for analyzing in real time urban traffic. Softwarization is achieved by means of a dedicated middleware component, T-Res, which adopts a virtual machine approach to upload and execute high-level, self-contained tasks expressed as Python scripts. In the considered context T-Res has been used to enable the dynamic allocation of computer vision algorithms and their subsequent tuning, but it general it can be successfully leveraged to provide IoT networks with the required level of programmability to fully integrate them in the 5G scenario. To this end, chapter 5 proposes the integration of the T-Res component into the standard oneM2M architecture. The main idea is to hide the T-Res interface behind the standard oneM2M interface, and make all control flows pass through the latter. Then it is oneM2M job, through the specifically defined commands, to perform the proper operations on the T-Res interface. This way T-Res potential can be conveniently exploited to implement oneM2M Common Service Functions (CSFs) such as the Application and Service Layer Management CSF and the Device Management CSF.

Chapters 6 to 8 deal with security and privacy in smart environments. In more detail, chapter 6 present a usage contol framework relying on a video surveillance system based on smart cameras to monitor users' behaviour. In order to protect users' privacy, the chapter proposes a privacy-preserving face-matching protocol carried out by the smart cameras deployed in the environment, continuously capturing users' faces, and the centralized usage control framework, which holds a database containing information about all the users, including a photo of them. The protocol allows the usage control framework to learn only the strictly necessary information about the users, while not revealing their identities. Chapter 7, on the other hand, presents a location-based privacy-preserving information service for mobile users. It allows service providers to deliver their services to users depending on their position, without actually learning their position. To this end, also in this case a privacy-preserving protocol has been designed to securely determine if there is an overlap between the areas into which service providers want to deliver their services and users' locations. In the end, chapter 8 tries to answer to the challenges related to mutual authentication highlighted in chapter 2 by proposing an approach to support certificate-based authentication in constrained IoT environments. The presented solution allows IoT nodes to flexibly delegate to powerful nodes the most demanding cryptographic tasks involved in certificate validation and to reduce bandwidth consumption by introducing a new type of compact certificate. The chapter also highlights the role that the proposed approach can play in the context of the oneM2M security framework in enabling standard secure machine-2-machine interactions.

Finally, chapter 9 concludes the thesis.

Part I

Decentralized, dynamic and open IoT ecosystems

CHAPTER 2

ENABLING TECHNOLOGIES AND TECHNOLOGICAL CHALLENGES OF THE FORTHCOMING IOT ECOSYSTEMS

2.1 Introduction

The term Internet of Things (IoT) refers to a scenario where the objects that surround people in their everyday lives have communication, computational, sensing and actuating capabilities. Such "smart objects", as they are usually called, are therefore able to communicate with each other by means of wireless technologies in order to provide high-value and cooperative services to people and other smart objects.

Nowadays the classic IoT scenario envisioned and considered in a large number of applications is static and closed. Smart objects contributing to the implementation of an application are a-priori known, as well as their features, the way they interact with each other, and the services they provide. This system architecture does not allow to fully exploit the enormous potential that such an ubiquitous and multifunctional ecosystem can offer. If each smart object or application was enabled to freely interact with smartphones, connected cars, cloud services and any other type of smart object that passes nearby, and to understand who they are, what they do, what they want, how they pay, this would have a disruptive impact on the emergence of new applications and services. Smart objects and applications may in fact autonomously reach agreements and organize themselves for shorter or longer periods in order to provide new functionalities that possibly had not been imagined when they have been deployed. The features of this new dynamic and decentralized scenario in terms of flexibility, efficient use of resources and adaptability make it very appealing, and this is why classic IoT systems are now evolving towards this model. In the evolution of the described scenario, depicted in Fig. 2.1 with some possible interactions among devices, several aspects must be taken into account. Issues related to contract management

(definition, automatic negotiation, and monitoring), service discovery and mutual authentication must be considered of utmost importance to enable the provisioning of dynamic services. Over the years the research community has produced a great number of survey papers detailing the IoT philosophy, its enabling technologies, trends and open issues from different points of view. A few remarkable examples are [1], where the authors identify different flavours of the IoT paradigm, namely the Things-oriented, Internet-oriented and Semantic-oriented visions, and present their enabling technologies along with an extensive list of applications requiring their organic integration and cooperation; [2], which proposes a service oriented architecture for the IoT and identifies the requirements and the enabling technologies of each service layer; [3], which has a special focus on the enabling standards and protocols for the IoT, including their relevant Quality-of-Service criteria and their integration in the Cloud and Fog computing paradigms; [4], mainly investigating technologies for a Cloud centric IoT. Several works also exist which focus on specific IoT topics, like security [5], trust [6] and cryptographic protocols [7], data mining [8], data-centric [9] and context-aware computing [10], middlewares [11], and operating systems [12]. However, a work surveying in an exhaustive and organic manner the technologies of automatic contract negotiation and management, service discovery and mutual authentication in a decentralized, dynamic and open IoT context is still missing.

This chapter surveys the main solutions related to these topics by providing a comprehensive analysis in the considered scenario. It is organized as follows. Section 2.2 analyzes the characteristics and requirements of this new scenario and identifies its enabling technologies. Sections 2.3, 2.4 and 2.5 report the solutions which are currently used in the web, wireless sensor networks, and already established IoT contexts for automatic contract negotiation and management, service discovery and mutual authentication. The order in which they are presented reflect the logical positions that, in our opinion, they will take in the technological stack that is needed to enable the new dynamic and decentralized IoT scenarios. So automatic negotiation, that is the key element in this context, stands on top of service discovery and mutual authentication technologies, which provide support to automatic negotiation and to a large amount of other technologies as well. Section 2.6 summarizes the challenges that arise when using the considered technologies in the emerging scenario and points out the issues which require a further research effort by the community. Section 2.7 concludes the chapter.

2.2 Emerging IoT scenarios: requirements and enabling technologies

A newly rising generation of applications is characterized by a greater degree of integration of the digital realm with the physical world. This trend is due to the progressive deployment in the environment of smart objects with sensing and actuating capabilities that enable the development of the so-called cyber-physical systems. Examples of application scenarios range

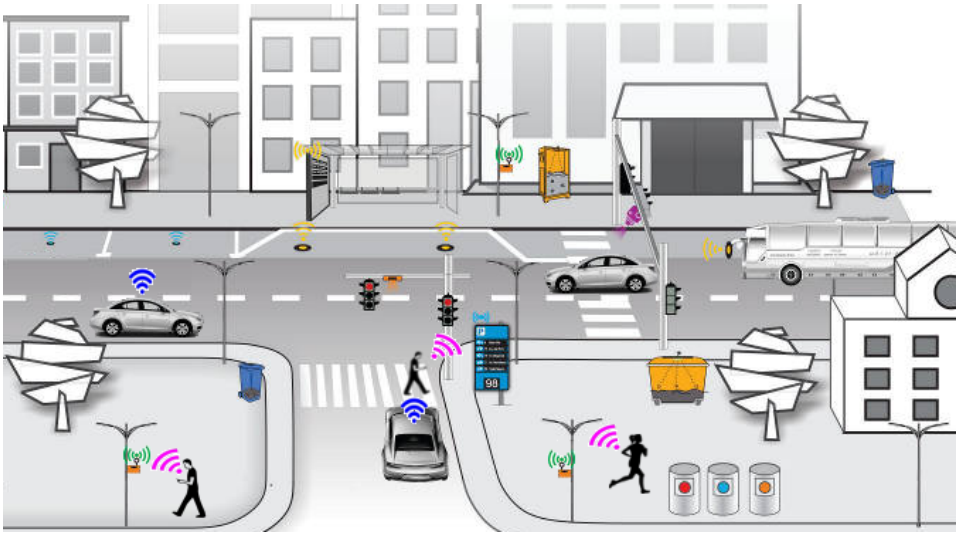


Figure 2.1: The reference scenario: heterogeneous smart objects deployed in the environment to build decentralized and dynamic services at run-time.

from Smart Cities to Smart Home, from Industry 4.0 to infrastructure monitoring, from e-Health to automotive. Moreover, the diffusion of personal mobile devices (e.g., smartphones or tablets) and the adoption of wearable devices increment the possibility for the people to interact in a continuous way with these cyber-physical systems. In this way the activities performed by a person can be seamlessly augmented by means of digital applications.

These emerging scenarios can be seen as an evolution of the IoT ones. From a higher point of view, they could be abstracted as a sort of "service oriented computing", where the services can be provided either by data centers, or personal devices, or smart objects, etc. while the same types of systems can run applications that retrieve, invoke and orchestrate these services according to the execution context needs.

Indeed the implementation of these scenarios must also take into account the evolution of the execution infrastructure. In this context the X-as-a-Service model and Application Programming Interfaces (APIs) are expected to play a central role. X-as-a-Service refers to the delivery of a broad range of technologies and functionalities as services over the internet. APIs, in the other hand, represent a very convenient means through which deliver such services since they act as entry points enabling flexible and regulated communications between distributed software components. The adoption of cloud-based solutions, with the access to virtualized resources according to the Infrastructure as a Service (IaaS), Platform as a Service (PaaS) or Software as a Service (SaaS) models can bring several key benefits, such as cost reduction and optimization, lower entrance barrier, and innovation acceleration. In particular, SaaS can be fruitfully implemented according to the API paradigm in order to improve the development of applications able to dynamically and flexibly combine services, potentially provided also by 3rd party actors.

Finally, the deployment of these cyber-physical application scenarios should consider the

advantages derived from the availability of Cloud infrastructures widespread in the environment, implementing the so-called Fog Computing. Fog Computing is a horizontal architecture which relies on heterogeneous, ubiquitous and decentralized edge devices to carry out storage and processing tasks without the intervention of third-parties. In particular, Fog Computing enables to bring near the devices in the environment (e.g., personal devices, wearable devices, smart objects, etc.) the resources and services offered according to the IaaS, PaaS, and SaaS approach, in order to:

- guarantee predictable latencies, compatible with the requirements of (near-)real-time applications;
- efficiently provide locality-specific services to support "geographical-aware" applications;
- support "glocal" applications, i.e., geographically distributed applications, but with the need of performing execution characterized by a local context (e.g., applications for controlling geographical distributed infrastructures);
- optimize the use of communication resources;
- improve the control on personal data usage (e.g., personal data sharing or diffusion, according to privacy requirements).

A representative example in this scenario is given by a connected car which arrives in a new city and have to interact with the local service providers in order to offer to the driver the necessary functionalities to make his car trip enjoyable. As an example, the driver probably wishes to find a parking spot which is near to his current position and at a reasonable price. The car can therefore negotiate with the city transportation company or with other parking lot managers on parameters such as the position, the price, the type of vehicle and the length of stay so that the proposal which guarantees the most satisfying trade-off is found and an agreement is reached. The connected car, on the other hand, is able to provide services which could be valuable, for example, to the city tourism office. The car can in fact disclose information, such as the city of departure or the followed route, which could be used, e.g., to build useful statistics for the development of a strategic plan aimed at increasing the number of tourists coming from certain zones. It is worth noting that, when the car enters the city for the first time, it has never met the parking lot providers before and it does not even know how to contact them. The same applies from the city tourism office perspective. Mechanisms which allow to find the suitable service provider and to be sure that the service provider (and also the service consumer) is who he claims to be, along with negotiation techniques, are therefore the necessary technological substratum for the class of applications of which the above example is just an instance.

2.2.1 Requirements and enabling technologies

In these scenarios some challenging requirements must be considered, namely, openness, horizontality, decentralization and dynamicity. All these deeply impact on the establishment of relationships between the applications and the orchestrated services. These requirements are mainly due to the need to support the mobility of devices executing the applications (e.g., a smartphone or an automotive board) and that of smart objects (e.g., the sensors provided by a smartphone, a connected car, or a wearable device, or tags associated to goods in a container). Therefore the implementations of these scenarios must be able to deal with:

- *openness*: the deployed applications and services could belong to different administrative domains. For instance, a smart city application running on a smartphone must be able to interact with the services delivered by smart objects deployed in several towns, without any previous knowledge or registration;
- *horizontality*: services should be decoupled from specific applications or application domains; a service could be invoked by applications potentially addressing different contexts;
- *decentralization*: applications and services run on decentralized systems and must be able to cooperate (e.g., for service discovery, authentication, negotiation, etc.) without a centralized point of control, due to technical constraints (e.g., latency, bandwidth) and to administrative ones (e.g., support to multiple domains);
- *dynamicity*: applications must be able to run in an evolving context, i.e., they must be able to retrieve and compose in a dynamic and flexible way the services and resources necessary to their execution, by taking into account constraints and requirements due to either mobility, resource availability, or invocation context.

The API model seems to become the most relevant paradigm to access services (from infrastructural to applicative ones) offered in a multi domain and decentralized context. The applications adopting such an API-based model to run in the previously described scenarios must be able to perform some critical operations to dynamically aggregate the services required for their execution, namely: *mutual authentication, service discovery and selection, and automatic service agreement negotiation and management.* These are in fact the technologies that, working in synergy, enable a scenario with the mentioned characteristics: service discovery and selection techniques allow to achieve horizontality and dynamicity, automatic service agreement negotiation and management techniques also provide support to decentralization, and mutual authentication technologies are fundamental to promote openness. Corresponding functions should be supported by the services or by the platforms offering them.

As a representative example in this respect, it is worth mentioning the TerraSwarm project [13]. The project aims at addressing the challenges of the depicted scenario by designing a

distributed operating system able to manage and orchestrate real and virtual resources hosted on cloud/edge/fog infrastructures in order to implement highly dynamic mobile applications. Interestingly enough, the proposed framework includes contract management, service discovery and security functionalities, which are therefore widely recognized as greatly important to fulfill the requirements of the envisioned scenario.

In the following the chapter analyzes state-of-the-art solutions addressing these functional areas by considering the previously highlighted requirements.

2.3 Contract management solutions

APIs are becoming one of the most important means of providing and selling digital services according to the X-as-a-Service paradigm. In this context APIs naturally enable the implementation of complex applications based on the composition of atomic services provided by actors which are heterogeneous in terms of resources, capabilities, and autonomy, with limited or without any involvement of people. The emerging IoT technologies perfectly fit into this scenario while providing enhancements, since they will constitute the technological platform which will make possible the delivery of a huge amount of valuable location-aware and timely services to the users. For instance, they enable the Fog Computing approach bringing closer to applications running on terminals and devices the services and the resources needed for their execution. In this context APIs providers, i.e. entities exposing their data or capabilities through APIs, and APIs consumers, i.e. entities wishing to get access to such data and capabilities, will operate in a more and more dynamic ecosystem, where they will need to establish, change and interrupt business relationships frequently. Examples are the already mentioned IoT scenario, where applications have to interact with the relevant sensors deployed in the environment, or the Smart City scenario, where connected cars have to rely on localized services. Such service ecosystems require mechanisms for coping with the dynamic negotiation, establishment, and monitoring of contracts which leverage on APIs to ensure the smooth running of the applications. Moreover, since the interactions in the reference scenarios will not occur between humans but between an increasing number of autonomous machines offering and providing services, the entire contract management process has to be fully automated [14]. The following subsections discuss technologies related to contracts definition, contracts negotiation and contracts monitoring respectively. The closing subsection focuses on payment mechanisms based on cryptocurrencies, as they take centre stage in contract management workflows and their underlying technology can play a role also in other relevant areas of the considered decentralized and open scenarios. A synthetic review is reported in Fig. 2.2.

	Contract Management						
	Definition	Negotiation architecture	Negotiation objects	Negotiation models	Monitoring	Security	Payment
[21]	x				x		
[22]	x				x		
[23]	x				x		
[24]	x				x		
[25]	x						
[32]		x					
[33]		x	x				
[35]		x			x	x	
[36]		x		x			
[37]		x					
[38]			x				
[39]			x				
[40]			x				
[41]			x				
[42]				x			
[43]					x		
[44]		x			x		
[45]					x	x	
[46]					x		
[47]					x		
[52]						x	x
[56]					x	x	x

Figure 2.2: Contract management solutions

2.3.1 Contract definition

The large adoption of the Service-Oriented Computing (SOC) in computer software design necessitated introducing means to allow the conclusion of contracts between services. In fact, when software elements become autonomic, adaptive and open, like in the SOC context, the implicit or explicit human-targeted static agreements that are used to hold in traditional vertically integrated computing systems are not valid anymore [15]. So, in order to make services interoperate correctly according to set of rules established at run-time, technological solutions have been proposed to automatically match service providers and service consumers depending on their characteristics and requirements, and to ratify the achieved agreements by means of binding contracts.

In [16] many definitions of contract in the context of SOC are reported (e.g., [17], [18], [19]), but in general they all agree on the fact that a contract is made up of technical and non-technical components. Technical components define the input and output parameters and the protocols supported by the service provider (WSDL Definition [20]), and the XML schema definition (XSD)

of the input and output parameters. Non-technical documents can include, among other things, Service Level Agreements (SLAs). SLAs define the Quality-of-Service (QoS) expected by the service consumer (in terms of cost, availability, latency, security, etc.), the metrics that will be used to verify whether the conditions of the contract are met and the remedies or penalties that will be applied if they are not. SLAs play a central role not only in regulating already ongoing business relationships, but also in supporting the selection of the best service provider by taking into account both functional and quality requirements.

While technical documents are statically defined by the service provider, SLAs are usually negotiable. They are often defined in plain text, using templates or toolkits. However, in order to enable automatic machine-to-machine negotiations, both technical and non-technical documents should be written in a machine-readable format. To this aim, different solutions have been proposed (e.g., [21], [22], [23]), but the de-facto standard for specifying and managing SLAs is the Web Service Agreement Specification (WS-Agreement) [24]. WS-Agreement is a specification developed by the Open Grid Forum which provides an XML-based language for the definition of agreements and a simple protocol for advertising the capabilities of service providers, negotiating agreements between service consumers and providers, and monitoring agreement compliance. All these functionalities are encapsulated in the *Agreement Management* component that WS-Agreement adds on top of the traditional service oriented architecture, which remains in charge of consuming and dynamically deploying and providing the services upon which the new upper layer has reached an agreement. To create an agreement, the *Agreement Initiator* (e.g., the Service Consumer) submits to the *Agreement Responder* (e.g., the Service Provider) an *Agreement Offer*. In order to make the *Agreement Initiator* able to understand which kind of agreement the *Agreement Responder* is willing to stipulate, the latter exposes a set of *Agreement Templates* that can be retrieved, manipulated and completed to build a valid offer and which can be leveraged during the service discovery phase. The format of the agreement templates, of the agreement offers and of the final agreements is the same, and it is defined in the WS-Agreement XML schema. It includes 3 structural components: the *Name*, an optional human-understandable description of the contract; the *Context*, which contains information about the involved parties, the services the agreement is related to and the duration of the agreement; the *Terms*, which represent the core part of the agreement. WS-Agreement defines two term types, *Service Description Terms*, and *Guarantee Terms*. *Service Description Terms* deal with the functional aspects of a service such as its interface description and endpoint references, whereas *Guarantee Terms* define the Service Level Objectives (SLO), i.e. the qualifying conditions upon which they apply and the possible penalties or rewards. As an example, a service level objective could be a bound on the average response rate, which must be met only when the qualifying condition of being in certain hours of the day holds. When an agreement is created, usually the Agreement Responder exposes the content of the agreement, its run-time state (pending, observed, rejected, completed) and the state of compliance with individual terms of the agreement (not determined, fulfilled, violated).

Among the so-called WS-* specifications, which include the already mentioned WSDL, XML and WS-Agreement, also falls SOAP (Simple Object Access Protocol) [25], a standard that in the past has been extensively used for the implementation of web services. In the last few years, however, RESTful web services [26] have gained popularity due to their performance, scalability, and flexibility. According to the REST paradigm both data and functionalities are represented as resources which can be accessed through Uniform Resource Identifiers (URIs) like Web links, and acted upon by using a small set of well-defined operations in conjunction with a stateless communication protocol such as HTTP. Precisely because of its less overheads, less parsing complexity, statelessness, and tighter integration with HTTP the REST architectural style has been considered the most suitable for constrained contexts such as the IoT scenario. In [27], for example, the author presents an architecture based on RESTful web services enabling end-to-end integration of constrained IoT devices with the Web. The proposed architecture leverages the Constrained Application Protocol (CoAP [28]), a lightweight implementation of HTTP featuring additional functionalities which are beneficial in the considered scenario, such as built-in service discovery and asynchronous message exchange mechanisms. The reduced requirements of such RESTful web transfer protocol are met also by constrained smart objects, which can therefore run it and perform end-to-end transactions with regular Internet nodes, and viceversa, according to the uniform RESTful paradigm. The transparent access to resources hosted on servers of another protocol is ensured by the proposed architecture through a cross-protocol proxy performing HTTP-CoAP mapping. The equivalent to WSDL for RESTful web services is WADL [29], but it is not officially standardized nor widely adopted. For this reason, since RESTful web services are not usually provided with a formal description of their interface, SOAP-based design appear to be more suitable when a formal contract has to be established. A RESTful implementation of the WS-Agreement specification, however, is presented in [30]. Here the authors mapped the WS-Agreement templates and agreements into REST resources and the operations that can be performed on them into HTTP methods, in order to provide a lightweight mechanism for SLAs management.

2.3.2 Contract negotiation

As already stated, the reference scenario will be populated by autonomous computing systems that will have to interact with each other in order to achieve the objectives they are designed for, which typically consist in the establishment of the most favorable business relationships according to certain metrics. The mechanism of automatic negotiation plays a central role in this scenario since it is through this process that autonomous agents, whether they are self-interested or cooperative, can reach mutually acceptable agreements. The development of a comprehensive and flexible negotiation architecture taking into account the heterogeneity and dinamicity of the involved entities.

In the past, researchers approached the topic of automated negotiation from three main

perspectives [31]: negotiation architectures and protocols, dealing with the set of rules that governs the interaction; negotiation objects, dealing with the definition of the terms over which an agreement must be reached and the types of operation that can be performed on such terms (change, addition, cancellation); agents decision-making models, dealing with the negotiation strategies the participants have to follow to reach their objectives.

In the field of negotiation protocols, the Web Service Agreement Negotiation protocol (WS-Agreement Negotiation) [32] is the standard de-facto. It was defined as an extension of the WS-Agreement specification when its basic negotiation protocol, the so-called "take-it-or-leave-it" protocol which only allows the service consumer to accept or reject the conditions proposed by the service provider without any further negotiation step, proved to be not enough for more complex scenarios. For this reason, WS-Agreement Negotiation adds a Negotiation Layer on top of the Agreement Layer specified by WS-Agreement. The Negotiation Layer defines a language and a protocol to negotiate an agreement. The negotiation process involves the exchange of offers and counteroffers, and once an agreement is reached, the Negotiation Layer relies on the lower layers to deal with contract creation, management and monitoring.

Several architectural solutions leveraging on cloud/fog computing have been proposed to deal with contracts management and negotiation in the considered scenario.

In [33] the negotiation component of the Building the Environment for the Things as a Service (BETaaS) platform is presented. BETaaS is a layered platform deployed on top of a distributed architecture made of a local cloud of gateways. It enables full integration of different M2M systems (Adaptation Layer), exposes smart objects to services (Things-as-a-Service Layer) and services to applications (Service Layer) by means of a service oriented interface, and provides QoS support to allow M2M applications to negotiate the required SLA. To this end, it relies on WSAG4J [34], a Java open source implementation of the WS-Agreement and WS-Agreement Negotiation standards. QoS functionalities are implemented at different layers of the platform, and they cooperate in a cross-layer manner in order to provide a global QoS support. In fact, the Service Layer is in charge of negotiating the QoS parameters with the applications that want to exploit the platform functionalities. Before doing that, the Service Layer must negotiate the QoS parameters with the underlying Things-as-a-Service Layer, so that each service can ensure the fulfillment of the the service level requested by the applications. BETaaS also provides agreements templates and defines QoS classes in order to facilitate the negotiation process between the different layers.

Another negotiation architecture designed for the IoT scenario is presented in [35]. As the previous one it leverages the WS-Agreement specification, but it adopts a centralized approach. In fact it relies on a third party platform, the Cloud Sensing Brokering Platform (SBP), which aggregates the offer of different sensor network providers who are willing to make their sensing infrastructures available to external users with different requirements. Providers and users register to the platform by providing information about what they offer and require, and SBP

supports negotiation by acting as a broker between them and performs contract monitoring at runtime. Users who need to access a sensing service communicate to SBP the characteristics of the required services, then SBP returns to the users a list of candidate providers on the basis of the service information acquired at registration time. Users select a provider and then SBP carries out the negotiation phase on their behalf. The considered negotiation process is however quite basic in this context, since SBP simply prepares a contract taking into account provider's and user's constraints and asks both parties to sign it. If the contract is satisfactory it is signed and stored, and all the subsequent interactions between providers and users are channelled through the SBP, which can therefore monitor the delivered service.

Also the negotiation architecture proposed in [36], i.e. the IoT Mediator Platform, is based on a third-party platform. The main difference from the previous one is that here both smart sensors and users can decide which negotiation strategy the platform should follow in their behalf. In fact, for each smart sensor and user involved in a negotiation, the platform create a Java Agent Development Framework (JADE) container which acts as an autonomous agent. Each container is configured with the negotiation strategy that the entities selected when they registered to the platform, enabling a more flexible and powerful negotiation phase. In fact, users and sensor can define different strategies to be applied depending on certain conditions, and users do not need to select in advance a provider among the possible ones but they can make the platform perform negotiations with all of them and get the more convenient deal.

Another centralized negotiation platform is the Marketplace, presented in [37]. Even if not specifically designed for the IoT, the proposed approach can be beneficial also in this context since it abstracts from the heterogeneous offers of different service providers and allows for the automated consumption of services hiding the complexity of different contract terms and negotiation protocols. This is achieved by resorting to microflows, i.e. scripts allowing to correctly interact with the services registered to the Marketplace that are supplied by the providers during the registration phase. Microflows can also be leveraged to guide the negotiation phase, relieving users from the burden of knowing the specific characteristics of the counterpart's negotiation protocol.

With regards to negotiation objects, several QoS models have been developed to identify the service parameters which are of particular relevance in the reference scenario and which therefore are better suited to be included in the contract terms. In [38] the authors identify five classes of M2M applications defined by different levels in term of required real-time transmission, data accuracy, and transmission priority. In [39] the service characteristics taken into account are interactivity, delay, and criticality. A QoS management system for Wireless Sensor Networks (WSNs) is presented in [40], where also features related to energy and computational capabilities are considered. In [41] services are divided into inquiry tasks, control tasks and monitoring tasks depending on their delay and reliability requirements, and different QoS parameters are identified for the application layer (Service Time, Service Delay, Service Accuracy, Service Load,

Service Priority) and the network layer (bandwidth, delay, packet loss rate, jitter). In [33] a simple schema composed of three classes of services has been adopted (Real-Time, Assured, Best-effort) and the negotiation phase takes into account features such as response time and service period.

Finally, on the subject of agents decision-making models, [31] gives an exhaustive overview of the key techniques in the field of automatic negotiation. Here the authors describe negotiation as a distributed search in the n-dimensional agreement space, where n is the number of terms over which an agreement must be achieved. For each participant to the negotiation only a sub-region of this space corresponds to an advantageous agreement, and the process of negotiation precisely consists in finding a mutually acceptable point in the agreement space which lies in the overlapping area between all these sub-regions. To this end, different negotiation strategies have been developed, from the simple Dutch auction to more complex approaches based on the game theory or heuristics. The selection of the decision-making model in the considered scenario must take into account the limitations of the involved entities while allowing them to reach advantageous agreements. In [42] the authors present a comparison between the concession and trade-off models in the IoT context. The concession strategy requires entities to gradually relax some of their requirements to make their offers more attractive, while the trade-off strategy allows entities to increase their requests for the parameters they care the most while relaxing them for parameters which are more important to the counterpart. Trade-off strategies can guarantee higher levels of users' satisfaction about the reached agreements, i.e. higher utility, but they can also lead to a lower success rate, i.e. a lower chance of reaching an agreement eventually. This happens because users often have only incomplete information about the relevance of the various parameters for the counterpart, so in their counteroffers they could move away from each other's preferences and make the negotiation fail. To overcome this issue the authors propose an approach relying on the concept of mixed strategy Nash equilibrium of a negotiation game with two pure strategies, i.e. concession and trade-off. A pure strategy Nash equilibrium is a strategy profile where all the parties involved in the game are following the best possible strategy, as long as the other parties' strategies are known and remain unchanged. On the other hand, in a mixed strategy Nash equilibrium only probabilistic information about the other parties' strategies is known, i.e. probability distributions over a finite set of pure strategies. In their work the authors propose a mixed strategy where the involved parties can switch between a concession and a trade-off strategy at each negotiation round, according to the considered probability distributions. Such strategy has proven to outperform the concession model in terms of utility and the trade-off model in terms of success rate. Another negotiation approach targeted to the IoT is discussed in [36], where the authors propose a strategy to dynamically adjust the negotiation parameters by considering the current status of the device, i.e. its remaining battery charge. In more detail, the authors take into account smart sensors which wake up periodically to measure environmental parameters and transmit them to the applications they have an agreement with. In order to save energy and maximize profits, sensors wish to reduce the wake cycle and sell the same

measurement to the largest number of costumers. To this aim, they can dynamically adjust the price of the measurements they offer in order to enforce the desired customers' behaviour. For example, during the negotiation sensors can accept lower prices for already taken measurements or for measurements performed in the same wake cycle, while requesting higher prices otherwise. Also the remaining battery charge can be taken into account to further reduce or increase the prices.

2.3.3 Contract monitoring

Establishing contracts would be useless if no methods for verifying the compliance of the negotiated terms with the actual characteristics of the delivered service are provided. The definition of an effective monitoring infrastructure is then of paramount importance in order to allow the service provider to take the appropriate countermeasures when a degradation in the provided QoS is detected. In case this does not happen, it must allow the service consumer to prove that the SLA has been violated and to determine the related penalties.

The WS-Agreement specification provides interfaces through which the run-time state of the guarantee terms of the agreement can be monitored. However how this state is computed and what kind of infrastructure must be set up to perform this task is not specified.

In [43] three SLA monitoring architectures are described: monitoring based on a Trusted Third Party (TTP), monitoring based on a trusted module at the service provider, monitoring based on a model on the service consumer side. TTP-based monitoring is performed by an independent module hosted on an external machine which can control and record all the communications between the service consumer and the service provider but which is not able to inspect their internal state. In order for this architecture to be applicable both the service consumer and the service provider must trust the TTP, therefore it should be identified during the negotiation phase and included in the SLA. Monitoring performed through a trusted module integrated into the service provider allow to observe its internal state, but it provides weaker verifiability if compared to TTP monitoring. In fact the service provider could misbehave by reporting incorrect information to the monitor module in order to cheat on the SLA requirements. The third option requires the service consumer to determine if the observed service provider's behavior diverges from its expected behavior. Using this method, however, it is impossible to prove to third parties that the SLA guarantee terms have been violated. It can only be used by service consumers as a means for establishing the trust level they put in certain service providers. Scientific literature mostly take into account monitoring solutions relying on TTP, either in Cloud ([44], [45], [46]) and IoT ([35], [36]) environments.

Monitoring can be performed either offline or online. Offline monitoring requires that all the interactions are securely logged and stored, in order to make them available for examination when a party suspects that the SLA terms have been violated. On the other hand, online monitoring involves a periodic testing on the SLA terms to check whether they are met, allowing immediate

response to violations. However, online monitoring is not efficient because of the continuous tests that must be performed to assess the SLA compliance. Offline monitoring, on the other hand, is not capable to certainly prove that a violation occurred since it does not allow to verify whether a logged performance degradation is due to service provider's fault or to external causes. A hybrid approach that tries to overcome the limitations and overheads of offline and online monitoring is the reactive monitoring [47]. Reactive monitoring uses most of the time an offline monitoring scheme, the so-called passive monitoring. Passive monitoring relies on digitally signed receipts to provide proof that a certain stage of an interaction has been reached without any of the parties violating the SLA. The involved entities exchange the proofs without the help of a TTP. When a dispute arises, a previously agreed TTP is contacted to verify the receipts provided by the parties and take the appropriate decisions. This mechanism can be used to monitor discrete and state-based services. In case of suspects, the parties can also decide to ask the TTP to temporarily perform online monitoring to assess the compliance of continuous QoS parameters. In this way, all the violations are timely detected with less overheads than traditional online monitoring.

2.3.4 Payment mechanisms

The topic of payment mechanisms in decentralized and open contexts is of particular relevance, and it deserves special focus. The adoption of a widely accepted, secure and cost effective means through which the entities reaching an agreement can pay and being paid for the negotiated services is indeed of utmost importance to pave the way to a broad diffusion of contract technologies in the considered scenario.

In this direction solutions based on cryptocurrencies, such as Bitcoin, are gaining more and more popularity . Bitcoin [48] is an open source payment system which does not rely on a trusted central authority to perform money supply and transactions verification. Its core component is the Blockchain, a distributed, decentralized and immutable data structure replicated and shared among the members of the network. The Blockchain is made up of chained blocks of data which have been validated by the network itself according to predefined mechanisms which prevent the misbehavior of its nodes.

So far the most successful application building on top of the Blockchain technology pertains to the transfer of digital tokenized assets, such as bitcoins or other flavours of cryptocurrencies. Briefly speaking, users who want to perform transactions using the Blockchain are provided with a pair of private/public cryptographic keys to be used in a digital signature scheme. Through digital signatures users can prove the ownership of units of bitcoins and authorize their transfer. Once the network has verified the validity of a transaction, this is stored in the Blockchain permanently.

This framework presents appealing features, such as openness, transparency, privacy preserving properties and low fees, that make it attractive to several contexts. The scenario addressed in this chapter undoubtedly stands out as one of those that could make the most out from the

Blockchain technology. In fact, the greater scalability and reliability of the Blockchain due to its distributed nature matches well with the characteristics of an ever changing and expanding autonomous machines ecosystem heading more and more towards a decentralized management model.

By leveraging on cryptocurrencies also devices can become bank account holders, since the only thing required for this purpose is the possession of a valid private/public key pair. This would allow intelligent machines to buy and sell services and autonomously pay for them. Therefore Blockchain can be fruitfully used to provide a billing layer enabling a marketplace of services between devices [49]. Interesting examples in this direction are Filecoin [50], which allows machines to autonomously trade their storage space, and EtherAPIs [51], which enables a secure marketplace for APIs based on distributed trust. In [52] the authors envision the use of Bitcoin to support the Sensing-as-a-Service paradigm, according to which also IoT devices can autonomously buy and sell their atomic services.

Bitcoin is a platform independent, open source technology also attractive for its stability, versatility and extendibility. One of the most famous sons of Bitcoin is Ethereum [53], a Blockchain-based framework designed to run smart contracts. A smart contract is a computerized transaction protocol that executes terms of the contract itself [54]. In the Ethereum context it can be conveniently described as a program stored in the Blockchain, which written in a specifically designed scripting language, Solidity, and executed in a tamper-proof fashion by the Ethereum Virtual Machine (EVM) running on the nodes of the network. Smart contracts are inspectable by any network participant and its output is deterministic and cryptographically verifiable, so that the parties who decide to accept the contract terms are automatically sure to get the corresponding reward if they respect them and the corresponding penalties if they do not. But smart contracts allow to run a wide range of general purpose computations on the Blockchain. For example Ethereum has been proposed as platform for managing IoT devices [55] and as a trusted logging infrastructure for APIs usage tracking [56] that could be leveraged in the context of contract monitoring.

The Blockchain technology can therefore provide effective solutions for a wide range of issues related to different aspects of contract management in the considered scenario. However it also presents a few limitations.

Blockchain operations are computationally expensive, and involve high bandwidth and storage overheads which can be unaffordable for the most constrained nodes of the network. In order to tackle these issues in the IoT context, for example, the authors of [57] propose the adoption of private and centrally managed immutable Blockchains for IoT nodes which are merged in a publicly accessible distributed Blockchain hosted on an overlay network made of powerful nodes. The approach presented in [58] leverages on the cloud and fog computing paradigms to relieve IoT nodes from the most demanding Blockchain tasks.

Another issue is related to the limited scalability of the distributed consensus mechanism

that the Blockchain adopts to validate transactions, since it involves only a small subset of the whole network. Nowadays a Bitcoin transaction is validated on average after 17 minutes, a figure that is going to grow with the expected dramatic increase in the number of transactions. These timings are incompatible with a wide range of applications in the considered scenario.

Strictly related to this issue is the problem related to transaction fees, whose average value for Bitcoin has already risen above 1$, and which are going to increase more and more with the increasing of the average transaction validation time since higher fees guarantee shorter delays. Transaction fees of this magnitude clearly do not allow microtransactions, which represent a big part of the transactions that are expected to be carried out in the machine economy.

In order to overcome such issues a new distributed data structure, the Tangle [59], has been recently proposed. The Tangle has the same underlying principles as a Blockchain, but it is organized as a Directed Acyclic Graph and it supports its own digital currency called IOTA. Moreover it relies on a distributed consensus mechanism involving all the nodes of the network, which therefore scales well, exhibits confirmation times of around 10 seconds and does not entail any transaction fee. For these reasons the Tangle is one of the most promising technologies to implement payment mechanisms in forthcoming decentralized and heterogeneous distributed systems.

2.4 Service discovery solutions

In the previous section the relevance that automatic negotiation will take in future IoT scenarios has been extensively highlighted. However, before two autonomous agents decide to undertake a negotiation concerning the QoS of a certain service, the service requester needs to know that the other party can indeed provide a service which falls in the typology he is interested in. Several service discovery architectures have been proposed to meet this need. The main technologies related to this subject are detailed in the following subsections, and a synthetic description is depicted in Fig. 2.3.

2.4.1 Service discovery architectures

In [60] two basic architectures for service discovery are reported: directory-based and directory-less. In the directory-based architecture a node can act as a service provider, as a service consumer or as a service directory (SD). Service providers register their services to the service directory, and service consumers become aware of the available services by querying the service directory. On the other hand, directory-less architectures do not rely on directories. Service providers and service consumers interact directly, according to predefined patterns, to advertise and retrieve services. In [60] the authors mention also hybrid architectures, which consider the coexistence of the already mentioned ones. In this case service providers can register their service to an SD, if they locate any in their vicinity, or broadcast service advertisements. In the same way, service

consumers can query the SD or broadcast their requests and wait for responses from SDs or other nodes.

In [61] the authors identify three main categories into which service discovery architectures can be classified: centralized, distributed and hierarchical architectures. Centralized and distributed architectures can be respectively mapped into the directory-based and directory-less architectures described in [60]. On the other hand, hierarchical architectures partition the network in clusters, and employ nodes with high capabilities, one for each cluster, that handle service registrations and service requests on behalf of the root SD. Then, due to their characteristics, hierarchical architectures can be considered as a subclass of the directory-based architectures. In [62] the authors categorize various proposed solutions according to the technologies they rely on, namely distributed and peer-to-peer discovery services, centralized architectures, semantic based solutions, search-engines for resource discovery and approaches relying on specific protocols (e.g. CoAP, DNS). In this chapter we follow the classification proposed in [60].

1) Directory-based architectures

The directory of all available services can be centralized, i.e., hosted by a single node, or distributed. i.e., maintained by several nodes that exchange information about known services.

Centralized architectures are mainly adopted in wired and wireless LANs where one or more fixed hosts take up the role of the directory. The Universal Description Discovery and Integration (UDDI) [63] registry is an example of centralized directory developed by the industry. Another related example is Jini [64], a centralized service discovery architecture addressing service discovery between Java-enabled devices. In Jini a Lookup server stores services published by service providers and replies to service consumers queries. There are also centralized discovery architectures that allow global service discovery. For example, Object Naming Service (ONS) [65] is a DNS-based networking service that makes possible to retrieve objects information through their Electronic Product Code (EPC). GS1 maintains a global ONS service directory that can be queried through DNS queries. In order to leverage the DNS infrastructure the EPC is encoded as a proper Fully Qualified Domain Name (FQDM), and then a standard DNS query is performed. In [66] the authors make use of ONS in conjunction with UPnP, a directory-less service discovery architecture detailed in the next section, to define a device and capability discovery protocol easing the composition of applications in a Smart Home scenario. UPnP requires devices to transmit XML descriptions of the services they host, an approach that can be too burdensome for constrained devices. To address this issue the proposed protocol makes devices transmit only their unique EPC, which is then used by a control point to retrieve and expose the list of the corresponding provided services by leveraging the ONS mechanism. In [67] Digcovery, a centralized global service discovery architecture targeted to the Smart City scenario, is presented. Digcovery can interact with devices belonging to heterogeneous technological domains such as NFC, ZigBee and 6LoWPAN, through specific APIs provided by dedicated Digrectories. Each Digrectory collects the services offered by the attached domain and manages new registered

Service Discovery						
	Centralized-directory based architecture	Distributed-directory based architecture	Directory-less architecture	String/attribute service description	Semantic service description	Security
[32]				x		
[63]	x			x		x
[64]	x			x		x
[65]	x			x		x
[66]	x			x		x
[67]	x			x		
[68]	x			x		
[70]	x			x		
[71]	x			x		
[72]		x		x		x
[73]		x	x	x		x
[74]		x		x		
[75]		x		x		
[76]		x		x		
[77]		x			x	
[78]		x		x		
[79]		x		x		
[80]		x		x		
[81]			x	x		x
[82]			x	x		x
[83]			x		x	
[84]			x	x		
[85]			x	x		
[86]			x	x		
[87]			x	x		
[88]			x	x		
[89]			x	x		
[90]	x			x		x
[91]			x	x		x
[92]		x		x		x
[93]				x		
[94]					x	
[95]					x	
[96]	x			x	x	
[97]	x				x	
[98]	x				x	x
[99]	x				x	
[102]	x				x	
[103]	x				x	

Figure 2.3: Service discovery solutions

resources. Moreover it relies on REST APIs, i.e. APIs that are implemented and invoked according to the RESTful paradigm, to enable look-up and filtering of resources based on context-awareness, resource types, and geo-location criteria. A search-engine-like resource discovery framework is presented in [68]. It relies on the resource discovery mechanisms supported by CoAP. In fact, CoAP servers must expose the well-known URI "/.well-known/core" as a default entry-point to allow third parties to retrieve the list of the hosted resources and their attributes, encoded according to the CoRE Link Format [69]. Through this entry-point the framework retrieves the properties of the objects, then it stores such properties in a central registry and exposes its search functionalities by means of REST APIs. The work proposed in [70] leverages on the aforementioned CoAP service discovery features at all levels in the network hierarchy to achieve global CoAP servers discovery. Here the authors introduced new well-known entry points, e.g. "/.well-known/servers", to allow the retrieval of information about all the reachable CoAP servers. By creating a hierarchy of linked CoAP servers, clients who want to discovery the services offered by the available sensor nodes need to interact only with the internet gateway representing their anchor point. Indeed internet gateways can interact with the newly introduced well-know resources in the lower hierarchy level to get information about the available sensor gateways, which in turn can get lists of the resources provided by the attached sensor nodes resorting to standard CoAP service discovery functionalities.

Although simple to manage and implement, centralized solutions are prone to scalability and reliability issues, exacerbated by the great number of nodes involved in the considered IoT scenario, and by their mobility. In [71] the authors try to overcome the aforementioned issues by adopting a grouping mechanism. In their architecture the directory agent (DA) partitions the service agents (SAs) in groups depending on their physical position. DA uses the capability information of the nodes to assign to the most powerful ones the role of group leaders, which will assist the DA itself in maintaining the service registry updated. Group members send the control messages to their group leader, which, in turn, aggregate the information received and communicate the result to the DA periodically. In doing so the status maintenance traffic remains localized in the network area from which it originates and the bottleneck associated with the centralized directory is mitigated.

Distributed directory architectures show desirable properties for the reference IoT scenario, such as scalability and robustness. By adopting this kind of architecture, however, providing global service discovery (i.e., make every node able to discover all the services the network provides) can be difficult. Several discovery approaches developed and widely adopted by the industry implements distributed directory architectures. In Salutation [72] each device is provided with a local Salutation Manager (SLM) holding the descriptions of the device's services expressed as attribute sets. SLMs of different devices communicate in order to discover services available in other devices and achieve global service discovery. The Service Location Protocol (SLP) [73] is an IETF standard addressing service discovery either in directory-based and directory-less

mode. In the directory-based operation Service Agents (SAs) register their services to Directory Agents (DAs) and User Agents (UAs) unicast their queries to the DAs. In SLP services are described by means of sets of attribute-value pairs, and services can be grouped into scopes. UAs can query for services using keyword based search, by leveraging service descriptions and SLP capabilities of substring matching. In literature works are reported proposing more sophisticated distributed directory approaches, where directory nodes continuously communicate with each other in order to disseminate service information. In [74] a backbone of directory-enabled nodes is formed using a Minimum Dominating Set algorithm. Service providers advertise their service to one or more members of the backbone, and the members of the backbone forward to each other service discovery requests that cannot be satisfied locally. In [75] backbone members proactively exchange service information ensuring a smart forwarding of service requests to nodes that are likely to cache the description of the requested service. In [76] the authors propose a distributed resource discovery (DRD) architecture relying on multiple resource peers organized in a P2P overlay. Such peers are in charge of handling resource registration and discovery of constrained devices. A resource registration component takes care of registering the descriptions of devices resources in terms of IP address, resource path, resource type, content type and endpoint name, while a resource discovery component invokes the look-up method provided by the P2P overlay to retrieve the proper resource value depending on the provided description. Moreover, a proxy layer allows handling CoAP and HTTP messaging. In order to uniquely identify resource descriptions the authors use CoAP based URIs with endpoint names generated by hashing the devices MAC address. A clustering approach is reported in [77], where clusters are formed on a physical and semantic proximity basis. Each cluster has its own Service Access Point (SAP), which acts as a cluster directory by handling service registrations and service requests. SAPs also exchange summaries about the services provided by their own clusters and form higher-level clusters iteratively in order to support global discovery. In [78] nodes are clustered depending on their mobility pattern. In each cluster only the clusterhead, re-elected periodically, stays awake permanently and answer discovery requests. A widely used approach for implementing distributed directory is based on Distributed Hash Tables (DHT) techniques, due to their scalability, efficiency, and robustness. DHT are distributed data structures used in P2P systems that spread the information objects across the peers of the network in a deterministic way. Such objects are stored as key/value pairs and can be retrieved using their key. A DHT approach specifically targeted to the IoT scenario is presented in [79]. Here the gateways act as directory nodes, keeping track of any device joining or leaving their own network. Gateways use CoAP built-in service discovery capabilities for service discovery, and form two P2P overlays, Distributed Local Service (DLS) and Distributed Geographic Table (DGT). DLS is a DHT based architecture which provides a name resolution service based on storage and retrieval of bindings between the URI identifying a resource and the related information. DGT is an overlay scheme built using the geographical location of the nodes that allow to easily retrieve information about

nodes and their services located in any chosen geographical position. The main drawback of DHT solutions is that they only support exact match queries on a single attribute, i.e., the object identifier. In [79], for example, DHT keys are the resources URIs. In [80] the authors propose a DHT scheme supporting more sophisticated queries. To this end they map the n-dimensional attribute domain onto a 1-dimensional key domain by using a linearization technique based on Space Filling Curves, and then they use a Prefix Hash Tree (PHT) to store the so obtained keys. A PHT is a distributed data structure based on a binary trie which can be built on top of a classic DHT. A trie is a tree whose nodes are labelled with the keys prefixes and which stores each key in the only leaf node labelled with the key prefix. The adoption of a binary search approach over such a data structure enable complex operations such as multiattribute and range queries.

2) Directory-less architectures

In directory-less architectures, the communication between service consumers and service providers is not mediated by any directory. They are in general simpler than directory-based architectures because they do not need directory selection and maintenance mechanisms. Service consumers simply broadcast their request and so do service providers with their advertisements. A well-known directory-less approach developed by the industry is Universal Plug and Play (UPnP) [81]. In UPnP devices advertise their services and, upon request, they also present the service descriptions using XML. In Bluetooth SDP [82] services are described through service records consisting of a set of attribute-value pairs, and service consumers can formulate queries based on such attributes. In the directory-less operational mode of the aforementioned SLP, UAs multicast service queries and all receiving SAs with matching descriptions respond using unicast messages.

A basic problem in directory-less architectures is the identification of the appropriate frequency of service advertisements in order to prevent excessive bandwidth and energy consumption and at the same time ensure adequate dissemination of information. To this end, several strategies have been adopted. In [83] and [84] the advertisement range is bounded to a certain number of hops, beyond which the advertisement message is dropped. In the probabilistic forwarding approach, instead, the probability that a service request is forwarded decreases with the number of hops that the request has already traveled [85]. In [83], [84] and [86] nodes cache service information acquired by neighbor service providers and advertise such services along with their own. Approaches based on information caching can also make use of selective forwarding approaches [83] [84], in which nodes forward service requests to their neighbors that are known to host the requested services, or can allow also intermediate nodes to respond to service requests [87]. A directory-less approach relying on DNS-based service discovery is described in [88]. Here the authors propose an enhanced multicast domain name service (mDNS) whose messages are extended with an additional information section which is meant to carry service features information. Such additional information allows to reduce the number of messages exchanged in the network: service consumers can in fact perform multicast queries by specifying the features

of the services they are interested in, narrowing down the responses only to the pertinent ones. Moreover service providers can send unsolicited multicast advertisement which can be efficiently ignored by disinterested IoT devices. In [89] the authors propose to include information about device availability among the advertised service information. This allows service providers to power off and save battery during the scheduled idle periods without the risk of missing service requests. The authors also propose that devices monitor the network conditions and accordingly adjust the service advertisement period. Devices may also coordinate with each other to elect group advertisers, which are nodes advertising services on behalf of other nodes, to further improve the network performances.

None of the described architectures encompasses negotiation aspects, meaning that they assume services must be accessed as they are, in a static fashion. Moreover, only a small subset of the such architectures explicitly address security issues. The proposed solutions rely on service proxies [64], usernames and passwords [72], and public key cryptography [63] [65] [73] [81] [82]. In [90] pre-shared keys and Elliptic Curve Cryptography are used for smart objects authentication in a service discovery framework. In [91], on the other hand, the authors ensure that services are discovered only by authorized clients by resorting to Identity-based encryption, an encryption scheme that allows messages to be encrypted according to an authorization policy in such a way that only the entities satisfying the policy can decrypt them. A similar goal is pursued in [92]. Here service directories broadcast to unauthenticated service consumers only partial information about the advertised services, depending on the policies set by the service providers in order to limit unnecessary information disclosure. On the basis of the acquired information service consumers can determine if they are interested in the advertised service and, only in this case, to perform all the required authentication procedures, in order to reduce as much as possible useless connection establishment overheads.

2.4.2 Service description

A basic feature enabling automatic service discovery in the reference scenario pertains to machine-readable service description formalisms. In fact, when billions of smart objects will be available, mechanisms will be needed to allow machines to autonomously discover them and understand their capabilities, to enable automatic usage by software applications. The definition of machine-readable service description formalisms is fundamental to allow full automatic negotiation of services since service description has to be included in contracts for machines to understand the characteristics of the service and the meaning of the contract terms they are negotiating.

The simplest method for describing services is through Unique Universal Identifiers (UUIDs), as adopted in [93]. However, UUIDs do not allow service consumers to make an informed decision regarding the selection among similar services because UUIDs do not convey information about service attributes. Moreover, UUIDs must be a-priori known to all the nodes of the network, and

this is hardly feasible in the foreseen dynamic scenario.

Service descriptions implemented as attribute lists is one of the most widely adopted approach, and on this techniques rely essentially all the discovery architectures described in the previous section. Such approaches are generally based on exact keyword matching, and then they need that service providers and service consumers have agreed on the keywords to work properly. Again, this is not feasible in the open context we have to deal with, not to mention the fact that through simple keyword matching techniques it is hard to get all and only the services of interest.

Another class of approaches addressing service description relies on ontologies and semantic matches. Semantic technologies enabling machines to understand data annotated with semantic metadata (e.g., Microformats [94]) are one promising solution to automate the discovery and composition of device functionalities to provide new services. In [95] the author propose a scalable service discovery algorithm for the IoT based on semantic similarity and semantic relativity. Semantic similarity expresses the possibility that two concepts in different contexts can be replaced with each other without changing the semantic of the text, while the semantic relativity can be described as the possibility for the two concepts to appear in the same corpus. By combining such metrics the algorithm can compute the similarity between service descriptions and service requests and select the most suitable services. A service discovery solution based on semantic similarity is presented in [96]. Here providers who want to make their services discoverable must register them to a Discovery Broker by providing, among the other things, a semantic description of the service. The Discovery Broker can in turn be queried by clients who must provide a semantic description of the services they are looking for. To answer the queries the Discovery Broker leverages a Semantic Matching Engine which receives two terms, i.e. two service descriptions, as input and returns their semantic similarity as output. The Semantic Matching Engine resorts to web search engines to compute the distributional profiles of words, and then use cosine similarity in combination with exact string matching and matching within a certain Levenshtein distance to compute the semantic similarity of the input terms. Semantic similarity information is then used by the Discovery Broker to return the most relevant services to the clients.

In [97] the authors present DiscoWoT, a REST stand-alone web service implementing an extensible semantic discovery service targeted the IoT scenario which allows to semantically identify resources whose network addresses are known. It relies on multiple and arbitrary discovery strategies that can be defined and injected by developers and users at runtime. A discovery strategy is basically a mapping from an arbitrary semantic description language to the DiscoWoT internal resource description format. When providers want to leverage DiscoWoT to make discoverable for all users a web-enabled device described through a custom semantic language, they must provide the corresponding discovery strategy. Then users can ask DiscoWoT to translate the semantic information provided by the devices they wishes to interact with into the common and understandable DiscoWoT description format.

The service proxy framework presented in [98] includes a semantic overlay layer to promote the integration of smart objects into the Internet by making available and exploitable the information attached to them. The semantic overlay provides the semantic model of the underlying IoT network by maintaining an IoT ontology capable of describing complex IoT systems. When registering their services to the framework smart objects transmit semantic metadata along with service advertisements. Then the framework maintains the semantic representation of the smart object services in order to allow semantic service discovery and the dynamic composition of the provided services.

The middleware proposed in [99] uses a semantic knowledge driven approach to facilitate the composition of IoT services and provide the desired workflows. It relies on the Semantic Sensor Network Ontology (SSN) [100] and Software Component Ontology (SCO) [101] to describe the available sensors and software components. The middleware builds a semantic description of the tasks users wish to perform through a question-answer approach, and then selects the sensors and software components which are able to accomplish the requested tasks. The middleware also use semantic information to propose alternative strategies when the resources required to carry out the desired tasks are not available, or to enhance the requested service with inferred additional context information.

In order to enable capability-based and context-aware service discovery and invocation the framework presented in [102] includes a Semantic Access Layer (SAL), a middleware provided with a knowledge-base of IoT services that can be queried through a dedicated REST interface. Services functionalities and attached real-world context information are described using domain-specific ontologies in order to overcome the heterogeneity of IoT services and devices. The SAL conveniently mediates between the underlying IoT services and the overlying process engine, which can therefore express high-level functional requirements and let the SAL discovery the most appropriate services to fulfill the ongoing tasks.

In [103], in the end, the authors integrate semantics functionalities into a IoT/M2M service delivery platform by implementing them either as a backbone service or as a Common Service Function compliant with the oneM2M specifications [104]. The provided semantic services include semantic annotation, semantic query, semantic enrichment, data conversion and ontology related functionalities. This approach allow the invocation of the powerful but complex semantics methods, including semantic service discovery, by means specific APIs, which can be therefore leveraged also by constrained devices in their operations.

2.5 Mutual authentication solutions

In the reference scenario any object with communication and computational capabilities will be able to act autonomously, interacting with other objects, providing and consuming services, thus making smart the environment in which it is deployed. In this context, security will play a

fundamental role. In fact, aspects like privacy and mutual authentication are going to be key factors to enable a wide range of applications and to make people accept the deployment of these technologies in the environments they live.

Security technologies in communication systems are designed to achieve goals such as confidentiality, integrity, data origin authentication, entity authentication, access control and availability. In addition to this general goals, security in the reference IoT scenario must also address issues that arise due to the peculiar characteristics and behavior of the involved nodes, which are expected to be constrained in terms of available computational and energetic resources, pervasively deployed and mobile. For these reasons security solutions targeted to the reference scenario must take into account efficiency and scalability aspects. Forward and backward secrecy must also be assured, in order to prevent a node from accessing messages that are sent after it leaves or before it joins the network [105].

It is worth noting that in the large set of technologies that need to leverage on a reliable security infrastructure to work properly also fall the ones related to automatic contract management and service discovery. With regard to contract management, confidentiality and authentication aspects are of paramount importance through all the negotiation phases and when signing a legal binding contract. The same is true when talking about service discovery: mutual authentication must be established when a service provider registers its services to a service directory and confidentiality must be ensured for service requests and responses not to disclose sensitive information.

Confidentiality over an insecure network is usually achieved by means of symmetric (or private key) cryptography and asymmetric (or public key) cryptography algorithms. Symmetric cryptography algorithms perform encryption and decryption operations by means of a single key that is shared by both the end-points of the communication. Asymmetric cryptography algorithms, instead, use two different keys, one for encryption and one for decryption. Confidentiality of messages exchanged over an insecure channel can be achieved by encrypting them using such algorithms. Symmetric key algorithms are in general considered faster and simpler to implement than asymmetric key algorithms, and many constrained devices are provided with hardware accelerators to support symmetric cryptography. On the other hand, asymmetric cryptography and Public Key Infrastructure (PKI) enable a more flexible and scalable key management. Public key cryptography can also be used to implement digital signature schemes, through which it is possible to authenticate the identity of a sender, protect data integrity and ensure non repudiation of origin. Symmetric key cryptography too can be used to guarantee message authentication and integrity by means of Message Authentication Code (MAC).

Since the most challenging prerequisite for cryptography-based mutual authentication is that the authenticating nodes securely share a cryptographic key, in the following solutions for key distribution and management targeted to constrained wireless systems and IoT contexts are presented. A synthetic description of the reported technologies can be found in Fig. 2.4.

Mutual Authentication			
Symmetric Cryptography	Key distribution techniques	Probabilistic predistribution	[106][108]
		Polynomial-based predistribution	[109][110][111][113]
		Unencrypted broadcasting	[115][116][117]
		Key distribution center	[119][120][122]
	Authentication protocols		[123][124][125][126]
Asymmetric Cryptography	Optimization techniques	Compressed security protocols	[131][135][136]
		Certificate whitelisting	[137]
		Dedicated hardware modules	[138]
		Delegation-based approaches	[140]
		Compact certificates and alternative public key schemas	[143][144][146][147][148][149]

Figure 2.4: Security and trust solutions

2.5.1 Key distribution and management for mutual authentication: symmetric cryptography

The simplest method of key distribution is to provide all nodes of the network with a single key before deployment. Although very efficient, this schema suffers from many drawbacks. First of all, if even a single node is compromised, then the disclosure of the unique key would compromise the whole network. Moreover, secure end-to-end communication cannot be achieved since all nodes share the same key. Providing each pair of nodes in the network with a distinct shared key overcome these issues, but it is not a scalable approach. In fact, each node must store a number of keys that is directly proportional to the number of nodes in the network and, when a new node joins it, all the previously deployed nodes must be provided with new keying material.

In order to limit the number of keys that a node must store, probabilistic key predistribution schemes have been proposed [106]. In such schemes nodes are provided at deployment time with subsets of keys randomly selected from a key pool. The subsets are overlapping so that each pair of nodes has a certain probability of sharing at least a key through which the nodes can secure their communication. If a source node wants to establish a secure connection with a target node which does not share any key with it, the source node can rely on the other nodes that have at least a key in common with it in order to construct a secure path to the target node. In [107] the authors proved that, for a path to be present among each pair of nodes in a network composed of n nodes, the probability that two nodes share a key must be at least $2 * ln(n)/n$. By using this approach, when a node is compromised a subset of the keypool is compromised as well, and this

can affect the communications between other nodes in the network. In order to increase the resilience against node capture [108] proposed a q-composite random key predistribution scheme, according which two nodes need to share q >1 keys to establish a secure connection. The rationale behind this technique is that by increasing the required key overlap it becomes exponentially harder for an attacker with a given key subset to break a link. However, according to [107], the size of the keypool must be reduced to maintain each node reachable from any other node, so an attacker needs to compromise fewer nodes to have access to a larger portion of the keypool.

Other techniques aiming at efficiently store keying information inside the nodes are the so called polynomial-based key predistribution schemes [109] [110] [111]. They rely on the mathematical framework proposed in [112]. In such schemes nodes are provided with a polynomial share derived from $f(x,y)$, a bivariate n-degree symmetric polynomial over a finite field Fq, where q is the minimal prime number greater than the length of the secret key. Given the node ID u, a polynomial share is defined as $f(u,y)$. Since for this class of polynomial holds the property that $f(x,y) = f(y,x)$, two nodes can select a common pairwise key by exchanging their IDs and computing the value of the polynomial for such IDs. The work reported in [113] presents a similar approach, differentiating from the previous one in the key space, which in this case is based on matrices of dimensions $(n+1)\cdot(n+1)$ [114]. These techniques have been proved to be n-collusion resistant, meaning that, as long as less than n nodes are compromised, the keyspace remains secure.

Solutions for pairwise key establishment based on the assumption that the amount of time an attacker needs to compromise a node is greater than the time all nodes require to complete key information exchange are presented in [115] and [116]. However, these solutions lack of flexibility since it is not possible for a new node to join the network after deployment or, when it is made possible, this raises other security issues [105].

A simple scheme based on the assumption that an attacker can monitor the communications only between a small fraction of the deployed nodes is presented in [117]. Here nodes simply broadcast their keys unencrypted and afterward, in order to reinforce the security of possibly compromised channels, exchange other keying information through multihop and multipath secure links. This kind of solution is effective only for scenarios with specific characteristics.

Authentication techniques that are an adaptation of the Kerberos protocol [118] have been proposed, having the advantage of being more flexible and scalable. Such schemes rely on a trusted third party with whom each node shares a secret master key. The trusted third party acts as key distribution center that securely provides nodes with ephemeral keys by encrypting them with the master key. Nodes can then secure their communications by means of the so obtained session keys. SNEP [119] is an example of this kind of protocols. A key management architecture based on a trusted third party is presented in [120]. Here nodes rely on trust anchors to securely establish DTLS [121] session keys with the interacting clients. The architecture supports the establishment of DTLS connections both in pre-shared key mode and in raw public

key mode, i.e. using out-of-band validated keys. Both cases imply that each node shares a unique symmetric master key with its trust anchor. Clients who want to interact with a certain node can ask for a session key the related trust anchor. The trust anchor, after having authenticated the clients, issues and securely ships the session key to them, along with additional cryptographic information that will be used during the DTLS handshake by the nodes to derive, using the master key, the keys for the ongoing sessions. Clients can also leverage the proposed architecture to get authorization certificates which can be used to authenticate with the nodes. Indeed, clients can request trust anchors to issue authorization certificates containing their own public keys, signed with the master keys of the nodes they wish to communicate. Such certificates, along with the public keys of the nodes, can be used by clients to establish DTLS connections in raw public key mode.

The approach reported in [122] proposes a low-power implementation of the Generic Bootstrapping Architecture (GBA) for the automatic provisioning of session keys between constrained devices and application servers. GBA is part of the security infrastructure of mobile network operators, so it uses the permanent secret stored into the Subscriber Identity Module (SIM) card to ensure mutual authentication of the communicating entities and derivation of a session key to protect data exchanges. Notably it does not require access to the mobile network but only IP connectivity to the infrastructure of the mobile network operator.

In [123] the authors present an authentication scheme for multi-gateway WSNs based on symmetric cryptography. In their framework users can register with a gateway in their surroundings and access geographically distributed sensors by leveraging on their respective gateways. Users and sensors are expected to share secret keys with their home gateways, and gateways must also share secret keys between each other. Authentication and session key agreement is achieved through a multi-step protocol providing protection to several kind of attacks. [124] proposes a symmetric-key-based authentication and session-key agreement scheme for constrained devices. By using pre-shared keys devices can authenticate each other and select different session keys for each authentication session without the need to exchange additional parameters. The paper also proposes an hash-based authentication and session-key agreement scheme targeted to very constrained devices. Both schemes are resistant to replay attacks, man-in-the-middle attacks and wiretapped secret key attacks. In [125] the authors design a lightweight secure authentication protocol specifically designed to be integrated in CoAP. The protocol is based on the 128-bit Advanced Encryption Standard (AES). Devices achieve mutual authentication by using pre-shared keys and challenging each other through a four-step handshake. In more detail, the authentication handshake starts with the client sending to the server the ID linked to the symmetric key they share. Then the server retrieve the corresponding key and use it to encrypt a randomly generated nonce and a possible session key, which are sent back to the client. The client authenticates by decrypting the received message, and then by sending a composition of the pre-shared key, the server nonce and a freshly generated client nonce encrypted with the

session key contained in the previous message. In the end the server sends back the session key and the client nonce encrypted with the pre-shared key to authenticate itself and to signal that, from that moment on, the selected session key can be used. In [126] the authors try to reduce the overheads introduced by the DTLS protocol by providing an authentication mechanism which delegate to powerful servers and to gateways, when present, the key generation and key distribution functions. Constrained devices and servers authenticate each other by exchanging and decrypting randomly generated nonces encrypted using pre-shared keys, and then servers generate and securely transmit fresh session keys to be used in the following transactions. The proposed scheme reduces the amount of exchanged messages during the DTLS handshake and enhances the security of IoT devices since it creates new session keys each time a new session is established.

2.5.2 Key distribution and management for mutual authentication: asymmetric cryptography

All the previously described schemes rely on symmetric cryptography, meaning that all the nodes in the network must be securely provided before deployment with the necessary keying information (e.g., a master key, a polynomial, a key share, etc.) through which they are able to get the necessary pairwise keys to secure their communications and authenticate each other. Although these techniques can be considered effective solutions when taking into account closed scenarios where all the parties belong to the same administrative domain, they fall short when moving to our reference scenario, whose distinctive features are openness and dynamicity.

As anticipated above, the shortcomings of private key cryptography are overcome by Public Key Cryptography (PKC). In the Internet, PKC is the dominant solution to solve key management and authentication issues. In fact, with PKC there is no need for the two end points of a communication to pre-share any keying information: the party who wants to initiate a secure connection can use the public key of the recipient to encrypt the information he wishes to send, and they are sure that only the owner of the corresponding private key will be able to decrypt it. In order to bind a public key to an identity, the nodes of the Internet rely on a Public Key Infrastructure (PKI). The main components of a PKI are the Certification Authorities (CAs), who are in charge of verifying that a given public key belongs to a certain entity and subsequently issuing a certificate attesting that. A certificate is an electronic document signed by a CA which can contain several elements, the most important being the entities' identifiers and their public keys. The most popular standard to encode certificates on the Internet is the X.509 [127]. IPsec [128] and TLS/HTTPs [129] [130], the main security protocol suites used in the Internet, heavily rely on PKI.

Although PKC and PKI are effective solutions for key management and authentication, they also present disadvantages. In fact, PKC operations are more complex and power hungry than private key cryptography operations, and this is why, even in the Internet, PKC is only used to

securely exchange keying information to be used within a symmetric key cryptography schema. In addition, the size of a certificate chain in the X.509 standard can easily reach the order of kilobytes. Such characteristics do not fit well in constrained networks since they could lead to unacceptable energy consumption and transmission overheads. However, it would be desirable for the IoT world to take advantage of these mature, well-established and widely adopted protocols and tools, also in order to achieve seamless interoperability with the classic Internet. For these reasons, many efforts are being made to adapt the security technologies used in this context to the reference scenario, taking into account its limitations in term of bandwidth, energy, and computational capabilities.

Some of the proposed solutions try to reduce the overheads related to the extension headers introduced by security protocols. In [131] a compressed version of IPsec is presented. Here the authors use context aware header compression mechanisms to safely discard header fields that are implicitly known by the nodes in the constrained network. They define a new Next Header Compression encoding (LOWPAN_NHC [132]) both for Authentication Header (AH) [133] and Encapsulating Security Payload (ESP) [134], the protocols IPsec uses to deliver its functions. In the same way, a compressed version of DTLS is presented in [135], where the authors propose an NHC encoding for some of the DTLS protocols headers, namely Handshake and Record protocols, and for ClientHello and ServerHello messages. Although these solutions are fundamental gears in the machinery that will allow achieving end-to-end security and full interoperability between the Internet and the IoT worlds, they do not address key management issues. The aforementioned adapted protocols, in fact, have been tested only on a pre-shared key basis, which, as already explained, is not enough flexible and scalable.

In order to provide a more powerful key management system even in constrained scenarios, a lightweight implementation of IKEv2 [136], the protocol used by IPsec for key exchange, has been implemented. Also in this case, the authors propose an NHC encodings for IKE header and plan to use Elliptic Curve Cryptography (ECC) as the asymmetric cryptographic system for automatic key exchange, on the grounds that ECC provides the same security level of the more commonly used RSA with considerably smaller key sizes.

Another line of research deals with finding solutions to make certificates related operations affordable for the IoT scenario. In fact, since certificates enable an efficient key management and no human intervention is required with mutual certificate-based authentication, they can be conveniently used to authenticate machine-to-machine communications. In order to achieve this goal different approaches have been undertaken. In [137] the authors provide a solution based on certificate whitelisting. In this case, all authorized certificates are explicitly enumerated in a certificate whitelist, a data structure containing references to a set of trusted digital certificates. A certificate is validated successfully only if it is contained in the whitelist, and it can be revoked by removing it. Different whitelists can be generated depending on policies or device classes, and distributed to the target nodes using remote configuration protocols such as SCP or

HTTPs. Although certificate whitelists are practical and efficient since no complex cryptographic operations are needed to perform certificate validation, they are mainly suited for closed scenarios, where the set of devices and communication relationships are well-known.

Another kind of approach relies on Trusted Platform Modules (TPMs), additional tamper-proof hardware modules that provide support for public key cryptographic computations in order to get higher performances. In [138] the authors present a TPM architecture supporting the RSA-based cipher-suites of DTLS. Although the results achieved in terms of execution time are interesting and the cost for the adoption of new hardware acceleration engines for ECC and RSA are decreasing [139], this class of solutions do not address the problem related to the considerable overheads for certificate transmission and validation, that is crucial in scenarios which have to deal with lossy links and energy limitations of the involved devices.

Different approaches making certificate-based DTLS authentication feasible for resource-constrained devices have been proposed in [140]. Here the authors propose certificate pre-validation at the gateway to reduce the communication overheads in the constrained domain. In this scenario the gateway inspects the certificate-based DTLS handshakes to check the certificate validity and cryptographically verify the certificate chain, and forwards the certificate to the smart objects only if the pre-validation was successful. As a result the communication overheads related to undesired handshakes are reduced. Smart objects can avoid to re-verify the signatures of the certificate chain if the gateway is completely trusted, or, otherwise, they have to perform these operations again, so limiting the achieved improvements to the reduction of the transmission overheads. However, this solution is applicable only when the gateway is completely trusted since a malicious gateway could easily forward invalid certificates and compromise the security of the communication. As an additional strategy, the authors promote the use of session resumption techniques to reduce as much as possible the handshake overheads. The key idea is to perform the most heavy computations only once, during the initial handshake, and then maintain a minimal session state that allows the party involved in the communication to easily re-establish the secure channel. In particular, the authors propose to offload the encrypted session state on the party which is equipped with more resources, and to this end, they introduce different offloading mechanisms. In the end, this work presents a handshake delegation scheme that is meant to be used in the presence of tightly constrained devices which are unable to perform certificate validation themselves. In this scheme the smart object and the gateway share a common key via a secure channel. When the smart object wants to communicate with a server, the gateway establishes a secure session with the desired server and then stores all the information needed to resume the session. Such information is then encrypted and handed out to the smart object, which is then able to establish a secure communication with the server by performing a session resumption handshake that does not involve any certificate-related nor public key operations.

Other proposed approaches try to reduce the overhead for certificate transmission using compact certificates. As already mentioned, ECC is increasingly used in constrained scenarios

due to its reduced certificates size and performance advantages in respect of RSA [141]. The size of a X.509 certificate for a 3072 bit RSA public key is about 768 bytes, while the size of a X.509 certificate for a 256 bit ECC public key, which provides the same level of security, is about 85 bytes, about 9 times smaller [142]. In order to further reduce certificates size, [139] also propose a list of recommendations such as avoiding unnecessary extensions and using certificates which are directly signed by a root CA to reduce the length of certificate chains. An alternative mutual authentication scheme targeted to the Smart City scenario outperforming both RSA and ECC in terms of required resources is presented in [143]. It relies on a new lightweight public key encryption scheme which is used by devices to transmit challenges and verify whether the recipient can respond correctly. The number of steps the protocol involves can vary depending on the required security level and on system parameters such as the size of the largest encryptable message. The protocol however does not address the binding between identities and public keys, which are assumed to be a priori known.

A different approach which makes use of implicit certificates is proposed in [144]. Conventional certificates include both a copy of the public key and the digital signature of the CA, and the public key is explicitly verified when the signature is verified. In implicit certificates, on the other hand, the public key and the digital signature are replaced by the public key reconstruction data. The public key can be reconstructed by using the reconstruction data and the CA public key. In this case, there is no explicit validation of the CA signature, instead the subject proves his identity by performing an action that requires his private key. The Elliptic Curve Qu-Vanstone scheme (ECQV) [145] is the foundation for the technique of implicit certificates, which due to their reduced size, up to 57 time smaller than explicit certificates, and the lightness of the operations involved in their validation seem to be very well suited for the reference scenario. In [146] implicit certificates are used to implement a pervasive lightweight authentication mechanism for distributed IoT applications. The proposed authentication scheme assumes a cluster tree topology of WSNs, where cluster-heads are resource rich entities managing resource constrained nodes in their surroundings. Cluster-heads also play the role of Certification Authorities for their managed nodes. Intra-cluster authentication is therefore straightforward, since it is obtained by means of implicit certificates issued by the respective cluster-heads. Cluster-heads also support inter-cluster authentication by issuing certificates on demand also to devices belonging to other clusters. Since devices do not have the security credentials to communicate with other cluster-heads, they rely on their own cluster-head to get the needed certificate. To this end cluster-heads are supposed to collaborate and authenticate each other by means of DTLS and RSA keys. [147] uses implicit certificates to implement an efficient authentication and key agreement protocol. reducing up to 86.7% the airtime consumption if compared to traditional approaches. In more detail the proposed approach reduces the amount of transmitted data by resorting to a modified version of the widely adopted ECDH key exchange protocol which sends the public ECDH coefficients in the implicit format, i.e. using implicit certificates. In order to get peer authentication

and fresh keys per each different session, two additional authentication messages follow the regular ECDH exchange to convey freshly generated nonces to be used for session key generation. The proposed protocol proved to reduce up to 86.7% the airtime consumption if compared to traditional approaches. Moving to a 5G-oriented perspective, [148] proposes an Identity-Based authentication scheme for the IoT relying on Software Defined Networks (SDN). Identity-Based cryptography is a public-key scheme where the binding between a public key and the identity of the public key owner is ensured by a trusted entity acting also as key generator. Here the central SDN controller, which is all-pervading in the network and whose public key is known to everyone, generates the ECC public keys and the corresponding identities for fog distributed nodes as a result of a registration procedure. Since the SDN is global, nodes can move to different gateway domains and be able to authenticate themselves with different gateways. In [149] the authors present a service-oriented authentication framework for 5G-enabled IoT services. They leverage the public-key signature scheme proposed by Pointcheval and Sanders, in conjunction with a Public Key Infrastructure, to implement service oriented anonymous authentication and key agreement. The proposed approach allows users to successfully authenticate with service providers and access the services they are entitled to access according to the provided credentials without revealing their real identities. The proposed framework also allows for privacy-preserving slice selection, i.e. the type of the accessed services is protected against external attackers, and for service oriented key agreement, meaning that user, service provider and network controller can negotiate a unique session key enhancing communication and service access security.

Neither service providers nor local fogs can learn any information about the subscribers of IoT services, but both are aware of whether users are legitimate to access IoT services.

2.6 Challenges and open research issues

In the following subsections we point out the challenges related to automatic contract management, service discovery and mutual authentication which arise in new dynamic and decentralized IoT contexts. In fact, such technologies will be the foundations of forthcoming IoT scenarios, but further investigation and research are needed to fully achieve this vision.

2.6.1 Contract management

Automatic negotiation will play a central role in the next generation of IoT compliant service oriented cyber-physical systems. In an open context where the behavior of the operating agents is partially or totally unknown, and the context of execution evolves very rapidly due to the mobility of the entities involved and the change of their business relationships and goals, negotiation is, in fact, the only technique that is capable of coping with the dynamics of these new ecosystems.

Section 2.3 reports an overview of the main technologies and protocols that have been developed in the context of SOC regarding contracts management including payment mechanisms,

and some approaches that make use of these technologies in already established IoT systems have been presented. Although these solutions are fundamental steps towards a full integration of negotiation technologies in the future scenario, many issues still remain unaddressed, concerning:

- *contract definition formalisms*: the heterogeneity of the involved entities in terms of computational and energetic capabilities requires a careful assessment of the employed formalisms to represent contracts, in order to reduce the transmission and storage overheads. As highlighted, the commonly used syntax for agreement representation is XML-based, that is too verbose and complex to be managed in constrained environments. For this reason, a more lightweight solution should be considered (e.g., JSON [150], EXI [151, 152]). Moreover, the ubiquity of the smart objects and their involvement in the delivering of an increasing number of services will require mechanisms to regulate the way these entities handle the huge amount of data they will have to work on every day. For this reason contract formalisms in the future should also allow the definition of obligations, both for service providers and consumers, about service usages, legal issues and business rules;

- *the negotiation architecture*: a flexible negotiation architecture should be defined which leverage on cloud- and fog-based solutions to efficiently explore the space of possible agreements, to effectively negotiate the agreement terms and to deal with the high speed at which the entities relationships and capabilities change. In this context, in fact, the energetic and computational resources of the nodes can vary very rapidly because of the change of the context of execution, the establishment of new agreements and the termination of old ones. For this reason, the negotiation process has to take into account the residual capabilities of the nodes and the speed at which they change;

- *the contract monitoring architecture*: in the traditional cloud context service providers, such as Amazon EC2, transfer to users the task of verifying if the SLA terms are met, and when users detect a violation they can inform the service provider who will carry out the necessary checks and eventually, if a fault is acknowledged, will pay the related penalty. If this architecture holds when there are many users and a few trusted service providers, it falls short when any smart object, sensor or appliance can become a provider and/or a consumer of digital services. It is therefore necessary to identify a monitoring architecture which takes into account the peculiarities of the new scenario under this point of view, specifically the heterogeneity of the involved actors in terms of computational capabilities and the homogeneity of the actors themselves in terms of reciprocal trust. In this context is foreseeable that it will gain more and more importance the role of third-party-auditor, a trusted party which will be in charge of verifying that the guarantees included in SLAs are met when disputes arise in such an environment teeming with a multitude of peer autonomous agents. The development of a monitoring infrastructure with particular focus

on reducing the generated overheads for more constrained devices is also an important goal to pursue;

- *payment mechanisms*: the Blockchain is an established technology which allow autonomous machines to monetize their contract-regulated interactions in a secure and transparent fashion, and it can also play a role in the enforcement and monitoring of the agreements through smart contracts. However its heavy computational, bandwidth and storage requirements do not fit well in an heterogeneous scenario involving also constrained devices, and its delays and transaction fees represent a problem for the realization of several interesting applications in this context. The Tangle, a recently proposed new type of distributed database seems to be one of the most promising solution to overcome the Blockchain shortcomings. However also the Tangle comes with its limitations. Storage requirements are still a main issue, since constrained devices are not able to store the whole Tangle data structure. To mitigate this problem [153] proposes to resort to pruning techniques and swarm clients to allow devices to shard and collectively store the database. Moreover the Tangle is less secure than the Blockchain in the sense that it becomes vulnerable if one party takes control of at least 34% of the network, against the 51% of the Blockchain. In the end, the Tangle does not currently support smart contracts. Further investigations and improvements in these directions are needed.

2.6.2 Service discovery

This chapter highlights the great number of industrial and research solutions proposed to deal with the discovery of services in a connected ecosystem. Such solutions have very different features depending on the characteristics of the scenario they have been designed for, whose participating entities can be more or less heterogeneous, mobile and trustworthy. Finding the proper architecture for a certain context is then just a matter of wisely selecting the one that best fits with its characteristics. However, a peculiar feature of our scenario is precisely its high dynamicity and the speed at which its characteristics change (e.g. the reachability of the nodes, their energetic resources, their business relationships, etc.), a fact that makes it difficult to find a one-size-fits-all service discovery architecture.

For this reason, research in this field should aim at providing a service discovery architecture that is:

- *self-adaptive*: it needs to be able to autonomously change its operational mode depending on the current context of execution. In fact, in the emerging scenario mobile service consumers can go through environments characterized by different levels of dynamicity. For example, a smartphone will be in an office when its owner is at work, and in a crowded downtown street when its owner is taking a walk. The first context is closed and static as much as the second one is open and dynamic, and the service discovery logic hosted on the smartphone

should be able to find the needed services in both situations, adapting its discovery strategy according to the characteristics of the surrounding scenario, to the features of the entities it get in touch with such as their expected availability and trustworthiness, and to its own status, i.e., the smartphone remaining battery charge;

- *lightweight*: it have to reduce the overheads connected with service discovery, mainly related to the network traffic generated when performing service discovery operations. Promising approaches to mitigate this problem involve cross-layer optimizations (e.g., [154], [155]) who use control messages exchanged by the lower network layers to disseminate service information, resulting in reduced transmission overheads;

- *semantic-aware*: it must be able to cope with the semantic misalignments which arise in an open context. For service discovery solutions to work properly, in fact, all the involved parties need to understand what actually are the offered services and the meaning of the required QoS parameters, and although standards exist for ontology representation (e.g., OWL [156]) a scalable solution which allows autonomous entities to negotiate agreement on the semantic of their interactions is still missing;

- *secure*: in an open scenario the operations of service registration, discovery and delivery must be performed in a safe manner, and service availability must assured. In order to meet such requirements it must be possible for service consumers and service providers to achieve mutual authentication. Moreover, identity management policies must be put in place to allow the delivery of services only to the entities which have the necessary access rights.

2.6.3 Mutual authentication

In order to fully achieve the vision of the foreseen scenario the definition of a sound and reliable security infrastructure is required. The openness and dynamicity of the context in which autonomous agents will have to establish business relationships for the provision of dynamically negotiated services give rise to the need of securely establishing the other party identity and possibly his relevant attributes. Moreover, the heterogeneity of the involved entities requires a careful evaluation of the proposed solutions in terms of the required computational and energetic capabilities. In order to achieve these goals will be paramount the design of proper:

- *public key infrastructure*: other solutions based on private key cryptography, in fact, appear not to be scalable and flexible enough to cope with the features of this scenario. Many efforts have been made to define a PKI suitable for the envisioned heterogeneous and dynamic ecosystem. The most promising are, in our opinion, the ones involving the adoption of new lightweight certificates, and certificate validation schemes leveraging on the cloud/fog paradigm. These solutions, in fact, allow to save bandwidth for certificate transmission,

make the certificate validation process more affordable for constrained devices and provide flexible mechanisms to dynamically delegate the heaviest operations to more powerful nodes. However, a widely supported standard for mutual authentication targeted to this scenario is still missing;

- *identity management system*: the ability of objects and appliances to autonomously decide to act on behalf of their owners, or on behalf of the people or machines they are offering a service to, requires a careful consideration on how to manage this exploding complexity. Each object can, in fact, have multiple identities, and different attributes may be attached to each identity. Knowing such attributes can guide the negotiation phase since they certify some features that the device/application should have in order for an agreement to be reached. Moreover, according to the data minimization principle, all and only the attributes relevant for the negotiation should be disclosed in order to preserve the privacy of the involved parties as much as possible. Identity providers usually achieve this by issuing tokens or attribute certificates, the validation of which in the reference scenario arise the same issues that arise for the validation of public key certificates. The definition of a comprehensive architecture able to cope with both these domains is then desirable.

2.7 Conclusions

The IoT revolution refers to a scenario where all the objects that surround people in their everyday lives have communication, computational, sensing and actuating capabilities in order to create a new generation of high-level applications. To pursue such a vision, and in order to enable a next-generation of context-aware and self-adapting applications, the limits of traditional IoT scenarios (i.e., static and closed) must be overcome. This chapter takes as its starting point the observation that classic IoT scenarios, which are typically organized in silos and designed for a specific application, are giving way to more open, dynamic and ultimately complex architectures. The technologies that will be critical to tame this increasing complexity are those related to the automatic negotiation, the discovery and the security of services. They indeed provide the technological framework enabling the openness, decentralization, horizontality and dynamicity in the forthcoming IoT scenarios. In the chapter, the most widely adopted solutions in these fields from industry and research are reported, and their features analyzed in relation to the characteristics of the envisioned new IoT scenario. In this respect, it emerges that the main limitations of the proposed state-of-the-art solutions reside in the fact that these are not designed to be used in ecosystems where the operating entities have heterogeneous capabilities and the context of execution changes rapidly. The definition of less verbose formalisms for contracts representation which take into account the peculiar characteristics of the new scenario, the design of proper payment mechanism, the adoption of compact public key certificates and the use of approaches which leverage on the cloud/fog computing paradigm to promptly respond to

the change of the context of execution and to enable the outsourcing of the most onerous tasks are, in our view, the most promising strategies to overcome the pinpointed challenges and fully realize the next evolutionary step of the Internet of Things. In the following the thesis mainly deals with security and privacy solutions fostering the scenario we depicted in this chapter. In more detail, part III presents solutions to enhance users' privacy in IoT/Smart-Cameras based video surveillance frameworks for Usage Control systems and in mobile location-aware applications. This part also introduces an architecture to enable standard machine-to-machine mutual authentication in constrained IoT environments.

CHAPTER 3

A 5G PERSPECTIVE FOR SOFTWARIZED IOT ECOSYSTEMS

3.1 Introduction

Today we are witnessing several techno-economic drivers which are paving the way to a profound digital transformation of the Digital Society and Economy. Among these drivers, we notice the diffusion of ultra-broadband, the increasing of IT performance vs its down-spiraling costs, Software Defined Networking (SDN) and Network Function Virtualization (NFV) paradigms, the growing availability of open source software and also Artificial Intelligence advances.

This transformation (often called "Softwarization") will bring 5G (the 5th generation of network infrastructures) to become an end-to-end software platform. In 5G, a profound integration of processing, memory/storage and networking resources and functions will be achieved through an hyper-connected ultra-broadband "fabric".

In this direction, a common reference model is emerging, based on two main pillars: 1) a physical layer which will include computing, memory/storage and network resources (up to the edge and the Users' premises); 2) a virtualization layer which will allow providing high-level abstractions of all the infrastructure resources, functions and services.

It is well known that SDN [157] and NFV [158] are two of the key enabling technologies. Virtualized Network Function (VNF) and services will be dynamically combined and orchestrated to create specific end-to-end "service chains" serving applications. The infrastructure will provide "slices", as "isolated" pool of resources, where to execute multiple chains to serve applications (specific QoS requirements).

This network transformation will dramatically increase the flexibility of network and service platforms while ensuring the levels of programmability, reliance and performance required by

future 5G scenarios and applications (e.g., IoT, Tactile Internet, Immersive Communications, Automotive, Indutry 4.0, Smart Agriculture, Omics and E-Health, etc). So 5G will be much more than one step beyond today's 4G-LTE networks.

In fact, multi-level APIs will allow Operators/Providers, Third Parties or Users to assemble "service chains" of elementary services capable of dynamically meeting applications' requirements, and on an "on-demand" basis. In particular, it will be feasible to reserve and flexibly re-allocate network resources, depending on the actual transfer or analysis activities ongoing in a particular data center, and, ultimately, to support data exchanges with high-capacity links and low-latency figures.

Clearly, there is the international awareness that in order to make this huge digital transformation really possible and sustainable it will be necessary to evolve the legacy networks and services management and operations processes; as a matter of fact, the complexity and dynamism of such future services scenarios will be too high for human operators. The operations of physical closed boxes will have to be replaced by automated processes acting over millions of virtual entities (e.g., Virtual Machines, Containers, etc). Network management, control and orchestration will need to exploit innovative autonomic methods and control techniques (e.g., self-*, adaptive control, machine learning, neural networks, etc).

In line with this vision, this chapter elaborates concepts and modelling of an Operating System (OS) seen as a the strategic platform for enabling management, control and orchestration of future 5G services infrastructures, which will span from powerful Cloud Data Centers to constrained IoT nodes. The OS will make the 5G a truly distributed artificially intelligent "nervous system" of the Digital society and Economy. Moreover the chapter addresses also the ways this transformation will bring to the unification of the service modeling: in fact "anything" (from resources, to functionalities, to capabilities, etc) will be provided/accessed/managed as a service/application through standard APIs, made available across the different levels of this future infrastructure (e.g., IaaS, PaaS and SaaS).

3.2 Cloud-Edge computing and the disappearing SDN

SDN and NFV are two enabling technologies which will allow virtualizing cross-layers network service and functions in order to execute/manage/provide them as sort of appliances onto a software platform, which will be fully decoupled from an underlying hosting physical infrastructure [159].

In this direction, also Multi-Access Edge Computing (MEC) has been recognized as another key technological paradigm for enabling future 5G infrastructures.

European Telecommunications Standards Institute (ETSI) has defined MEC as a network and service paradigm for offering application developers and content providers cloud-computing capabilities and an IT service environment at the edge of the network.

Eventually, 5G will integrate fixed-mobile networks with highly distributed Cloud-Edge Computing facilities composed by (a limited number of) big-medium Data Centers (literally replacing current Telecommunications Central Offices) and a large number of small-medium Data Centers at the edge of the current infrastructure (i.e., in the access/distribution segments).

Interestingly Telecom Infra Project[1] (sister project to the Open Compute Project) set-up an Edge Computing Working Group focusing on lab and field implementations for services/applications at the network edge, leveraging open architecture, libraries, software stacks, into a MEC platform. If the integration of Cloud and Edge Computing will enable to pursue the functional disaggregation of network and service functions, SDN (and NFV) will require to build an highly flexible and programmable networking "fabric".

Notable example of strategic moves in these directions are reported in the next sub-section.

3.2.1 Examples of Google and FB infrastructures

According to Google vision, networking should be aiming for Cloud 3.0, a "serverless" computing architecture deeply based on actionable intelligence and machine learning.

Google infrastructure (Fig. 3.1) is currently based on four pillars: the B4-B2 (the WAN interconnect-peering metro), Andromeda (NFV), Espresso (SDN) and Jupiter (DC networking).

Espresso, Google's peering edge architecture, represents an interesting example of SDN strategy deployment. In particular, Espresso transforms the routing functionality into a distributed system that extracts the aggregate performance information of end-to-end network connections: in other words it allows to dynamically choose from where to serve Users based on measurements

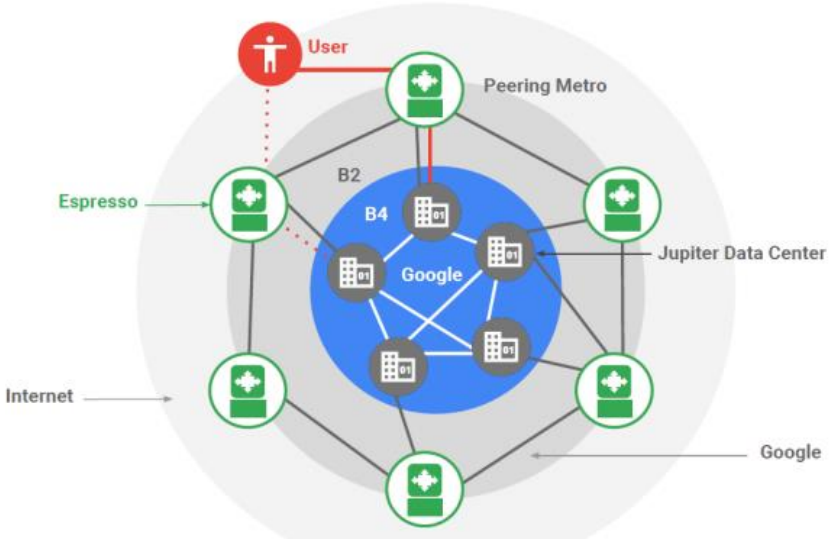


Figure 3.1: Google Infrastructure.

[1] http://telecominfraproject.com/

of how network connections are performing in real time. Specifically, Expresso runs Border Gateway Protocol (BGP) on servers co-located with label-switched fabric. Packet processors insert a label onto every packet, routers read the label. A local controller within every metro programs the label-switched fabric. The servers are sending summaries of how the flows are behaving in real time to a "global controller."

Andromeda is the NFV orchestration point (with API) for delivering virtual networks, virtual functions and services such as firewalls, routing, and forwarding rules while also providing Distributed Denial of Service (DDoS) protection, load balancing, and additional security features. In particular, Andromeda controls all the containers and VMs spread across the DC network fabric: some are also associated with soft switches, fabric switches, packet processors, cluster routers and disaggregated storage. It should be noted that Andromeda itself is the basis for delivering Cloud Platform networking services with high performance, availability, isolation, and security. All leverage the underlying Andromeda APIs.

Jupiter is the SDN DC interconnecting fabrics capable of supporting more than 100k servers (in 10 years Google DC fabrics increased more than 100x).

Jupiter fabrics can deliver more than 1 Pb/sec of total bisection bandwidth[2] (100k servers to exchange data at 10 Gb/s each). The three main design characteristics are: 1) Clos topologies; 2) Merchant silicon; 3) Centralized control protocols. A key point is controlling the growing complexity by collecting and distributing dynamically changing link state information from a central, dynamically elected, node in the fabric; then individual switches could calculate forwarding tables based on current link state relative to a statically configured topology.

This brief description of the Google architectures provides a picture about the Google's SDN migration path, which moved in stages from a fully distributed monolithic control and data plane hardware architecture to a physically decentralized (though logically centralized) control plane architecture. The long term paradigm is "Data Centers as a Computer" with centralized OS platforms (for network and service management and engineering).

Also for Facebook, SDN networking is an integral part of its infrastructure (Fig. 3.2), which includes:

- FBOSS, Wedge, and Backpack which are pieces of hardware and software architectures that power the open switches running in data centers, a key step toward a disaggregated network;

- Express Backbone and Open/R for making up SD-WAN, a dedicated private backbone network that leverages centralized traffic engineering in a global controller coupled with a new distributed routing system.

[2]Bisection bandwidth is the maximum amount of bandwidth in the data center measured by bisecting the graph of the data center at any given point.

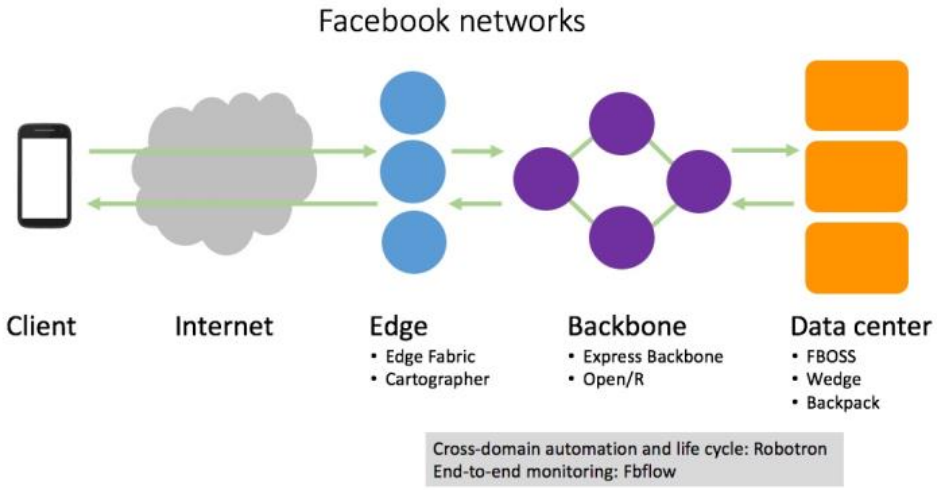


Figure 3.2: Facebook Infrastructure.

Concerning cross-domain automation/life cycle, Robotron is a system that handles every part of the network lifecycle across all network domains, from data center to backbone and edge. Host-based system Fbflow provides end-to-end flow monitoring.

In a nutshell, Facebook's global infrastructure shares information, ranging from photos and videos to messages through a global network of Points of Presence (PoPs), each connected to tens or hundreds of networks. Again, the long terms vision aims at disaggregated open hardware architectures (e.g., see Telecom Infra Project) overcoming costly and monolithic deployments. Amazon, Alibaba, AT&T [160] and other Tech.Co. (not addressed here for sake of brevity) are sharing the same basic principles for the long term evolution of their network and service infrastructures.

3.3 An Operating System for 5G

As previously mentioned, the long term evolution towards 5G will see the border between Cloud-Edge computing and the network systems blurring; these domains will "merge" together in a sort of "continuum" of resources and functions, offering flexibility and programmability through global operations.

Orchestration capabilities, which are already a key element for exploiting Cloud Computing capabilities, will eventually become an essential part of the Operations of future services infrastructures.

In Cloud Computing, orchestration is a mature concept and is generally referred to as the automation of tasks involved with arranging, managing and coordinating services deployed across different applications and enterprises, i.e., administrative domains, with the purpose of exposing them as single service instances. Orchestration of 5G services will have to span across

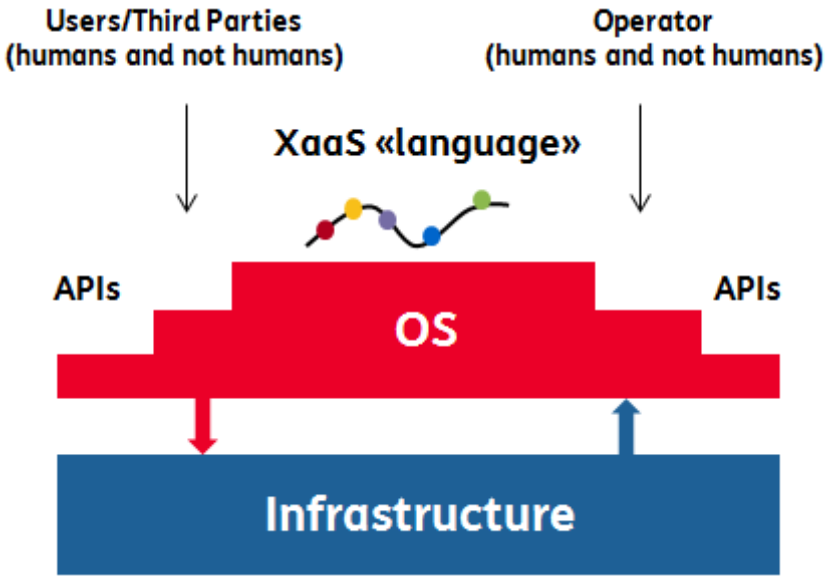


Figure 3.3: OS model for future 5G services infrastructure.

the different service levels: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

On the other hand, in 5G future service infrastructures, IaaS will assume a new meaning, as the CPU, storage, and network resources are not only hosted in a collection of Data Centers, but also deployed into the PoPs of the Telecommunications network (in its Core, Edge and Access segments). As a matter of fact, Telco's PoPs are already being transformed into medium-small Edge Data Centers.

The PaaS layer will feature a pool of software appliances that facilitate the end-to-end lifecycle of developing, testing, deploying, and hosting services and applications. Some examples of these appliances are databases, web servers, application servers, Apache Hadoop, Apache Storm, and load balancers, each of which will be integrated with other network appliances (e.g., telecom/internet middle-boxes) to design complex service chains, functions and applications.

Eventually SaaS will integrate multiple, interoperable PaaS and IaaS resources to deliver services and applications to the end users. In this broader perspective, orchestration should satisfy both horizontal and vertical interoperability requirements: horizontal, when considering the interoperability between the same tiers in different infrastructure stacks (such as cross-SaaS, cross-PaaS, or cross-IaaS); and vertical when addressing downstream-compatible infrastructure tiers in different stacks.

Along these lines, an open, vendor-agnostic, and interoperable North-Bound Interface (NBI) is expected to allow orchestrators to effectively control the underlying heterogeneous infrastructures and, ultimately, orchestrate multi-technology resources.

In the long term, the concept and model of Operating System for the 5G (5GOS) [161] can be

defined as an integrated software platform for enabling management, control and orchestration processes in future 5G infrastructures.

As well known, in computing systems, the adoption of such an operating system facilitates and encourages applications development by providing controlled access to a set of coherent and high-level abstractions of hardware resources (e.g., memory, storage, communication) and information (e.g., files, directories); in a similar way the 5GOS will provide controlled access to high-level abstractions of the 5G infrastructure resources (e.g. abstractions of computing, memory/storage and networking) thus enabling any vertical applications to be executed on the 5G infrastructure (Fig. 3.3).

The 5GOS, for example, could be designed with the same principles as a Linux kernel, with different levels of abstractions. It should run on every machine and provide applications (e.g., Hadoop, Spark, Kafka, etc.) with API's for resource management and scheduling across networks and services environments. Basically the 5GOS should provide automated resources management, scheduling processes placement, facilitating inter-processes communication, and simplifying installation and management of distributed functions and services.

In the past, the barrier for creating a Network Operating System has been rather high due to the quantity and complexity of the functional requirements for both software and hardware. Several previous attempts have been made with partial success depending on the targeted use case. However, Network Operators, Technology Providers and Innovation projects have made major progresses over the last few years.

Some of these concepts have been validated in proof-of-concept prototype (FROG) [160]. ONAP [162] initiative is developing some of the main building blocks applicable for the 5GOS.

It should be mentioned that 5G will have to face the security challenges typical of current Telecommunication infrastructures, but with a radically new and IT-oriented perspective (related to Softwarization).

In fact, SDN and NFV will bring the typical IT vulnerabilities. Furthermore, the increased capillarity of the network, reaching also the Fog and IoT nodes, may lead also to an increasing number of points of potential attacks, while at the same time a wider and wider diffusion of "cognitive systems" (e.g., Machine Learning, Deep Learning, Artificial Intelligence, Self-Organizing Networks, etc.) will create new vulnerabilities.

On the other hand, the same enabling technologies will also provide new instruments to mitigate risks. For example, the use of big data analysis and A.I. systems for inferring proactive actions (even based on early-warning signals of attacks), the adoption of SDN flexible features for fast traffic steering (e.g., quarantine, honey pots, slicing segregations), and the exploitation of NFV for virtualizing security appliances to be linked into the service chain.

3.4 Example of OS functional architecture

The 5GOS should include the basic functions of any operating system (e.g., bootstrapping, device drivers, process management, shell access) obviously extended for a 5G infrastructure.

Moreover the 5GOS should be the repository for basic network state information. This implies a shared data structure capable of supporting multi-vendors systems and applications for enabling sharing of common data amongst different protocols. Data structures include network state information, i.e., data about systems and interface state, information base (FIB) state, Neigh table and routing information base (RIB) and policies.

Standardized data models are required using, for example, a data-modeling language such as YANG. Northbound interfaces (NBI) to management&control systems may include NetConf/YANG and gRPC, and other mechanisms for telemetry.

As an example, [163] refers to the netlink protocol as a primary method for populating data structures and acting as bridge between the operating system and control and management plane applications. This will allow integrating existing Linux applications with limited modification.

In fact, as a general remark, it makes sense [163] that the 5GOS should be as closely aligned with one of the major Linux distributions to benefit of the existing ecosystems.

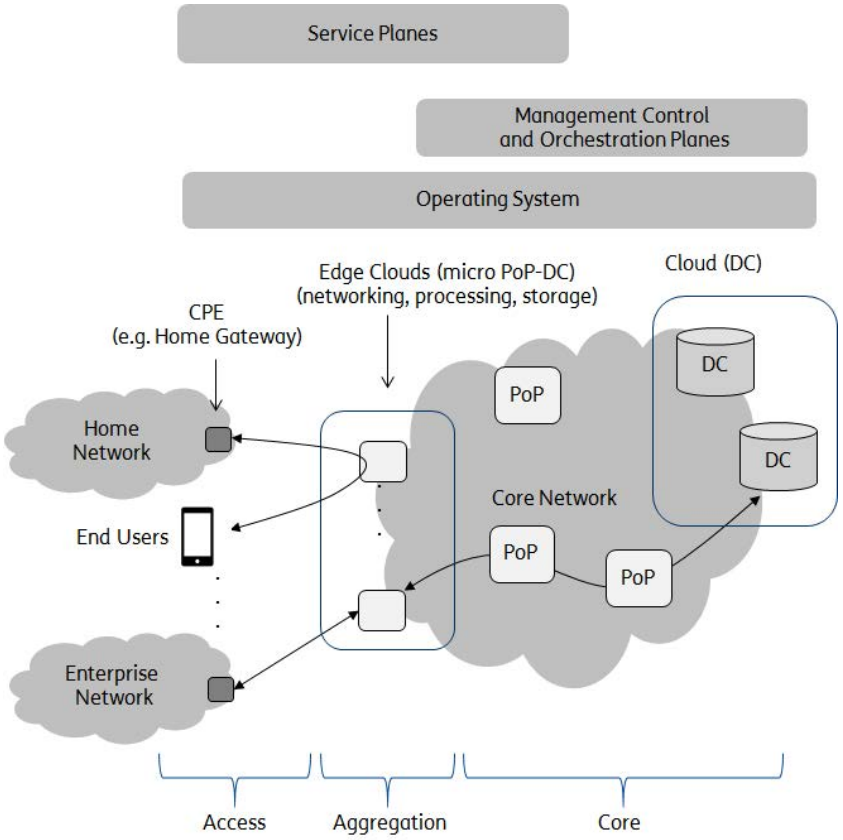


Figure 3.4: OS integrating Cloud and Edge Computing.

The infrastructure data plane is offering an abstraction layer of the physical resources functions and services (e.g., general packet forwarding, processing, storage/memory) of the 5G.

Another innovative aspect of the 5GOS is that it will reach, with some "nervous" terminations, even the Fog Computing and IoT resources, up to end User's terminals: thus 5GOS is really becoming a platform for executing Anything-as-a-Service crossing all layers of the infrastructure (IaaS, PaaS and SaaS).

One of the most important task which has to be supported is the orchestration of resources, functionalities and services across the Cloud and Edge Computing resources. Future scenarios will in fact look like the one depicted in Figure 3.4, where Edge Clouds, made of mini-data centres integrated with edge PoPs, and Fog computing resources, such as IoT nodes wearable devices, will need to be functionally integrated with the Cloud Computing resources.

A service chain which will have to be executed in an infrastructure slice (or even more across multiple slices) may require hooking orchestrated resources and functions which are both local (edge/fog) and centralized (cloud).

Typically this implies solving constrained optimization problems with various systems and methods (e.g., heuristics, machine learning, etc.) and require the access repositories of network state information (i.e., actionable big data).

3.5 Conclusions

The term 'ICT' has been often used to express the potential convergence between IT and Telecommunications. Today, it is more and more commonly accepted that Softwarization is going to impact the wider industry structure as a continuation, and substantial reinforcement, of this general convergence trend. In this directions, 5G will be much more than one step beyond today's 4G-LTE networks: 5G will be a truly distributed artificially intelligent "nervous system" of the Digital Society and Economy.

The programmability (e.g., with SDN), flexibility (e.g., with NFV) and high levels of automation (e.g., with AI) of 5G operations will optimize both investments (CAPEX) and operative costs (e.g., OPEX). Not only: 5G will support new service paradigms, maybe still be beyond our imagination: some examples would involve applications of the IoT, Tactile Internet, advanced Robotics, Immersive Communications and, in general, the Anything-as-a-Service paradigm. "Anything" means that any resources, functionalities, capabilities will be provided/accessed/managed as pieces of services/applications through standard APIs, which will be made available, in a secured way, across the different levels of this future infrastructure. Clearly, in order to make this huge digital transformation possible and sustainable it will be necessary to evolve legacy networks and services infrastructures and the management and operations processes.

To this extent, this chapter has elaborated concepts and modelling of an Operating System (OS) as a distributed software platform for enabling management, control and orchestration

of future 5G services infrastructures. Operators and Service Providers can adopt the 5GOS as a strategic instrument not only for operating efficiently their 5G infrastructures but also for exploiting diverse roles and business strategies (e.g., Bit-Pipe, Smart Bit-Pipe, Service Enabler, Service Provider).

A global effort is still required from hardware&software vendors to participate in the OS architectures standardization: moreover the design and development activities will have also include collaborations between academia, industry bodies and standard forums (e.g., ETSI, ITU-T, Linux Foundation, OCP-TIP, IETF).

Part II

Softwarized Internet of Things

CHAPTER 4

SOFTWARIZED IOT SOLUTIONS FOR SMART CITIES

4.1 Introduction

By 2050 over 70% of world population will live in cities, metropolitan areas and surrounding zones. There is thus a strong interest in making our cities smarter tackling the challenges connected to urbanization and high density population by leveraging modern ICT solutions. Indeed, progresses in communication, embedded systems and big data analysis make it possible to conceive frameworks for sustainable and efficient use of resources such as space, energy and mobility which are necessarily limited in crowded urban environments. In particular, Intelligent Transportation Systems (ITS) are envisaged to have a great role in the smart cities of tomorrow. ITS can help in making the most profitable use of existing road networks (which are not always further expandable to a large extent) as well as public and private transport by optimizing scheduling and fostering multi-modal travelling. One enlightening example of the role of ITS is represented by the seemingly trivial parking problem: it has been shown that a share of the total traffic of 10% (with peaks up to 73% [164]) is represented by cars cruising for parking spaces; such cars constrain urban mobility not only in the nearby vicinity of usual parking zones, but also in geographically distant areas, which are affected by congestion back-propagation. Indeed, the cruising-for-park issue is a complex problem which cannot be solely ascribed to parking shortage, but is also connected to fee policies for off-street and curb parking and to the mobility patterns through the city in different periods and days of the week. Indeed, cheap curb parking fees usually foster cruising, with deep impacts on global circulation, including longer time for parking, longer distances and a waste of fuel, resulting thus in incremented emissions of greenhouse gas. As such, the problem cannot be statically modeled since it has clearly a spatio-temporal component [165], whose description can be classically

obtained only through detailed data acquisition assisted by manual observers which – being expensive – cannot be routinely performed. Nowadays, however, ITS solutions in combination with the pervasive sensing capabilities provided by Wireless Sensor Networks (WSN) can help in tackling the cruising-for-parking problem: indeed by the use of WSN it is possible to build a neat spatio-temporal description of urban mobility that can be used for guiding drivers to free spaces and for proposing adaptive policies for parking access and pricing [166].

More generally, pervasive sensing and ubiquitous computing can be used to create a large-scale, platform-independent infrastructure providing real-time pertinent traffic information that can be transformed into usable knowledge for a more efficient city thanks to advanced data management and analytics [167]. A key aspect for the success of these modern platforms is the access to high quality, high informative and reliable sensing technologies that – at the same time – should be sustainable for massive adoption in smart cities. Among possible technologies, probably imaging and intelligent video analytics have a great potential which has not yet been fully unveiled and that is expected to grow [168]. Indeed, imaging sensors can capture detailed and disparate aspects of the city and, notably, traffic related information. Thanks to the adaptability of computer vision algorithms, the image data acquired by imaging sensors can be transformed into information-rich descriptions of objects and events taking place in the city. In the past years, the large scale use of imaging technologies was prevented by inherent scalability issues. Indeed, video streams had to be transmitted to servers and therein processed to extract relevant information in an automatic way. Nevertheless, nowadays, in an IoT perspective it is possible to conceive embedded vision nodes having on-board suitable logics for video processing and understanding. Such recent ideas have been exploited in cooperative visual sensor networks, an active research field that extends the well-known sensor network domain taking into account sensor nodes enabled with vision capabilities. However, cooperation can be meant at different levels. For instance in [169] road intersection monitoring is tackled using different sensors, positioned in such a way that any event of interest can always be observed by at least one sensor. Cooperation is then obtained by fusing the different interpretation of the sensors to build a sort of bird's eye view of the intersection. Instead in [170] cooperation among nodes is obtained by offloading computational tasks connected to image feature computation from one node to another. With respect to these previous works, one of the main contributions of this chapter is the definition and validation of a self-powered cooperative visual sensor network designed for acting as a pervasive roadside wireless monitoring network to be installed in the urban scenario to support the creation of effective Smart Cities. Such ad hoc sensor network was born in the framework of ICSI project [171], which aimed at providing a platform for the deployment of cooperative systems, based on vehicular network and cooperative visual sensor network communication technologies, with the goal of enabling a safer and more efficient mobility in both urban and highway scenarios, fully in line with ETSI Collaborative ITS (C-ITS). In this direction, and following the ICSI vision, the proposed cooperative visual sensor network is organized as an IoT-compliant wireless network

in which images can be captured by embedded cameras to extract high-level information from the scene. The cooperative visual sensor network is responsible for collecting and aggregating ITS-related events to be used to feed higher levels of the system in charge of providing advanced services to the users. More in detail, the network is composed of new custom low-cost visual sensors nodes collecting and extracting information on: (i) parking slots availability, and (ii) traffic flows. All such data can be used to provide real-time information and suggestions to drivers, optimizing their journeys through the city. Further, the first set of collected data regarding parking can be used in the Smart City domain to create advanced parking management systems, as well as to better tune the pricing policies of each parking space. The second set of data related to vehicular flows can be used for a per-hour basis analysis of the city congestion level, thus helping the design of innovative and adaptive traffic reduction strategies.

Extraction and collection of such ITS-related data is achieved thanks to the introduction of novel lightweight computer vision algorithms for flow monitoring and parking lot occupancy analysis, which represent another important contribution of this chapter; indeed, the proposed methods are compared to reference algorithms available in the state of the art and are shown to have comparable performance, yet they can be executed on autonomous embedded sensors. As a further point with respect to previous works, cooperation is here obtained by leveraging a Machine-2-Machine (M2M) middleware for resource constrained visual sensor nodes. The middleware is able to compute aggregated visual sensor node events and to publish them using M2M transactions. In this way, the belief of each single node is aggregated into a network belief which is less sensitive either to partial occlusion or to the failure of some nodes. Even more importantly, the visual sensor network (and the gain in accuracy that is possible to obtain thanks to the cooperative approach) is not only proved through simulation or limited experimentation in the lab, but is shown in a real, full-size scenario. Indeed, extensive field tests showed that the proposed solution can be actually deployed in practice, allowing for an effective, minimally invasive, fast and easy-to-configure installation, whose maintenance is sustainable, being the network nodes autonomous and self-powered thanks to integrated energy harvesting modules. In addition, during the tests, the visual sensor network was capable of gathering significant and precise data which can be exploited for supporting and implementing real-time adaptive policies as well as for reshaping city mobility plans.

The chapter is organized as follows. Related works are reviewed in Section 4.2, focusing both on architectural aspects and computer vision algorithms for the targeted ITS applications. In Section 4.3 the main components used for building the cooperative visual sensor network are introduced and detailed, while in Section 4.4, the findings of the experiments for the validation of the embedded computer vision logics (Section 4.4.1) and of the IoT-compliant middleware for event composition (Section 4.4.2) are reported together with the global results of field tests (Section 4.4.3). Section 4.5 ends the chapter with ideas for future research.

4.2 Related works

With the increasing number of IoT devices and technologies, monitoring architectures have moved during the years from *cloud* based approaches towards *edge* solutions, and more recently to *fog* approaches. The main drivers of this progressive architectural change have been the capabilities and complexities of IoT architectural elements. As the computational capacity of devices has increased, the intelligence of the system has been moved from its core (cloud-based data processing) to the border (edge-computing data analysis and aggregation), pushing further until reaching devices (fog-computing approach) to spread the whole system intelligence among all architectural elements [172]. One of the main features of fog computing is the location awareness. Data can be processed very close to their source, thus letting a better cooperation among nodes to enhance information reliability and understanding. Visual sensor networks in the ITS domain have been envisioned during the years as an interesting solution to extract high value data. The first presented solutions were based on an edge-computing approach in which whole images or high-level extracted features were processed by high computational nodes located at the edge of the monitoring system. Works such as [173] and [174] are just an example of systems following this approach. More recent approaches leverage powerful visual sensor nodes, thus proposing solutions in which images are fully processed on-board, thus exploiting fog-computing capabilities. Following this approach, several works have been proposed in the literature by pursuing a more implementation-oriented and experimental path, works such as [175] and [176] must be cited in this line of research, and, more recently a theoretical and modeling analysis [177]. By following a fog computing approach, the solution described in this chapter proposes (i) a visual sensor network in which the logic of the system is spread among the nodes (visual sensors with image processing capabilities), and where (ii) information reliability and understanding is enhanced by nodes cooperation (cooperative middleware) exploiting location awareness properties.

As discussed in the Introduction, the proposed visual sensor network is applied to two relevant ITS problems in urban mobility, namely smart parking and traffic flow monitoring. Nowadays, besides counter-based sensors used in off-street parking, most smart parking solutions leverage two sensor categories, i.e. *in situ sensors* and *camera-based sensors*. The first category uses either proximity sensors based on ultrasound or inductive loop to identify the presence of a vehicle in a bay [178, 179]. Although the performance and the reliability of the data provided by this kind of sensors are satisfactory, installation and maintenance costs of the infrastructure have prevented massive uptake of the technology, which has been mainly used for parking guidance systems in off-street scenarios. Camera-based sensors process videos streams captured by imaging sensors thanks to the use of computer vision methods. In [180] two possible strategies to tackle the problem are identified, namely the *car-driven* and the *space-driven* approaches. In car-driven approaches, object detection methods, such as [181], are employed to detect cars in the observed images, while in space-driven approaches the aim is to asses the occupancy status of a set of predefined parking spaces imaged by the sensor. Change detection is often based on background

subtraction [182]. For outdoor applications, background cannot be static but it should be modeled dynamically, to cope with issues such as illumination changes, shadows and weather conditions. To this end, methods based on Gaussian Mixture Models (GMM) [183] or codebooks [184] have been reported. Other approaches are based on machine learning, in which feature extraction is followed by a classifier for assessing occupancy status. For instance, in [185], Gabor filters are used for extracting features; then, a training dataset containing images with different light conditions is used to achieve a more robust classification. More recently, approaches based on textural descriptors such as Local Binary Patterns (LPB) [186] and Local Phase Quantization (LPQ) [187] have appeared. In [188], which can be considered as the state of the art, Support Vector Machines (SVM) [189] are used to classify parking space status on the basis of LPB and LPQ features and an extensive performance analysis is reported. Deep learning methods have also been recently applied to the parking monitoring problem [190]. All the previously described methods are based on the installation of a fixed camera infrastructure. Following the trends in connected and intelligent vehicles, however, it is possible to envisage novel solutions to the parking lot monitoring problem. For instance, in [191], an approach based on edge computing is proposed in which each vehicle is endowed with a camera sensor capable of detecting cars in its field of view by using a cascade of classifiers. Detections are then corrected for perspective skew and, finally, parked cars are identified locally by each vehicle. Single perceptions are then shared through the network in order to build a precise global map of free and busy parking spaces. A current drawback is that the method can provide significant results and a satisfactory coverage of the city only if it is adopted by a sufficient number of vehicles. Similarly, several methods based on crowd-sourcing [192] have been reported in the literature, most of which rely on location services provided by smart-phones for detecting arrivals and departures of drivers by leveraging activity recognition algorithms [193, 194].

For what regards traffic flow monitoring, the problem has received great attention from the computer vision community [195], even for the specific case of urban scenario [196]. Nevertheless most of the existing methods use classical computer vision pipelines that are based on background subtraction followed either by tracking of the identified blobs or of the detected vehicles (see, e.g., [197]). Such approaches are too demanding in terms of computational resources for deployment on embedded sensors. Among the various approaches and algorithms used, only few of them are for a real-time on site processing: among them, Messelodi et al. in [198] perform a robust background updating for detecting and tracking moving objects on a road plane, identifying different classes of vehicles. Their hardware is not reported, but in any case the heavy tracking algorithm let understand that it cannot be an embedded and autonomous platform. A similar approach is reported in [199] where the tracking is based on Optical-Flow-Field estimation following an automatic initialization (i.e. localization), but the final goal is more related to a 3-D tracking and tests were performed on a laboratory PC. Another feature-based algorithm for detection of vehicles at intersection is presented in [200], but again the used hardware is not reported, and

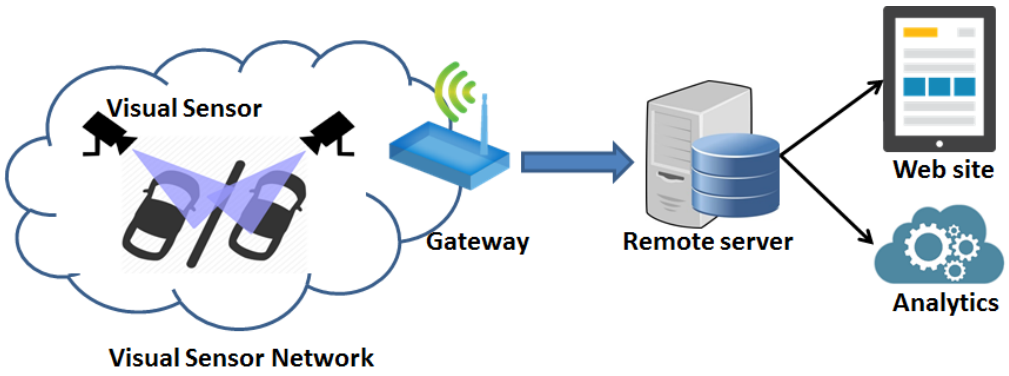


Figure 4.1: System architecture.

the tests seem to be performed only on lab machines. An interesting embedded real-time system is shown in [201], for detecting and tracking moving vehicles in nighttime traffic scenes. In this case, they use a DSP-based embedded platform operating at 600MHz with 32MB of DRAM, their results on nighttime detection are very good, but yet their approach did not have to cope with low-energy constraints. In [202], the real-time classification of multiple vehicles, also performing a tracking based on Kalman filtering, is performed using commercially available PCs, yet results are around 95%. Lai et al. in [203] propose a robust background subtraction model facing lighting changes problems with slowly moving objects in an *acceptable* processing time. But the used hardware is not described and the final acceptable processing frame rate is around 2.5fps, which is not acceptable for a normal traffic condition. The same problem arises in [204], where they perform real-time vision with so-called *autonomous tracking units*, which are defined as powerful processing units. Their test-bed are parking (in particular airport parkings) with slow moving vehicles and their final processing rate is again very low, i.e. below 5fps. Finally, in [205] the algorithm based on Support Vector is not used in real-time condition and, even if the processing times and the hardware are not reported, it must be a powerful unit, yet obtaining around 95% accuracy.

4.3 System Architecture and Components

In this section, the system architecture is reported before presenting the prototyped visual sensor node and the two key components that enable the creation of a cooperative visual sensor network: namely, computer vision logics, especially designed for deployment on embedded sensors, and the IoT middleware. The vision logics are meant to be deployed on board each single sensor in the visual sensor network, which is then able to provide autonomously its interpretation of the scene at *sensor level*. These local processing results are then integrated and converted into a more advanced, robust and fault tolerant understanding of the scene at *network level* [206], leveraging a middleware layer capable of event composition over resource constrained sensor networks. Both

the computer vision logics and the middleware solution have been used in the design of a visual sensor network, that has been physically deployed and validated on the smart camera prototype described in Section 4.3.1.

4.3.1 System Architecture and Visual Sensor Prototype

The high level system architecture of the deployed monitoring sensor network for urban mobility is reported in Fig. 4.1. It is mainly composed of three components: (i) the visual sensor devices able to exploit on board processing capabilities of the scene while providing M2M communication, (ii) the system gateway, acting as connection point between devices belonging to the visual sensor network and the Internet world, and (iii) the remote server in which detected events and data are stored for both analytic purposes and visualization on web portals. The following of the section focuses on the monitoring part of the system by reporting motivation and design choices behind the realization of the visual sensor node.

Although many nodes that might support the creation of a visual sensor network are currently available (see, e.g., [207, 208] for a review of some of them), nevertheless none seems to be satisfying for targeting outdoor ITS applications and to have capabilities for M2M communication. Actually, M2M communication is seen to be a key aspect to drive the shift from classical cameras and centralized video processing architecture to smart cameras and distributed video analytics over heterogeneous sensor networks. For these reasons, the design of an integrated IoT node was addressed, taking into account several requirements both from the functional and non-functional perspective. Indeed, the node should have enough computational power to accomplish the computer vision task envisaged for urban scenarios as described in Section 4.3.2 but, at the same time, it should be based on low power and low cost technologies. In this way, the nodes might be used to setup an autonomous, self-powered network in the city, using whenever possible photo-voltaic panels or other energy harvesting opportunities. Low cost components and architecture, in addition, guarantee that –once engineered– the node can be manufactured at low cost in large quantities, which is a fundamental aspect for the sustainability and wide scale adoption of the proposed technology. As for network communication, the node should be ready to support the interfaces needed in the IoT domain and, in particular, to support the middleware described in Section 4.3.3.

Inside the node, two main logical components have been identified, corresponding to networking and vision aspects. In particular the *networking component* takes care of communication both by managing M2M transactions and by interacting with the vision processes, e.g., by requesting their activation, transferring of their results, or setting parameters and behaviors of computer vision algorithms. Thus, networking component needs to be operating most of the time to guarantee responsiveness of the node to incoming requests and it must be low-consuming, but no high computational resources are needed. By converse, the vision component consumes many computational and energetic resources to run the algorithms reported in Section 4.3.2 when the

node is operational; however, if there is resource shortage, policies might be adopted to slow down computation, entering eventually a best-effort behavior without affecting the overall functioning of the sensors.

It is worthwhile to notice that the visual sensor network is intrinsically privacy-preserving. Indeed, images are processed locally at sensor level, without the need to transfer them to other locations, and then disregarded. Therefore, confidential information contained in the acquired images is not at risk, since they are neither stored nor accessible from a centralized location. Although it was not a primary scope of this chapter (and, thus, it has not been taken into account in the implementation), a further security level on the communications inside the visual sensor network can be added; for instance, security concerns might be addressed using the methods proposed in [209], where an elliptic curve cryptography approach for IoT devices is adopted and applied to the smart parking domain.

For the realization of the new node, a custom printed circuit board (PCB) has been designed to have the maximum flexibility of use while maximizing the performance/consumption ratio. A good compromise has been achieved by using a *Freescale CPU* based on the ARM architecture, with support for MMU-like operating systems GNU/Linux. This architecture has the advantage to integrate within it a PMU (Power Management Unit), in addition to numerous peripheral interfaces, thus minimizing the complexity of the board. Moreover, the CPU package of type TQFP128 allowed to minimize the layout complexity, since it was not necessary to use multilayer PCB technologies for routing. Thus, the board can be printed also in a small number of instances. The choice has contributed to the further benefit of reducing development costs, in fact, the CPU only needs an external SDRAM, a 24MHz quartz oscillator and an inductance for the PMU. Also considering the footprint of running programs, a 64MB SDRAM has been selected, which gives the possibility to keep in main memory a number of full resolution images more than sufficient for addressing the targeted applications. The chosen architecture has been proved to have an average consumption measured at the highest speed (454MHz) less than $500mW$, which makes it suitable for using energy harvesting strategies. A microSD slot is present, which is essential for booting the system, booting the kernel and file-system associated (EXT4); the board can be upgraded simply by changing the contents of the microSD. The PCB is connected to a SEED-EYE device [210] for managing networking aspects and has the capability to integrate camera sensors supporting USB Video Class device (UVC). The selection of a low-cost device brought to an easy-to-buy and cheap camera, the *HP HD 2300 Webcam*, that has been used during the experimentation.

4.3.2 Embedded vision logics for visual sensor networks

In the proposed cooperative visual sensor network, each node consists of an embedded sensor equipped with vision logics able to perform real-time scene understanding. The goal of this analysis is two-fold: (i) exact detection of parking slot availabilities in a parking lot, and (ii)

traffic flow analysis on relevant roads for parking. The major issue described here is the balance between the need to take into account low cost, scalability, and portability requirements, and the production of reliable and efficient computer vision technologies deployable on an IoT smart object.

Among the various scenarios, the differences in specific requirements are substantial: real-time constraints for traffic flow versus so-called near real-time (i.e., processing fulfilled in temporal terms of minutes) for parking slot monitoring; smaller area of interest to monitor a two-lane road, wider area to monitor a parking lot with several spaces; need for a fast frame acquisition rate for performing an efficient traffic flow monitoring. In the following, the two specific scenarios and the solutions implemented for solving them are analysed separately.

4.3.2.1 Parking lot availability scenario

As discussed in Section 4.2, the car-driven and the space-driven approaches are the two main strategies to deal with the problem of detecting parking lot vacancies. In car-driven approaches, features detection methods are employed to detect cars in the observed images, while in space-driven approaches the aim is to detect empty spaces rather than vehicles. Recent works proposed in the literature show very good performance (see Table 4.2 in the next section). Although this scenario, as mentioned above, allows for less restrictive processing constraints (e.g., not a strict real-time processing), state-of-the-art algorithms generally require performing hardware and it is not possible to deploy them on low memory/low computational power sensors. Various algorithms have been studied and designed to be deployed on the proposed visual sensor network for the analysis of parking lot occupancy status. The methodology chosen is based alternatively on the analysis of single frames, or on frame differencing, in order to highlight the changes in the Regions of Interest (RoI) with respect to an adaptive GMM background reference image [211]. For outdoor applications, background cannot be static but it should be modeled dynamically, to cope with issues such as illumination changes, shadows and weather conditions. In order to improve the robustness of the algorithm with respect to environmental light changes, normalized versions of the images are computed and used, with respect to global illumination parameters (average and variance of both the current and reference image). To improve computational efficiency, image analysis and frame differencing for detecting changes are performed only on predetermined RoI in the acquired frame. Each of the RoI corresponds to a specific parking slot, it is set up manually with the help of a graphic tool, and can be of any polygonal shape (this helps to avoid occlusion like trees, poles or other static artifacts). In figure 4.2-A) the green zones are the defined parking lots RoI. For each of the regions a confidence value of the occupancy is computed in real-time. The output of a single node is a vector of probability values of the occupancy for each parking slot.

In the first phase, the input image is evaluated with respect two lightweight algorithms (a car-driven and a space-driven ones). The car-driven method searches for car features in the predefined RoI. A fast edge-detector, the Canny operator [212] is used to obtain a very crisp image

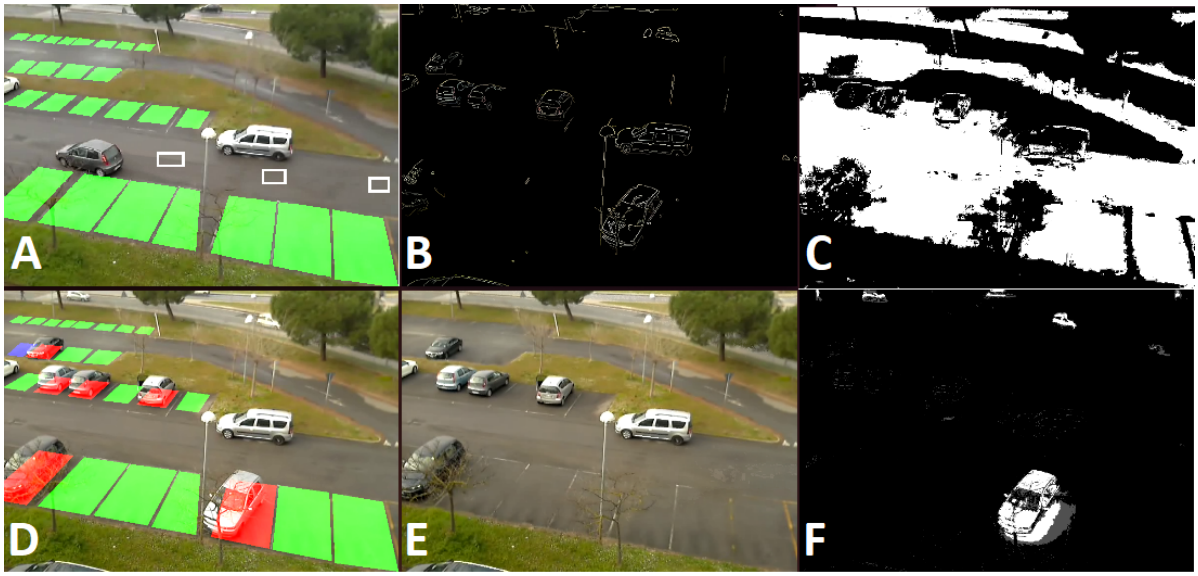


Figure 4.2: A) RoI for a set of parking lots are set up manually with the help of a graphic tool. Small rectangles on the driveway define the samples for asphalt detection B) The output of the Canny edge detector C) White regions represent areas where asphalt is detected D) Current input image with augmented reality displaying the status E) Background image F) Frame differencing to detect major status changes.

of the vehicles contours of the frame acquired at time t as it is shown in figure 4.2-B; we guess if a vehicle occupies the RoI R_k calculating the index $e_k(t)$ which is proportional to the ratio of edge pixels with respect to the square root of total number of pixels in R_k, i.e.:

$$e_k(t) = \#(\text{edge pixels in } R_k)/\#(\text{pixels in } R_k)^{(1/2)} \tag{4.1}$$

For an empty slot $e_k(t)$ is very close to zero, while slots with active edges (i.e., most probably not empty) have a higher value (e.g., experimentally in the range $5-12\%$). This index cannot be used if a pole or a tree partially create an occlusion of the lot (they can be misinterpreted as a presence in the lot).

The second index came out from a space-driven approach: considering that an empty slot should appear as plain asphalt concrete, we aim to detect asphalt areas based on the color characteristics of small samples on the driveway (see figure 4.2-A); we consider a subset of Hue and Saturation in the HSV color space (we discard the brightness to be more robust against different illumination); furthermore these values are periodically updated to include changes occurring due to time/light changes, rain, etc. For every input frame, the asphalt detection acts like a binary filter (see figure 4.2-C)

The index $a_k(t)$ is the ratio of asphalt pixels and total pixels of a RoI, i.e.:

$$a_k(t) = \#(\text{asphalt pixels in } R_k)/\#(\text{pixels in } R_k) \tag{4.2}$$

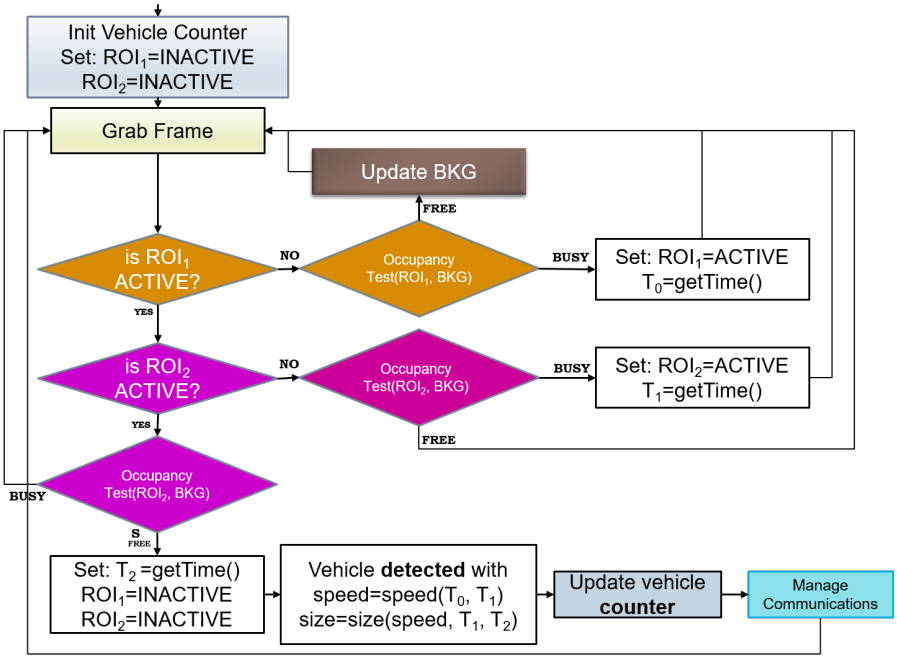


Figure 4.3: Flow chart of the traffic flow monitoring algorithm.

This index is really trustworthy if it is close to 1. As a rule of thumb for value greater than 90% the lot can be consider available. The index of a not-empty lot is generally below 20% but cars with similar color of the asphalt sometimes arrive at higher value. However the presence of different non gray area like windows, tires, plates and so on do never scores like plain asphalt.

Using and combining these two indexes, we compute a final evaluation of occupancy probability $P_k(t)$ of the slot k as seen from this sensor:

$$P_k(t) = e_k(t) * (1 - a_k(t)) \tag{4.3}$$

This joint index is valuable mainly to determine the initial status (analysis of the initial frames) or when there is a sudden light change that blinds the camera. After the first k frames (being k a number between 10 and 30) the background image is available (see figure 4.2-E) and from that moment all the major events are determined by frame differencing. This is a very reliable detector also with our 0.5 fps because during parking every vehicle move is slow (figure 4.2-D shows the very clear shape of the parking car – notice the absence of this car in the background image). If a major change happens in a RoI, it is very easy to be detected and the system keeps track of the history of these changes.

4.3.2.2 Traffic flow monitoring scenario

On the contrary to the previous scenario, in this one restrictive processing constraints exist, due to the need to detect not only all the passing vehicles but also in view of a deeper analysis of the traffic flow, i.e., sensing average speeds and categories of vehicles.

With respect o the classical computer visions techniques reviewed in Section 4.2, an ad hoc lightweight method was designed which is suitable for deployment on embedded devices. For the deployment on the visual sensor network, background detection is performed only on small quadrangular RoI which are sufficient for modelling physical rectangles under a perspective skew. In addition, lightweight methods are implemented for background modelling, that is determining if a pixel belongs to foreground (i.e., meaning that it is changed w.r.t. the previous scene), or to the background (i.e., pixel unchanged). Three main approaches have been considered, i.e., frame differencing, static background and adaptive background. The latter class of algorithms proved to be the most robust for use in uncontrolled outdoor scenes. The background is constantly updated using both the previous background model and the latest acquired actual image.

The data extraction procedure starts by taking as input one of several RoI for each lane suitably segmented in foreground/background. When processing the frame acquired at time t, the algorithm decides if the RoI R_k is occupied by a vehicle or not. Such an occupancy test is based on the ratio of pixels changed with respect the total number of pixels in R_k, i.e.:

$$a_k(t) = \frac{\#(\text{changed pixels in } R_k)}{\#(\text{pixels in } R_k)} \tag{4.4}$$

Then $a_k(t)$ is compared to a threshold τ in order to evaluate if a vehicle was effectively passing on R_k. If $a_k(t) > \tau$ and at time $t-1$ no vehicle was detected, then a new transit event is generated. If a vehicle was already detected instead at time $t-1$, no new event is generated but the time length of the last created event is incremented by one frame. Finally, when at a time $t+j$ no vehicle is detected (i.e., $a_k(t+j) < \tau$), the transit event is declared as accomplished and no further updated. Assuming that the vehicle speed is uniform during the detection time, the number of frames j in which the vehicle has been observed is proportional to the vehicle length and inversely proportional to its speed. In the same way, it is possible to use two RoI, RoI_1 and RoI_2, lying on the same lane but translated by a distance Δ, to estimate the vehicle speed. The algorithm uses two RoI for classifying vehicles with respect to their size and speed class (see Fig. 4.3). At the beginning both RoI are set as inactive. Then a frame is grabbed from the sensor and an occupancy test is run on both RoI sequentially. If a RoI becomes busy according to the occupancy test, it is marked as active. RoI_2 is only tested for occupancy when RoI_1 is active. When both RoI are active, a transit event has occurred. The algorithms continues to grab frames until RoI_2 becomes free. At this point, the transit event is concluded since the vehicle has left the detection area. It is then possible to classify the event. Using counters based on the elapsed time and knowing the distance Δ among the RoI, the vehicle speed and size are computed. Notice that the main loop described is a simplified version of the one actually implemented, where there are some further controls and heuristics to avoid false alarms. For instance, a timeout is set for the RoI_2 to become active after RoI_1 has. Indeed, since the RoI are within a few meters distance, no more than some seconds can elapse from the occupation of the first to the occupation of the second one.

4.3.3 IoT middleware for event composition

In an ITS system, in which the roadside network is composed of an IoT-compliant visual sensor network devices, the remote management of the nodes as well as their cooperation and data collection functionalities can be all managed by an IoT middleware. In this respect the ICSI Middleware proposed in [213] has been extended and adapted to forward both simple and aggregated visual sensor node events towards remote gateways by using Machine-2-Machine (M2M) transactions [214]. Further, the middleware enables a remote configuration of the IoT visual sensor nodes, which turns out to be useful in several scenarios, for example when event aggregation strategies are adopted. The possibility of aggregating events in the roadside segment enables in-network processing capabilities which can be used to increase the robustness of the information. In fact, by using visual sensor nodes a possible problem is the temporary occlusion of the field of view, which can be overtaken deploying different visual sensors monitoring the same scene, while requiring an event processing heuristic inside the network. When an event, e.g., a parking slot becoming free or busy, is detected by more than a single sensor it is necessary to process and aggregate all the events to provide a final decision on the status of the slot. In this depicted scenario the node responsible for the in-network event aggregation could be a single point of failure in case it fails, e.g., due to battery depletion. To tackle this problem the middleware implements a dynamically reconfigurable system, based on standard interfaces, capable of moving event aggregation tasks from node to node at runtime. Such a functionality, better described in the following, is implemented by enabling a virtual machine based approach in the part of the visual sensor node responsible for the IoT communications.

Considering a pervasive roadside segment based on visual sensor nodes, the ICSI Middleware is a software component instantiated on each node. The high level middleware architecture is shown in Fig. 4.4, while the ICSI WSN high level architecture is depicted in Fig. 4.5. The middleware has been designed following the Component-based Software Engineering approach, which promotes the Separation of Concerns and the Single Responsibility principles. According to these principles, the ICSI middleware has be partitioned into distinct modules which encapsulate very specific, not overlapping functionalities, and which communicate with each other via inter-

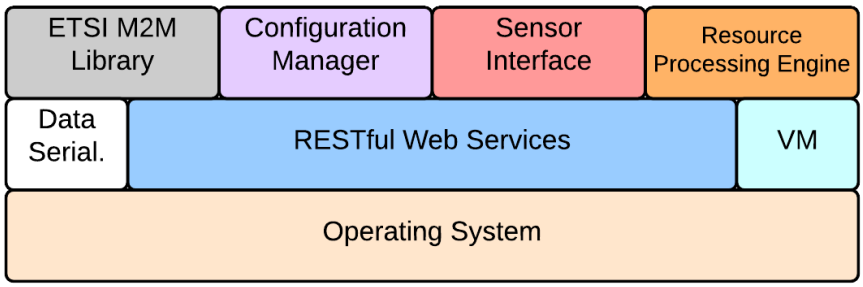


Figure 4.4: Middleware architecture

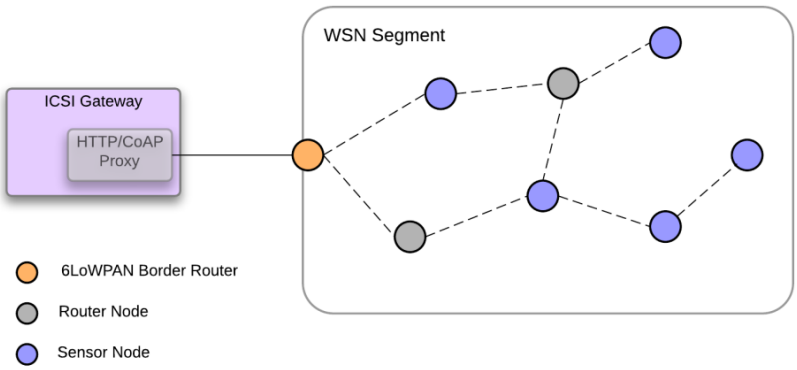


Figure 4.5: WSN architecture

faces. The adoption of this approach allowed to come up with a high cohesive and low coupled system, with all the implied benefits in terms of maintainability and flexibility. The core modules are placed on-top of the operating system and make use of networking, virtualization and data serialization services provided by embedded software libraries. The main middleware components are: i) the Resource Processing Engine (RPE), taking care of in-network event processing tasks; ii) the ETSI M2M library, supporting standard communication with system gateways; iii) the Sensor Interface, a software library abstracting the functionality of on-board sensors; iv) the Configuration Manager, enabling the remote configuration of the node.

The whole middleware has been developed on top of Contiki OS [215], an open-source operating system targeted to IoT compliant devices and fully supporting IoT protocols, i.e., 6LoWPAN [216, 217], RPL [218], and CoAP [219]. Contiki has been preferred to other embedded operating systems such as Riot and TinyOS because of its full and certified implementation of the IoT stack. In order to support some of the advanced features of the ICSI RPE, we added client-side support for CoAP Observe, which is needed by RPE to "observe" the input resources. Moreover we implemented IPv6 loopback communication (i.e., the possibility for a process to communicate with another process on the same node using sockets), needed by the RPE tasks to interact with resources belonging to the node on which they are deployed. The RESTful Web Service [220] component, based on the CoAP protocol, handles all network data inputs, outputs and implements a resource directory service. Such a component uses other important features implemented in the CoAP implementation provided by Contiki: i) block-wise transfers to manage large resource representations; ii) resource observation, to publish changed resource representation to all subscribers; iii) sending of the acknowledgment for a request and the relative application data in separate responses in order to avoid undesired request retransmissions and timeout when the request involves time consuming tasks; iv) resource discovery using the CoRE Link Format by automatically generating the /.well-known/core handler at which all resources are registered. The

```
/tasks                  # list currently installed tasks [GET]
      /{task_name}      # retrieve/create/delete a specific task [GET|PUT|DELETE]
                  /is   # retrieve/update the input sources [GET|PUT|POST]
                  /pf   # retrieve/update the data-processing function [GET|PUT]
                  /od   # retrieve/update the output destinations [GET|PUT|POST]
                  /lo   # retrieve/observe the last output produced [GET]
```

Figure 4.6: T-Res interface.

middleware virtualization capabilities have been integrated by using PyMite [221], a lightweight Python interpreter on which tasks and applications can be instantiated at run-time by uploading Python bytecodes. Python has been preferred to other commonly used interpreted programming languages such as Java and Javascript, because the former does not allow the definition of processing functions by means of scripts and the latter cannot be compiled into bytecode, and both these characteristics do not fit well in constrained scenarios. PyMite has been specially designed to be executed on devices with scarce resources, e.g., microcontrollers with limited Flash and RAM capabilities. The Data Serialization Library component provides data serialization services in the EXI format, a very compact representation of XML, which provides better performance in compressing M2M messages [222].

The in-network event composition is performed by the RPE module. Such a component is placed on top of the virtualization module, and it is based on T-Res [223], a framework enabling in-network processing in IoT networks.

T-Res comes with the CoAP-based RESTful interface depicted in Fig. 4.6, which allows to remotely instantiate, configure and delete tasks on a node. T-Res tasks are defined by their input sources (`/is`), the data processing they perform (`/pf`), and the destination of their output (`/od`). According to the RESTful paradigm the input sources and the output destination are CoAP resources specified by means of their Uniform Resource Identifiers (URIs), while the processing function resource `/pf` contains the Python bytecode of the task to be executed. It is possible to dynamically modify the behavior of a task by invoking the proper CoAP methods on such resources. Specifically, it is possible to change the processing function bytecode of a task simply performing a PUT on the relative `/pf` subresource providing the new bytecode. By allowing the run-time instantiation of tasks as Python bytecodes, the event processing function can be moved at run-time from one node to another, e.g., to a node having a better connectivity or a larger residual energy, completely decoupling the source of the information from the single physical sensor. The Configuration Manager is the component responsible for the global configuration of the ICSI Middleware. Each configurable component is registered through the RESTFUL Web Service and provides the APIs to allow remote configuration. This component provides access to a wide range of functions, including: i) RPE configuration (input/output resources and processing); ii) Over-the-air software update; iii) Energy Policy Reconfiguration (duty cycle configuration); iv) Networking and general purpose maintenance features. The ETSI M2M library uses the services provided by the data serialization libraries and exposes a set of APIs to encode/decode M2M messages according to ETSI specifications.

4.4 Results

The proposed visual sensor network was tested separately for the vision and networking parts in order to validate each single logical component; then the prototype was actually installed in the city of Pisa for an extensive field trial.

4.4.1 Evaluation of embedded vision logics

In this section, tests and evaluations of the algorithms for traffic and parking monitoring are presented. These first tests were performed in order to be able to deploy the algorithm on the actual visual sensor network, through an assessment of the performance. Processing of these data was performed off-line in lab, but an important note is that all the data used were real-time acquired data from the real world.

4.4.1.1 Parking lot availability tests

Following the description of the background modelling given in 4.3.2.1, in Fig. 4.7 the background model at a specific frame t of the acquired test sequence is shown (Fig. 4.6a) and its corresponding processed output (Fig. 4.6b). In the processed frame the red colour means *busy* because the belief of occupation is very strong due to the high number of edges and low asphalt detection inside the RoI. On the contrary, green parking slots indicate the status *available* because a very large area of asphalt is found within and few pixels come out from the edge detector. The blue colour expresses *uncertainty* because the combination of edge and asphalt beliefs is between the two thresholds of the hysteresis. In this latter situation, no change occurs in the status, i.e., the final output remains the same of the one given for the previous frame. There is a difference between the right-most blue slot and the others: in the bottom right slot, there is such an uncertainty because the vehicle has just arrived and it is not integrated in the background yet as can be seen from the picture on the left. The asphalt detection considers the background image (for stability reason), so only when the vehicle is part of it (i.e., after about 30 frames) the colour will change to red. The uncertainty condition of the other three slots are due to the partial occupation of the slot by entities that are not big enough to be classified as a vehicle (e.g., a person walking in and parts of vehicles from alongside slots).

In order to evaluate the performance of the method various video sequences were acquired during different times of the day (i.e., with different light conditions) and a comparison was made between the log of algorithm output versus a manually annotated ground truth, i.e., acquired frames were manually pointed out when a status change happened, that is an available parking lot became busy or the contrary. The Overall Error Rate (OER) is the metric proposed in [188] and is defined as the ratio of total errors (i.e., False Positive plus False Negative) and the total responses (i.e., False Positive plus False Negative plus True Positive plus True Negative):

$$\mathrm{OER} = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{FP} + \mathrm{FN} + \mathrm{TP} + \mathrm{TN}} \tag{4.5}$$

Notice that in an outdoor scenario (which is the setting of the experimentation carried out in the chapter), even in absence of parking status changes, there are a number of non stationary elements that might interfere with the algorithm. These include for instance light changes, casted shadows and temporary occlusions. Furthermore in order to compare the system with others that operate at single frame level (their output is the state of each slot based on the analysis of a single image) it seems reasonable to consider every single frame output as the complete set of the slots' status; thus the denominator of the ratio corresponds to the multiplication of the slots monitored and the total frames considered:

$$\text{ErrorRate} = \frac{\sum_{i=1}^{\text{TotalFrames}} (\text{FP}_i + \text{FN}_i)}{\text{TotalFrames} * \text{TotalSlots}} \tag{4.6}$$

where $\text{FP}_i + \text{FN}_i$ is the total number of errors made when analyzing frame i.

In Table 4.1, the results of several sample acquired sequences from separate cameras are presented.

Table 4.1: Performance of parking lot monitoring of 5 separate devices

	Monitored slots	**Total frames**	**False Hit events**	**Missed events**	**Total FP**	**Total FN**	**ERROR RATE**
CAM A	23	5357	10	24	238	594	0.675%
CAM B	22	5145	8	22	285	693	0.864%
CAM C	17	5260	7	14	156	396	0.617%
CAM D	16	5225	6	8	222	269	0.587%
CAM E	15	5305	6	5	211	197	0.513%

(a) (b)

Figure 4.7: Example of parking lot analysis: background model at time t (a) and real-time output at time t (b).

The average error rate based on five different cameras is 0.65%. Although a number of papers in literature has been presented on the topic, an accurate comparison is difficult to be proposed,

Table 4.2: Comparison of related work

Reference	**Error rate (%)**	**Features**
Wu et al. 2007 [224]	6.5	*Color*
Sastre et al. 2007 [185]	2.2	*Gabor filters*
Bong et al. 2008 [225]	7.0	*Color*
Hichihashi et al. 2009 [226]	2.0	*PCA*
Huang and Wang 2010 [180]	1.2	*Color*
DeAlmeida et al. 2015 [188]	0.4	*Texture*
Amato et al. 2016 [190]	0.4	*CNN*
Proposed method	0.65	*Edge and color*

since most of the approaches are far from being based on a "low-cost low-consumption embedded platform", yet their performance are not radically different from the one achieved by our method. A comparison with some related work can be found in Table 4.2.

In particular, up to the best of our knowledge [188] is considered as the most performant approach in the state of the art; it is based on complex features (LPB and LPL) and it is target for high end computer. Recently [190] showed good result using deep learning on a smart camera based on Raspberry Pi platform; nevertheless the training phase of the network has been executed offline with performing hardware and the necessary memory size to use the Convolutional Neural Network is ten times the one available on the visual sensor node proposed in this chapter; furthermore the system proposed in [190] needs about 15 seconds to analyze an image and calculate its output, while our system achieves the result in 2 seconds.

4.4.1.2 Traffic flow monitoring tests

Traffic monitoring data were acquired using a temporary installation of the sensors on the same site of the final deployment, so that test sequences acquired reflected the real conditions and traffic typology. In the following Fig. 4.8 an image acquired from one of the installed sensors shows the monitored road with the RoIs highlighted for each lane. For the upper lane, vehicles are first passing over R_1, then after a time t they will pass over R_2. The distance between the two RoIs is fixed and known, thus a computation can be performed to establish the length and speed classes of each vehicle following the algorithm presented in 4.3.2.2.

In Fig. 4.9 the example of a detected vehicle with the corresponding RoIs highlighted is shown.

The frame rate of the camera sensor is known and, considering the width of the RoI, it allows to catch the entrance of a car and the empty space between two passing cars with a safe margin. Thus, each vehicle can be detected individually, and independently from its length. As reported in the algorithm in Fig. 4.3, the main variable is the time t occurring between the transition from occupation of R_1 to occupation of R_2 and then to the exit of R_2. The analysis of the two RoIs is performed independently, so that one can result in an *occupied* status while the other may result

Figure 4.8: Traffic flow analysis: view from sensor test set-up and example of vehicles transit in the field of view of the sensor, which may cause occlusion to the upper lane.

(a) (b)

Figure 4.9: Traffic flow analysis: detected vehicle from sensor test set-up (a) and the same frame processed with the RoIs highlighted (b).

free.

The RoI for the traffic monitoring are two for each lane, and the distance among each RoI in the same lane is measured at the road surface level and used in order to compute the vehicles speeds and length classes. Three speed classes and three length classes were used. Test sequences have been acquired in real traffic conditions and then used for testing the algorithm. The ground-

truth total for these sequences was the following: 124 vehicles transited (70 along the lower lane, 54 on the upper lane) and having the following length classes: 11 with length between 0 and 2 metres (7 lower lane, 4 upper lane); 98 with length between 2 and 5 metres (55 lower lane, 43 upper lane); 15 with length 5 and more metres (8 lower lane, 7 upper lane).

It is important to note, for the following results, that the lengths were inferred on the basis of the recognition, by a human observer by sight, of the specific car models. Moreover, the algorithms compute a speed class estimate, for this data the ground truth is based on preliminary tests made on cars which were equipped with a GPS and recorded their own speeds. The total classification results are shown in Table 4.3.

Table 4.3: Classification performance of the traffic flow monitoring.

	Total	**Lower lane**	**Upper lane**
Total transited vehicles	124	70	54
Correctly identified vehicles	118 (**95.2%**)	69 (**98.6%**)	49 (**90.7%**)
False positives	3 (**2.4%**)	1 (**1.4%**)	2 (**4%**)

As it was expected, the results on the upper lane were slightly less precise; nevertheless the global performance for this test has to be considered positive, yielding a percentage of more than 95% of correct detections. Furthermore, a classification for the length classes is reported, where the 3 nominal classes for length are shown in Table 4.4. As already stated, the lengths have been extracted from car manufacturer data. Another test was made regarding speed classes, but in this case we did not have a complete ground-truth data because the estimates were feasible only by eye-sight, thus we did decide not to show the outcome of this test.

In order to show the relevance of our proposal, we report, in the following Table 4.5, a comparison with other state of the art algorithms, evaluated with respect to correct identification

Table 4.4: Example of length classes classification (lower lane).

Lower lane	Ground truth	Correct Class.	False Pos.	Efficiency
Length $0 \div 2m$	8	8	0	100%
Length $2 \div 5m$	57	56	1	96.6%
Length $5 + m$	5	5	0	100%
TOTAL	70	69	1	97.2%

Table 4.5: Comparison of performances and computational power for different algorithms.

	Performance	**Hardware/Processing notes**
Messelodi et al. 2005 [198]	82.8%	*Hardware not reported* **not an embed. platform**
Ottlik&Nagel 2008 [199]	83%	*Hardware not reported* **off-line processing**
Saunier&Sayed 2006 [200]	88.4%	*Hardware not reported* **off-line processing**
Chen et al. 2011 [201]	97%	$DM642$ **DSP-based embed. platform** 600**MHz-**32MB **DRAM**
Rad&Jamzad 2005 [202]	96%	**Pentium II** 800**MHz processing 11fps**
Lai et al. 2008 [203]	89%	*Hardware not reported* **but processing** 2.5**fps**
Semertzidis et al. 2010 [204]	91%	*Hardware not reported* **but processing <**5**fps**
Chen et al. 2012 [205]	96%	*Hardware not reported* **off-line processing**
Proposed method	95.2%	**ARM Architecture** 454**MHz-**64MB **SDRAM**

rate (i.e., a performance index), and the computational power used for these solutions. Obviously, an effective and complete comparison is impossible, due to the different tasks, goals, algorithms, case studies used by each cited work.

4.4.2 Evaluation of middleware capabilities

This section reports the ICSI Middleware performance evaluation in a laboratory testbed. The main purpose of such a campaign is to evaluate the middleware capabilities in a controlled environment. In the following of the section, the laboratory testbed is first described, then the middleware performance are reported in terms of event notification delay, by considering both event composition and data encoding techniques required for the transmission of M2M messages. Moreover, since ideally the middleware is targeted to constrained devices, a feasibility assessment is reported considering the actual networking component board introduced in Section 4.3.1.

The testbed setup is reported in Fig. 4.10. It is mainly composed of three visual sensor nodes, a border router (BR) device able to gather data from the roadside network, a proxy able to translate CoAP messages in the HTTP format, and system able to receive HTTP messages from the proxy while supporting the Gateway Service Capability Layer (GSCL) component defined in the ETSI M2M architecture. In the picture the arrows labeled with the `Observe` method describe the monitoring relationships involving the three visual sensor nodes. In detail Node A is

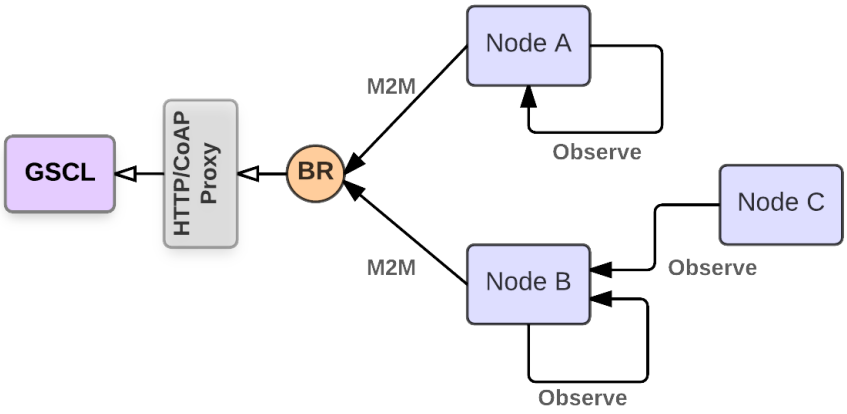


Figure 4.10: Experimental setup in the laboratory testbed.

configured to monitor only local events, while Node B monitors event locally and receives event notifications generated by Node C. In Node B an event composition is performed, the basic events are aggregated by the RPE and sent to the GSCL through the proxy as new aggregated resource. The proxy and the GSCL are hosted on the same laptop. The measured transmission latency in event notification is reported in Table 4.6, it includes the time required to: (i) compose the event, (ii) compress and send the event as M2M message, (iii) convert the in-network message in the HTTP format before sending it to the GSCL component. In the table the overall event notification delay is reported as a function of the message dimension, considering the number of data packets in which it must be divided. As it is easy to expect, the delay increases as a function of the message dimension, and it is mainly due by transmission and encoding latencies, while with the implemented event composition logic (weighted averaging function) the RPE delay is negligible. Increasing the size of the message bigger delays can be experienced. However, the resulting delay is fully compatible with the dynamics of the considered ITS applications.

Table 4.6: Event notification latency.

Message Size [bytes]	Message Blocks [#]	Sensor to GSCL [ms]
104	2	176.01 ± 0.27
155	3	227.79 ± 0.42
228	5	287.06 ± 0.38

The middleware feasibility assessment of the proposed middleware solution has been analyzed for the networking component device, the SEED-EYE board [210]. Such a device is characterized by 512 Kbyte of Flash memory and 128 Kbyte of RAM memory. The whole middleware requires

Figure 4.11: Picture showing the final field test installation.

44.80% of the ROM memory and only 26.50% of the RAM, leaving a significant space available for the user-defined event composition functions.

4.4.3 Experimentation in the field

The final deployment of the visual sensor network has been arranged in a specific area of the city of Pisa featuring an important commuter parking lot and a main road for accessing both the parking facility and the city center. In order to have an autonomous system, all the mounted sensor nodes were equipped with photovoltaic panels and batteries for long term duration.

In Fig. 4.11 two of the sensor nodes of the installed network are shown. It is important to notice that in the installation only normal poles were used for fixing the sensors that, being completely wireless, did not require any further work for cabling. The full installation required less than two days of work of a team constituted by 2 technicians and 2 computer scientists taking care of sensor fixing and configuration.

A long term monitoring has been performed on both traffic flow and parking lot monitoring, the results span over a period of two months. Each of the test sets used different metrics and evaluation methods, in particular, the parking lot tests analysed the occupancy ratio daily rates and trends, while the flow monitoring tests analysed the traffic flow rate on daily and global basis as well as the average speeds and the cumulative vehicle count curves (i.e., N-curves).

4.4.3.1 Parking lot monitoring tests

A total number of 71 slots (66 regular, 4 disabled people, 1 e-car charging) were monitored by installing 12 sensor nodes. About 20 slots were in the field of view of more than one sensor in order to test and validate cooperative sensing functionalities of ICSI middleware. In particular, for each of these slots, event composition was performed aggregating the measures produced by each sensor in charge of its monitoring. As in Section 4.4.2, weighted average was used in aggregation: each sensor contributed to the average with a weight proportional to the area in

pixel of the region in the image corresponding to the monitored slot. By evaluation of the log of the network, it resulted that event composition leads to a reduced number of fluctuation, allowing to filter out events due to temporary occlusions in the field of views. In addition, event composition allowed to cope with failure of one of the nodes. Indeed, tests have been executed switching off artificially a sensor in the network. The visual sensor network was able both to reconfigure itself building new routes when needed as well as for publishing aggregated notification regarding parking lot status.

In the following Table 4.7 a resume of the cooperative monitoring performance is reported. The monitoring regards the slots which are surveilled by two sample cameras (i.e., 12 slots), comparing each individual camera results (i.e., columns CAM A and CAM B) only for the slots monitored by both, versus the cooperative weighted results (i.e., shown in the column COOP.). As it can be seen the worse results from single cameras are heavily increased, as the error rates drop from 6% and almost 5% down to only 1% of errors, counting on a total of 78840 events on which the occupancy detection algorithm is called, computed as the product of every slot in each acquired frame.

Table 4.7: Cooperative monitoring results.

	CAM A	CAM B	COOP.
ERRORS	4870	3674	804
%	**6.2%**	**4.7%**	**1.0%**

Besides these technological consideration, the network was able to collect useful data for understanding the dynamics of parking lot access and usage. A typical working day scenario is reported in Fig. 4.12 where the occupancy status recorded every 15 minutes is shown for each of the slots categories (i.e., Reg-Occ: regular slots; Dis-Occ: disabled people slots; EC-Occ: e-Charging slots; TOTAL: Total % of available slots).

A more extensive assessment of these results is given by the monthly aggregated data, for the whole months of November and December until the last day of the year. Obtained results confirm the usage of the selected parking lot as a typical long-stay swapping parking used also for mobility exchange reasons (e.g., fast bus stop link to centre and train station). In fact, the occupancy ratio reaches its highest values early and quickly in the morning and only decreases slowly in the afternoon to get to its minimum close to the office closing time. Another relevant information regards weekends and Christmas holiday period (last week of December) low usage of the parking (i.e., never exceeding 35% of the capacity). One more suggestion comes looking at the peculiar behaviour during the week before Christmas (i.e., Dec. 21^{st}-23^{rd}); there is yet a decrease in the total occupancy, and an increase in the latter time slots (e.g., 14-18), a possible

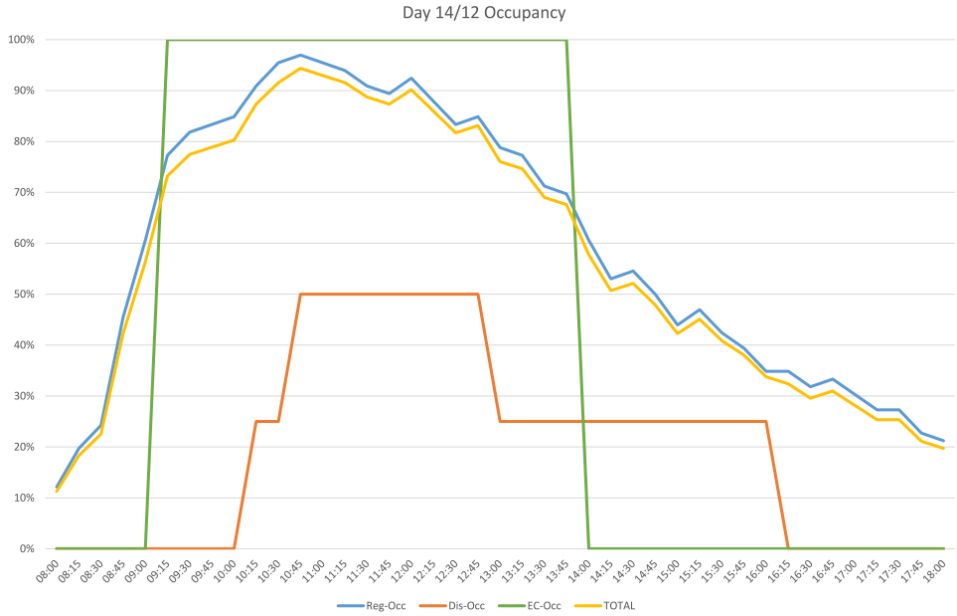


Figure 4.12: Occupancy status of the parking on December 14th.

explanation of this can be inherent to a higher flow of people staying longer in the city centre for Christmas shopping.

4.4.3.2 Traffic flow monitoring tests

The long term flow monitoring tests were performed along Via Pietrasantina (the north access way to the city of Pisa). Data acquired covered different aspects: the amount of vehicles, the average speed categories detected, and the aggregation of these data on time slots and daily basis. The first data analysed is the traffic flow rate on a single working day. Data are evaluated on a 15 minutes basis.

As it was expected the flow ratios, in a normal working day, have the highest values in the early morning hours to quickly decrease at around 10:00, to increase again during lunch-time (i.e., that is also around closing time of schools). A rapid decrease happens around 14:00, increasing again but to a slightly lower value, after 16:00 for the closing office time, which is more distributed in time (e.g., spanning from 16:00 till 20:00). Another interesting data is the total amount of vehicles: in that day the amount was 2749, which considering the over 95% correct identification ratio confirms to be in range of the typical working day data available from the Pisa mobility agency.

The comprehensive results obtained during the two months monitoring are given, by means of monthly views of the vehicles-per-hour data aggregated by time slots (example in Fig. 4.13 showing the month of December). A brief analysis of the results brings to some remarks: differently from the parking lot tests, the traffic flow shows large decrease only on Sundays and

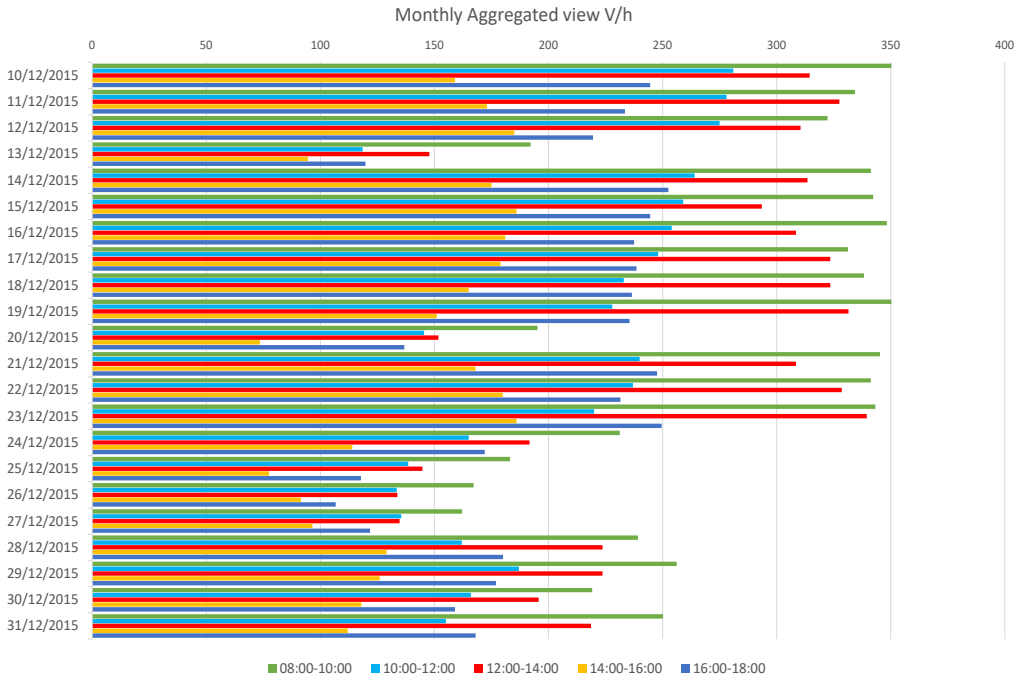


Figure 4.13: Aggregated vehicles per hour data by time slots for the latter 3 weeks (reduced only for a better readability) of December 2015.

holidays, while Saturdays see a traffic flow comparable to other working days. Sundays and holidays (i.e., 8th December, 25th and 26th December) report the average traffic flow ratio to drop more than 50%. Moreover, alike the parking lot scenario, during Christmas holidays (i.e., after 23rd December) there is a decrease around 30% of the traffic flow.

4.5 Conclusions

In this chapter a prototype of visual sensor network has been presented where each camera-equipped node embeds special vision logics to understand urban mobility and extract relevant real-time data for its analysis. Globally, the network is endowed with an IoT middleware that enables cooperative sensing by offering the possibility to perform event composition at network level. In this way, the network is insensitive to hardware failure and to temporary occlusion in the field of view of some nodes. The capability of the network has been demonstrated in a field test that highlighted the suitability of the proposed solution in dealing with parking and traffic flow monitoring and providing high quality real-time information. Indeed, the network was able to capture variations in urban mobility pattern produced under special circumstances such as festivities, which would have been difficult to collect at large scale without an IoT solution as the one proposed in this chapter. Furthermore the visual sensor network can support the collection of data in other smart city contexts simply by extending the already developed vision logics.

CHAPTER 5

STANDARDIZING SOFTWARIZED IOT ECOSYSTEMS

5.1 Introduction

In recent years the continuous advancements in the electronic field, as well as the development of new cost-effective wireless communication systems, have fostered the so-called IoT vision [227]. According to the IoT concept, in the near future billion of smart objects [228] will be connected among them following the Internet paradigm while exploiting Machine-2-Machine (M2M) communications. In an M2M scenario, all devices will be able to communicate and cooperate with each other without human intervention, thus creating effective and advanced smart environments. Smart Cities [229], Smart Homes [230], and Smart Factories [231] will be able to improve human lifestyle in every sphere.

In the above depicted scenario, the actual trend is to make IoT objects more and more intelligent leveraging on a new class of high-performance and low-power microcontrollers being developed in greater numbers by silicon producers. The capability of such devices, usually placed at the leaves of the IoT network, to execute complex algorithms enables the so-called Fog Computing approach, in which the intelligence of an IoT system is distributed in all the architectural elements [232]. Fog Computing is just a facet of softwarization, the evolutionary trend which is impacting the telecommunication world nowadays. Softwarization is leading to an increasing decoupling between hardware and software, to a pervasive use of virtualization technologies and to a more blurred distinction between the core network and the terminals [233]. Considering IoT objects as effective and intelligent components of the global Internet network, it is desirable to bring the softwarization trend even to these nodes, making them able, among the other things, to change at runtime their application logic. Moreover, such objects need to expose the result of the computational tasks they are executing, and the tasks themselves, as standard resources provided by embedded web services [234], thus following a particular standard aiming

at harmonizing M2M interactions. A powerful method to enable application logic reconfigurability in IoT objects is the adoption of a virtual-machine based design in which execution tasks can be abstracted as network resources. In this way, the running logic can be easily changed through RESTful methods implemented by well-known application protocols (e.g., HTTP, CoAP). Virtual-machine based methods are considered a strong enhancement with respect to over-the-air (OTA) programming techniques, the main solution in enabling device reconfigurability through the change of the whole firmware or particular modules [235]. While OTA techniques have been already proposed in conjunction with OpenFlow to enable softwarization [236] in IoT systems, the use of virtual-machine based techniques has not yet proposed. Considering IoT objects based on microcontrollers, a powerful framework implementing the above described functionality enabling a Task Resource Abstraction approach able to change at runtime the application logic is T-Res [237]. Such a framework, proposed by Alessandrelli et al. in 2013, consider the use of a Python interpreter in microcontroller-based devices, as well as CoAP as application protocol implementing the RESTful paradigm. Even though the use of T-Res has been already envisioned in M2M networks [238] its adoption and integration in recent standardized M2M architecture has not yet been proposed, as well as its real feasibility in new generation devices. In this chapter we consider T-Res as a framework capable of enabling softwarization in IoT objects. Thus its integration in the oneM2M [239] architecture is presented and discussed before evaluating its feasibility in terms of application logic execution time and power consumption by using the SensOne board [240], an innovative device designed to be used in pervasive IoT systems with energy saving requirements.

The rest of the chapter is organized as follows. Section 5.2 first summarizes T-Res and oneM2M, then it discusses the integration of T-Res in the oneM2M architecture. In Section 5.3 the evaluation of the oneM2M compliant T-Res is discussed by reporting measured execution time, power consumption and memory footprint values when several algorithms are executed. The T-Res approach is compared with a classical approach in which a monolithic firmware runs on the selected device. Conclusions are drawn in Section 5.4.

5.2 Enabling softwarization in the IoT

In order to bring softwarization to the most remote and constrained leaves of the network, a key feature that each node should support is the ability to dynamically change its behaviour (i.e., the adopted routing policies, the application logic in execution, etc.) at runtime. Moreover, this change should be able to be triggered without the direct involvement of any human, due to the fact that the exploding number of connected devices will increasingly be managed by autonomous service logics running on the Cloud. For this reason, standard methods which enable automatic node management and reconfiguration should be defined. To this end, in this chapter we take into account the oneM2M architecture since it represents a promising and wide supported effort to standardize M2M communications. Moreover, we propose a T-Res based implementation of the

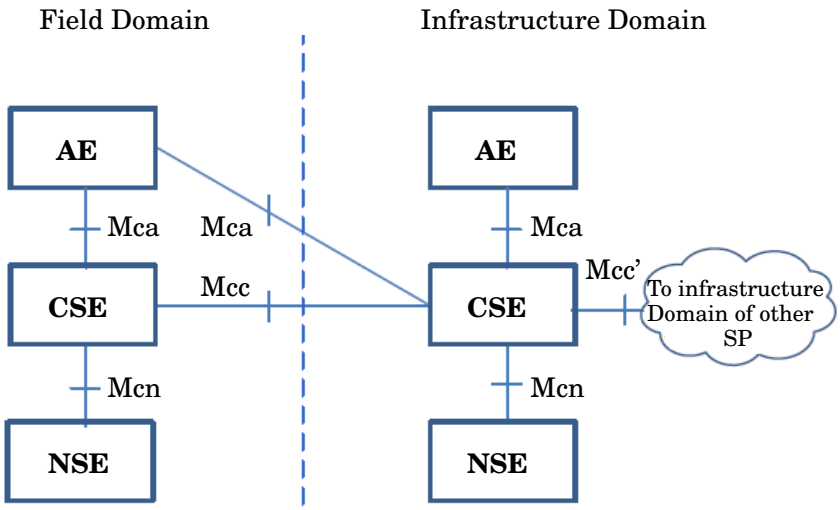


Figure 5.1: oneM2M functional architecture.

aforementioned functionalities, which are essential milestones on the path to enable softwarized IoT scenarios.

5.2.1 T-Res

T-Res is a simple and powerful IoT framework, already described in section 4.3.3, which allows to easily reconfigure at runtime the application logic that is in execution on each node of the network in the form of simple Python tasks.

On nodes running T-Res, the administrator of the network can create, update or delete the hosted applications by performing the well-known CRUD (create, read, update and delete) operations on the appropriate REST resources (Fig. 4.6). Although T-Res enables node reconfiguration at runtime, it does not provide a standard interface to perform this operation. In the next paragraphs we first introduce the oneM2M architecture, then we discuss the integration of the T-Res framework in the architecture in order to provide application management functionalities in a standardized fashion.

5.2.2 oneM2M

oneM2M is a global organization that brings together some of the most preeminent standards development organizations (ETSI, CCSA, TTA, etc.) and industrial players (IBM, Intel, Cisco, etc.) in the world. The organization aims at defining architectural and Application Programming Interface (API) specifications to provide a common framework supporting a wide range of M2M and IoT technologies and applications.

Regarding the oneM2M functional architecture [239], reported in Fig. 5.1, this is a simple horizontal architecture that fits within a three layer model taking into account applications, services, and networks. The application layer, on top of the stack, hosts the Application Entities

(AEs), which implement the applications service logic and provide a standardized interface to interact with them. In the middle layer reside the Common Services Entities (CSEs), which are instantiations of a set of Common Service Functions (CSFs) that can be accessed by other CSEs and AEs. At the bottom of the stack, the Network Service Entities (NSEs) provides services to the above CSEs so that they can access the underlying network in an agnostic manner. Communication between the three layers occurs through dedicated reference points.

Among the CFSs that a CSE should provide, there is the Application and Service Layer Management (ASM) CSF. Such a CSF uses the Configuration Function (CF) and the Software Management Function (SMF) provided by the Device Management (DMG) CSF to allow the management of the AEs and CSEs (Fig. 5.2). In particular, the SMF provides lifecycle management for software components and associated artifacts (e.g., configuration files, scripts, etc.), thus being able to remotely configure the parameters and the application logic of the tasks in execution on the end nodes.

5.2.3 Integrating T-Res in oneM2M

All entities in oneM2M, including AEs and CSEs, are represented as resources. In this respect, T-Res fits nicely in the oneM2M architecture since everything, even the tasks to be executed, is abstracted as a REST resource. However, the REST interface that a oneM2M node must provide is significantly different from the one exhibited by a T-Res node, and a careful adaptation is required.

The root of the REST interface that each oneM2M node must exhibit is the CSEBase resource, which represents a CSE. A partial representation of the structure of the CSEBase resource is depicted in Fig. 5.3. Through the CSEBase resource and its subresources it is possible to reach the AEs registered with that CSE and interact with them in a standard way. Subresources of the CSEBase resource are, among the others, the AE resources and the mgmtCmd resources. AE resources hold information about the AEs registered with the CSE, while mgmtCmd resources

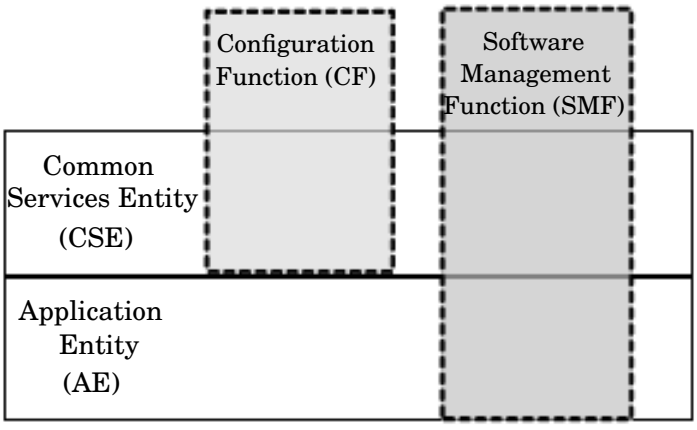


Figure 5.2: Application and service layer management functions.

model management procedures to be executed on the node. By acting on specific subresources of the mgmtCmd resource, it is possible to pass parameters to the command, trigger the command and monitor its execution.

Starting from these observations, we propose the following solution to keep exploiting the power of the T-Res framework and, at the same time, to have it integrated into a standard architecture. The main idea is not to allow the direct interaction with the T-Res interface, but to make all control flows pass through the oneM2M interface, as depicted in Fig. 5.4. In particular, we envisage the definition of different mgmtCmd resources, one for each operation that is possible to perform in the T-Res framework. Thus, for example, a mgmtCmd resource of type "Create Task" should be defined so that it accepts as input all the parameters needed to create a T-Res task, namely the name of the task, the input and output resources and the processing function. When triggered, the command would take care of all the necessary T-Res operations to correctly instantiate a task, plus other oneM2M specific actions, such as, for example, register the new application with the CSE by creating the correspondent AE resource.

In the proposed solution we decided to keep separated the oneM2M and the T-Res parts to preserve the modularity of the system. The management commands in the oneM2M interface do not directly invoke the T-Res functions for tasks management, even if it was in theory possible since the two software components reside on the same node. Instead, the management commands interact with the T-Res REST interface by performing normal CoAP operations. According to this design choice, when triggered, the above mentioned "Create Task" oneM2M management command should perform a PUT operation on the "task" resource of the T-Res interface. This operation is necessary to instantiate a new task resource inside T-Res and initialize the corresponding data structure. Moreover, all the necessary PUT operations on the proper subresources of the newly created task resource must be performed to define its input and output resources and processing function. If any error occurs at the T-Res level (e.g., there is no space to allocate another task) it should be reported at the oneM2M level to allow the system to take the

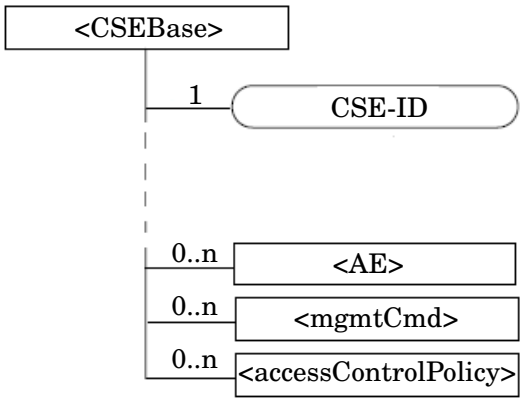


Figure 5.3: CSEBase partial structure.

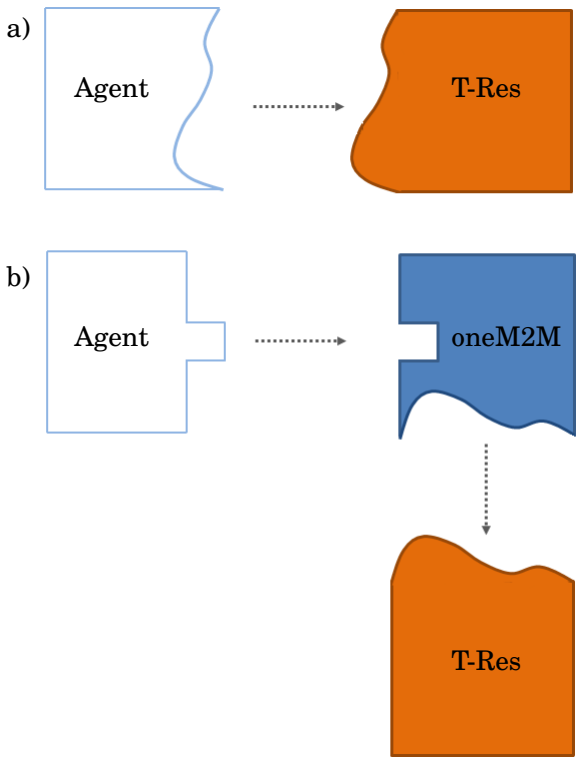


Figure 5.4: a) The agent must know the T-Res interface to perform the desired action. b) The agent is agnostic about the T-Res interface and can interact with it through the standard interface provided by oneM2M.

proper remedial actions.

All the other T-Res operations can be translated into oneM2M-compliant commands in the same way, thus making T-Res an effective solution to implement the oneM2M application management layer and support softwarization in constrained IoT networks.

It is worth noting that the approach we followed to embed T-Res in the oneM2M architecture could be successfully used also for the integration and standardization of other already existing solutions. In this sense, therefore, an important contribution of this chapter is to present a general method to retrofit legacy frameworks to make them oneM2M compliant.

5.2.4 Security considerations

The possibility to remotely change the execution logic on the nodes at runtime rises serious security issues. In fact, the lack of a proper security infrastructure could lead to dramatic scenarios with thousands of compromised nodes executing malicious code. T-Res has no built-in security functionalities, but since it makes use of a CoAP interface, it could easily take advantage of security protocols, such as the Datagram Transport Layer Security (DTLS) protocol, a solution providing end-to-end privacy and authentication. In addition, the oneM2M architecture also

allows performing a more fine-grained control by putting in place access control mechanisms. Each oneM2M resource, including the mgmtCmd type, can, in fact, be linked to a resource of type accessControlPolicy through which it is possible to define access control rules. An evaluation of these technologies will be addressed in future works.

5.3 Performance evaluation

Even though the integration of T-Res in the oneM2M architecture is able to provide a powerful standard framework aiming at enabling network softwarization in the IoT, its real feasibility in new generation devices must be assessed. In [237] such feasibility has been proved in terms of memory requirements by considering the WiSMote platform [241]. In this work, the execution and power consumption evaluations are performed with the SensOne board [240]. The SensOne is an advanced device specially developed for battery powered wireless sensors networks. It features a low-power 32MHz ARM Cortex-M3 by Texas Instruments with 512kB of flash memory and 32kB of RAM. This platform provides enough memory and computational resources to execute the lightweight PyMite environment for complex middleware solutions with communication capabilities based on the IEEE802.15.4 standard.

In this section, the proposed virtual-machine based approach is compared to a C based counterpart. The goal of these comparisons is to assess the feasibility of the oneM2M oriented T-Res for a real deployment. To this end, three example algorithms, characterized by different complexity levels, and chosen from *dada's perl lab* [242], are considered: (i) multiplication between two 5x5 matrices *mat*, (ii) heapsort of a 100-elements vector *heap* and computation of the first 17 Fibonacci numbers *fib*. These functions have been implemented as a Python snippet run by T-Res and as a native C-code application.

In Table 5.1, the applications are compared in terms of used Flash and RAM sections. Although not always of utmost importance in modern microcontrollers, the memory consumption becomes relevant for code maintainability in large scale deployments. In our tests, the Flash and RAM footprints have been quantitatively measured by analyzing the relative executable file for each case. More in detail, both implementations have been executed onto the Contiki OS [243], and in Table 5.1 only application specific footprints are considered, while libraries and run-time call stack are excluded. The Flash footprint considers the *.text.** linker sections relative to each application case. The RAM footprint, instead, has been evaluated as the maximum allowable *.stack* section plus the un/initialized *.bss* and *.data* sections. The T-Res occupation memory results are significantly higher, especially in terms of Flash usage with respect to the C implementations. The footprint increments are mostly due to the PyMite environment overhead, which is loaded along with the programmable T-Res memory slots. According to Table 5.1, Flash and RAM requirements of T-Res implementations are equivalent throughout the test cases. Indeed, the installed Python scripts are self contained within the already allocated memory space, without any other memory requirement. Although the larger memory area usage, T-Res remains a feasible

Table 5.1: Memory footprint comparison.

	Function	Flash (kB)	RAM (kB)
T-Res	mat	61.0	25.1
	heap	61.0	25.1
	fib	61.0	25.1
C	mat	0.44	15.3
	heap	0.51	15.3
	fib	0.24	14.2

solution to be deployed into low-cost embedded platform with runtime Python script installation support. As Table 5.1 shows, in fact, T-Res just uses 12% of the available Flash and 78% of the available RAM.

Table 5.2 shows the results related to power consumption, which is evaluated by measuring the instant supply current during the CPU execution time. The measure has been performed by recording the voltage drop across a shunt resistor with a digital signal oscilloscope. Whatever the application being run, the CPU exits the low-power mode (32KHz) and uses the full clock source (32MHz), without frequency scaling technique. During the CPU execution time, an average supply current of 26.9 mA is measured till the execution ends. Therefore, the energy results shown in Table 5.2 depends exclusively on the execution time. Longer execution times result in greater energy consumption values. As expected, the execution time and the energy consumption are higher for the T-Res applications. The increase factor is 275, 146, 212 for *mat*, *heap*, *fib* respectively. In the *heap* execution, the performance gap is lower with respect to other cases due to the reduced computation required. However, when the T-Res application is not in execution, the energy consumption remains comparable with the low-power mode. This last point is particularly important when dealing with coarse periodic task activations, where CPU execution time is only a fraction of the low-power mode.

Table 5.2: Execution time and energy consumption. Voltage 3.3V, Average current 0.0269 A.

	Function	Exec. time (s)	Energy (mJ)
T-Res	mat	0.0824	7.3150
	heap	0.1318	11.7000
	fib	0.9800	86.9900
C	mat	0.0003	0.0266
	heap	0.0009	0.0799
	fib	0.0046	0.4080

The overall results show an energy consumption 211 times higher for T-Res applications than C implementations. In our tests a preliminary lifetime evaluation of a node can be estimated

by considering a standard 18650 form-factor LiPo battery (1800mAh at 3.7V). Supposing a periodic task activation of 10 seconds the most time consuming T-Res *fib* function results in a battery lifetime of around 670 hours (27 days), which is sufficient for battery-supplied embedded monitoring solutions in different scenarios. In next generation IoT devices, which will feature hardware support for virtualization with lower-power consumption and energy harvesting technologies, the oneM2M/T-Res approach can be accounted for large scale deployments leveraging the softwarization trend.

5.4 Conclusions

Softwarization is a global evolutionary trend that is going to transform the world of telecommunications deeply. The ubiquitous use of virtualization techniques and the rise of the Fog Computing paradigm are symptoms of this technological revolution. Following this trend, and considering the necessity of supporting M2M communications in such networks, the chapter proposes the integration of T-Res, a virtual-machine based framework for programming abstraction, into the oneM2M architecture. This solution aims at providing constrained IoT nodes with the application management capabilities needed to push the softwarization wave to the very edge of the network.

The first part of the chapter presents the T-Res framework, and the oneM2M architecture. Moreover, the approach we have undertaken for the integration is discussed in detail. The second part analyzes the feasibility of the oneM2M compliant T-Res approach in real scenarios, by presenting the results of real experiments we conducted on an innovative and real IoT platforms. Obtained results show that, although the virtual-machine approach based on T-Res involves heavier energy and memory requirements than the traditional monolithic firmware approach, it is an effective way to allow application logic reconfigurability in IoT nodes, thus enabling their incorporation in the forthcoming softwarized ecosystems.

Part III

Secure Internet of Things

CHAPTER 6

A PRIVACY-RESERVING SMART CAMERAS BASED VIDEO SURVEILLANCE FRAMEWORK FOR USAGE CONTROL SYSTEMS

6.1 Introduction

Video surveillance systems are currently widely adopted both in business and public environments due to the need of security in critical environments. To this purpose, video surveillance systems with advanced features, such as humans tracking, have been proposed in the scientific literature, and they are already or they will be soon available on the market at affordable cost. Some companies adopt video surveillance systems to enhance the physical security of their premises by detecting unauthorized accesses. Video surveillance may be considered invasive for the privacy of the people, but it is needed as measure to guarantee people's security, for example to prevent terrorist attacks. Thus, effective video surveillance systems, besides simply detecting the presence of unauthorized people, should be able to preserve the privacy of authorized people who access the monitored environment. In addition, face recognition is heavily used in online photo albums and social networking platforms that have become popular for sharing photos with family and friends. These platforms support automatic detection and tagging of faces when images are uploaded.

This chapter describes a video surveillance framework based on the Usage Control model. It focuses on the prototype implementation and on the approach we propose to enhance the privacy preserving capability of our framework using the Secure Two-party Computation (2PC) cryptographic technique.

Our framework aims at detecting people located in the monitored environment, identifying them, and continuously evaluating the Usage Control policy to check whether they have the right

to stay in the monitored places. The Usage Control policy takes into account attributes, paired with such people and with the monitored environment, which change over time. Hence, the policy could be satisfied when a person enters the monitored environment, but a subsequent update of one or more attributes could cause a policy violation. So, as soon as the policy evaluation determines that one person is no longer authorized to stay in the monitored environment, an alert is raised, and the video stream is recorded.

The main contribution of this chapter is the design and implementation of a working framework adopting the Usage Control model in the video surveillance scenario: in this way, the security policy is continuously evaluated while people are in the monitored environment, instead of being evaluated only to decide whether they are allowed to enter it. As reference scenario, we suppose that a company wants to enforce a security policy in which visitors must be always accompanied by an employee when they are in some rooms of the company. This is a Usage Control policy that can be naturally expressed and enforced by the proposed framework, since it requires that a given condition, i.e., the presence of an employee in the room, is continuously verified while the visitor stays in that room. In addition, this chapter focuses on how the Usage Control model can be exploited to satisfy the security needs of the video surveillance scenario while guaranteeing users' privacy. To this aim, we propose two privacy-preserving solutions: the first is a soft solution that avoids to record the video stream when users are authorized by the Usage Control system to stay in the monitored room. Instead, the second solution enhances our usage-control system to privately compare faces using the Secure Two-party Computation (2PC) technique. Our solution can be used to extend the capability of the face recognition component by giving the possibility to authorize employees without knowing their real identity.

The structure of this chapter is the following: §6.2 describes some related work, §6.3 describes some background concerning the Usage Control model and the technique we adopted for face recognition. Our Usage Control approach is presented in §6.4, instead §6.5 details the architecture of our framework and gives some details of the prototype we implemented. §6.6 extends our video surveillance part presenting a privacy-preserving solution to identifies people. Finally, §6.7 draws the conclusions.

6.2 Related Work

The following works make use of visual or cryptographic techniques to protect users' privacy. The authors of [244] propose the adoption of a PrivacyCams to encode video stream directly by cameras. Sensitive details are hided in the stream and only authorized people can access the raw data. More specifically, they suggest the adoption of level of authorizations to enable users to unveil sensible details depending on their authorization grant. Fidelao et al. [245] propose a privacy architecture for sensors, and in particular for video surveillance cameras. They suggest the use of a privacy buffer to detect and tag private data coming from sensors.

Tagging of data is done by means of privacy filters that label the data as private or not private. Korshunov at al. in [246] borrow the warping algorithm from the animation and artistic fields to protect privacy of users in the video stream. The warping algorithm shifts Pixels into slightly different locations making the original face very difficult to be identified. Sohn et al. [247] propose something close to Korshunov. In fact, they use the JPEG Extended Range (JPEG ER) technique to scramble users' face on surveillance video content. Their results demonstrate how JPEG ER provides privacy-sensitive face regions giving a good level of protection in a stream of 30fps with a resolution of 4CIF, i.e., $704x576$ pixel. Instead, Saini et al. [248] propose an interesting study on privacy leakage of a streaming coming from multiple cameras. The model they present establishes the privacy loss due to three main features: *what* (i.e., activities), *when* (i.e., time), and *where* (i.e., location). Moreover, the same authors propose a framework that implements anonymous surveillance by decoupling the knowledge on the context from the recorded video and requiring that the video are monitored remotely [249]. Serpanos et al. in [250] face security and privacy issues in distributed smart cameras. In particular, they point out privacy and security properties that video-surveillance systems should provide starting from the architecture set up. Winkler et al. in [251] presented on a survey on video-surveillance in which the authors discussed the state of the art in security and privacy protection, and show how these works cover security aspects into data-centric, node-centric, network-centric, and user-centric security.

The following works specifically use Secure Two-party Computation techniques to perform face recognition operations in a privacy preserving fashion. In [252] and [253] the authors use secure two-party computation and homomorphic encryption schemes to make the well-known face recognition algorithm Eigenfaces privacy preserving. In particular they use the Paillier additively homomorphic scheme [254] to make the server compute the encrypted values of the euclidean distances between the sample face image and all the face images stored in a database without disclosure of information between the client and the server. In [252] the minimum between the encrypted distances and the relative face image is found through the Damgård, Geisler and Krøigaard [255] [256] cryptosystem, while in [253] the authors use a garbled circuit. In [257] face images are represented as 900 bit vectors encoding spatial and appearance information, and the matching algorithm consists in finding the face image(s) in the database at Hamming distance from the sample face image below of a certain threshold. In this scheme client and server compute the encrypted value of the Hamming distance between each face image in the database and the sample image using additive homomorphic encryption and then perform oblivious transfer to figure out whether such distance is below of the predetermined threshold. Homomorphic encryption is also used in [258] to achieve privacy preserving image retrieval in a cloud computing environment and face recognition using SIFT.

The following works make use of policies to enable actions in specific conditions. Wickramasuriya et al. [259] suggest a solution that hides camera objects to keep users' privacy by specifying policies to access regions of the video stream. They also localise and authenticate users

by means of RFID tag that, however, can be seen as a weak point since they may undergo identity transfers. Birnstill et al. in [260] make use of usage-control system to define policies that are differently evoked on two operational modes and regulate the actions to execute the video stream. In particular, the default mode occurs in a normal/quite situations, and here an operator is not able to see the video stream coming from the cameras. While, on the alarm mode the camera shows the video stream to the operator.

Differently from the above papers that make use of policies, our work propose a video-surveillance framework based on the Usage Control model with two solutions to preserve the privacy of the identified people. As far as we know, works as that one of Birnstill et al. in [260] use policies that are enforced to trigger actions to enable or disable the video recording. So, the alarm mode is not triggered according to an attribute state evaluated by the policy, but it is triggered by other external components. Then, the policies only manages the access to the video stream. Even in the work of Wickramasuriya et al. policies regulate the access to different region of the video stream, and they use RFID as authentication system. Instead, we propose a video surveillance framework that entrusts the management of users' identity to a face recognition algorithm in a privacy-preserving fashion, and uses Usage Control policies which take into account attributes of the users, of the monitored rooms and of the environment that change over time. In this way, besides preserving the privacy of the users authorized to enter the monitored rooms, our framework is also able to detect when a user who was previously authorized to enter a room should leave that room because of a policy violation cause by a subsequent attribute update.

6.3 Background

6.3.1 Usage Control Model

The Usage Control (UCON) model improves the traditional access control models with *mutable attributes* and new decision factors besides *authorizations*, i.e., *obligations* and *conditions*. This section summarizes the main concepts of the UCON model; a detailed description can be found in [261, 262, 263, 264].

Mutable attributes represent features of subjects and resources (objects) that can be modified due to the access decision process or to the usage of resources. This could impact the rights of other accesses that are in progress [265]. For instance, the number of people in a room, which changes every time a person enters or exits the room, is a mutable attribute paired with the room, which is the resource that is accessed. *Immutable attributes*, instead, do not change their value frequently, and their modification is only done through administrative actions. For instance, the role of a person in a company is an immutable attribute and can be updated after a career advancement (e.g., from employee to department head). To face the mutability of attributes during the usage of a resource, the Usage Control model evaluates the policy before *(pre-evaluation)* and during the usage of that resource (*ongoing-evaluation*). The policy re-evaluation during the

resources usage reduces the risk of their misuse when a given permission is no more valid.

Authorization predicates are needed to determine if a subject has the permission to access a specific object. To perform this check, the decision making phase uses subject/object attributes, and the actions on a object requested by a subject. The UCON model defines two categories of authorizations: pre-Authorizations (*preA*), here the decision phase is done when the subject requests to access the object, and ongoing-Authorizations (*onA*), here the decision phase is done while the access is in progress.

Obligation predicates state if specific requirements are fulfilled to access objects. In particular, Pre-oBligation (*preB*) predicates verify the requirements before the access, while ongoing-oBligations (*onB*) continuously check that the requirements are fulfilled.

Conditions are requirements that do not depend on subjects or objects. They evaluate environmental or system status, such as current time.

The UCON model has been successfully adopted in several scenarios, such as Web, Grid, or Cloud to protect the usage of several kind of resources. Sandhu et al [266] propose their model in collaborative computing systems, such as the GRID environment. Their model is based on a centralized repository for attribute management. Immutable attributes are managed in push mode (i.e., the attributes value is submitted to the authorization service by the user himself), instead the mutable attributes are managed in pull mode (i.e., the attributes value are collected by the authorization service just before their use).

The authors of [267] propose an Usage Control enforcement mechanism for applications. They propose a prototype to control data in a social network, and their mechanism allows the data owner to avoid that data would be printed, saved, copied&pasted, etc., from unauthorized users.

In [268], the authors propose the adoption of the Usage Control model to enhance the security of IaaS Cloud services. The proposed authorization system regulates the usage of the Virtual Machines provided by the IaaS service, and it goes beyond traditional authorization systems because it interrupts (suspends) the usage of running Virtual Machines when the corresponding rights do not hold any more.

The work in [269] tackles the problem of Usage Control of multiple copies of a data object when they are stored in distributed systems. Moreover, the Usage Control policies specifies the parties that will be involved in the decision process. In fact, a policy could be evaluated on one site, enforced on another, and the attributes needed for the policy evaluation might be stored in (many) different locations.

6.3.2 Face Recognition

Face recognition is one of the most common techniques used by humans for visual interaction. Face recognition is used in contexts like video surveillance, recognition of individuals in airports and border crossings, authentication and access systems. However, in these scenarios is often difficult to recognize faces, since faces can be affected by:

- aging;
- different facial expressions;
- changes in lighting and background;
- changes in position with respect to the camera;
- presence of glasses;
- partial occlusions of the face.

Despite these problems, in the last thirty years researchers have been active in designing and developing recognition systems as reliable and robust as possible to tackle the above issues. Face recognition methods are classified as: i) feature-based and ii) holistic.

6.3.2.1 Feature-based

Here, images are processed to have distinctive facial features such as the eyes, mouth, nose, and others. In addition, geometric relationships are calculated among those facial points to reduce the input facial image to a vector of geometric features. Finally, statistical pattern recognition techniques are applied to match faces.

Featured-based techniques are quite robust to position changes of the input image since features point are extracted a priori of the matching analysis of the image. However, the major disadvantage of these approaches is the difficulty to automatically detect feature.

SIFT. Scale-Invariant Feature Transform (*SIFT*) is an algorithm to detect and describe local features in images. These are used to obtain reliable matching between different views of the same object and for this reason they are invariant to scale and orientation. SIFT follows the following four step to extract features:

1. it localizes potential interest points in the image by detecting the maxima and minima of a set of Difference of Gaussian (DoG) filters applied at different scales all over the image;
2. it discards points with low contrast;
3. it assigns an orientation to each key point based on local image features;
4. it computes a local feature descriptor based on the local image gradient, and transforms the orientation of the key point to provide orientation invariance. Every feature is a vector of 128 dimension distinctively identifying the neighborhood around the key point.

To recognize a face with those others stored in a database, SIFT extracts the features from the available faces. Then, SIFT extracts the features from the new face and compares them with

Figure 6.1: Image transformation using Eigenfaces.

those ones found before. The face with the largest number of matching points is considered as the closest one. In particular, a feature is considered similar to another one when their distance is less than a specific fraction of the distance to the next nearest feature. This allows SIFT to reduce the number of false matches.

6.3.2.2 Holistic

Here, faces are identified using global representations rather than on local features of the face. Holistic techniques fall into two groups:

- Statistical: a 2D array represents the image and recognition is performed comparing the input face with all other faces stored in the database. A weak aspect of this approach is the classification in a high dimensionality space. For this reason, other techniques suggest the use of statistical dimensionality reduction methods to get the most meaningful feature dimensions before performing the recognition.

- Artificial Intelligence (A.I.): it utilizes neural networks and machine learning techniques to recognize faces.

It is important to point out that these techniques work under limited circumstances, like equal illumination, scale, pose, and so on, and in addition they are computationally very expensive.

As part of the Holistic techniques, we provide a brief overview of the SIFT and Eigenfaces algorithms since we use them within our video surveillance architecture for face recognition.

Eigenfaces. Eigenfaces is a facial recognition algorithm based on the dimensionality reduction approach of Principal Components Analysis (PCA). The basic idea is to treat each image as a vector in a high-dimensional space. Then, PCA is applied to the set of images to produce a new reduced subspace that captures most of the variability between the input images. The Principal Component Vectors (eigenvectors of the sample covariance matrix) are called the *eigenfaces*. Each input image can be represented as a linear combination of these eigenfaces by projecting the image onto the new eigenfaces space. Then, identification is performed by matching the faces in this reduced space. The algorithm consists of the following steps:

1. it normalizes the M facial images contained in the considered dataset to line up the eyes and mouths resampled at the same pixel resolution. All images must have the same dimensions H x W;
2. it represents the images as (H x W)-dimensional vectors and computes the average face vector;
3. it subtracts the average vector from each vector;
4. it constructs the matrix A with dimensions (H x W) x M, containing in its rows the vectors obtained in the previous step;
5. it computes the eigenvectors of A^TA, corresponding to the eigenvectors of the covariance matrix AA^T relative to the largest eigenvalues, which are the ones responsible for the most of the variance in the dataset;
6. it selects and normalizes the K most significant eigenvectors, that become the eigenfaces;
7. it projects all the faces in the dataset in this K-dimensional space.

In the recognize step, Eigenfaces projects the face that must be recognized into the K-dimensional Eigenfaces space. Then, the distance between the input face and all faces stored in a database is computed, the face with the minimum distance is selected, and if this distance is less than a fixed threshold, therefore a match is found.

6.4 Usage Control in Video Surveillance

We based our video surveillance framework on the Usage Control in order to be able to enforce security policies for the entire time that a person is inside a monitored room. When cameras detect a person in the room, this person is identified and the policy is evaluated. At first, the video stream is temporary recorded into a buffer. During the identification phase, faces are detected from the video stream and they are compared with those ones stored in the Face Database to find an identification match. When a person is properly matched, some attributes, which belong to him/her, are retrieved from an Attribute server. The role of the identified person in the company is an example of such attributes. Then, these attributes are used to perform the pre-evaluation of the security policy. If the result of the evaluation is “deny”, the policy states that the person cannot stay in the room, and an internal alarm is trigged to force the person to leave the room. In parallel, the temporary video buffer is permanently stored as well as the current video stream captured by the cameras. Otherwise, if the pre-evaluation result is “permit”, the temporary video buffer is discarded, the current video registration is blocked and the person can stay in the room. However, while the person is inside the room, the ongoing-evaluation of the policy is executed and since some mutable attributes may change, these may cause a policy violation. In fact, as

soon as the ongoing-evaluation of the policy results in a violation, the alarm is triggered, the video stream is recorded and the person is forced to leave the room.

Policy:	1
Target:	2
(o.id = "CED")	3
Rule-1:	4
target:	5
(a.id = "enter(s, o)")	6
pre-authorization:	7
("employee" ∈ s.role)	8
pre-update	9
(o.numEmployee++)	10
post-update	11
(o.numEmployee−−)	12
Rule-2:	13
target:	14
(a.id = "enter(s, o)")	15
pre-authorization:	16
("guest" ∈ s.role) AND	17
(o.numEmployee > 0) AND	18
pre-condition:	19
(e.workingTime = TRUE)	20
on-authorization:	21
(o.numEmployee > 0)	22
on-condition:	23
(e.workingTime = TRUE)	24

Table 6.1: Usage control policy example

The following example shows the advantages resulting from the adoption of a Usage Control based authorization system in video surveillance. We suppose that a company regulates the access to some critical rooms of its department requiring that guests can access them only during the working hours, and they are accompanied by an employee when they are in these rooms. This kind of policy cannot be easily expressed by traditional access control models because they allow to evaluate the policy at access request time only. Instead, the company requires that the authorization system continuously verifies that each person in the room is authorized to stay there. Thus, our framework is the right solution to enforce that policy since it is designed to continuously evaluate ongoing policies.

Table 6.1 shows a representation of this Usage Control policy in a human readable language.

The object (denoted by o) is one of the rooms of the company that the Usage Control system must protect, and the subjects (s) is the person who enters the room. Entering the room is the action (a) performed on the object. The environment is represented by e. This policy is a subset of a policy that regulates the access to all the monitored room of the company. In particular, this policy exploits the following attributes: o.id represents the ID of the rooms, a.id represent the ID of the actions controlled by the policy, s.role represents the role of the subject in the company, o.numEmployee is a mutable attribute and represents the number of employees who are in the room in a given moment. The policy takes also into account an attribute of the environment, denoted by e.workingTime, which states whether the current time is within the working hours. The entire policy consists of two rules. *Rule-1* (lines 4-12) regulates the access of an employee to the data center room, and *Rule-2* (lines 13-24) regulates the access of guests in the same room. More specifically, line 3 defines the ID of the object protected by this policy, which is "CED"and it is the room that hosts the data center of the company.

The target of *Rule-1* is the action of entering a room (line 6). Instead, the pre-authorization expressed in lines 7-8 allows a subject with role "employee" to execute the action. If the pre-authorization is satisfied, the pre-update defined in lines 9-10 is executed and the attribute *numEmployee* of the room is incremented. This value will be decremented by the post-update (lines 11-12) when the subject leaves the room.

Even, the *Rule-2* targets the action of accessing a room (lines 14-15). The pre-authorization (lines 16-18) states that a subject is allowed only if he/she has attribute role "guest", and the value of the attribute *numEmployee* paired with the room is greater then zero. The pre-condition (lines 19-20), instead, states that the current time must be within the working time to permit the access. The on-authorization in lines 21-22, is continuously evaluated and states that the value of the attribute *numEmployee* of the room must be greater then zero also during the access time. Finally, the on-condition (lines 23-24) requires that the current time must be within the working time while the subject is located in the room.

6.5 Architecture of the Video Surveillance Framework

This section describes the architecture of the proposed Usage Control video surveillance framework, which is shown in Figure 6.2. The *Video Camera*s (VCs) are installed in the rooms to be monitored, and they capture the video streams for the identification and video surveillance processes. In our reference scenario, we assume that a set of cameras are installed in such a way that they provide a total coverage of the monitored rooms. A set of guidelines for designing effective video surveillance scenario set up, called "adversary and scenario engineering", is presented in [270]. The cameras are connected to the *Video Stream Elaborator* (VSE), which is the component in charge of detecting faces in the video streams it receives. The VSE performs the

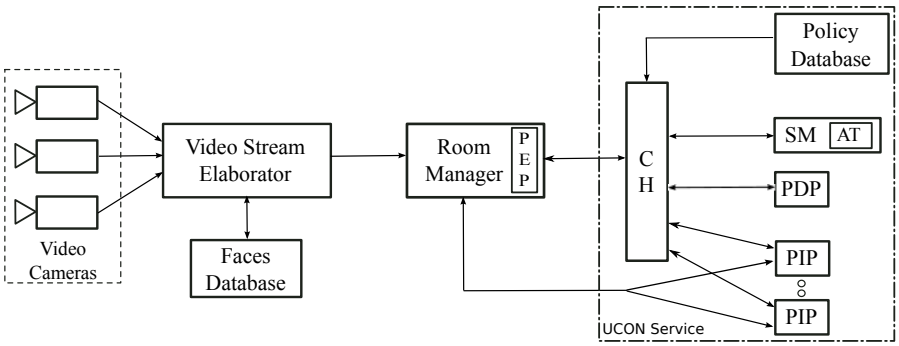


Figure 6.2: Architecture of the proposed surveillance framework

face detection exploiting the Haar[1] feature-based cascade classifier, which is an effective object detection method based on a machine learning approach where a cascade function is trained with positive and negative images that are useful to detect objects in other images. The VSE applies the classifier to each frame it receives from the cameras. If one or more faces are found, the VSE executes the face recognition step using the SIFT algorithm (§6.3.2), in order to match the detected faces with those ones stored into the Faces Database.

If one person (or more) has been detected in the room, the VSE sends the related ID, say A, to the *Room Manager* (RM), along with other attributes of A retrieved from the Faces Database, e.g., the role of A in the company and the department he belongs to. In order to know whether A has just entered the room or he was already there, the RM manages a table that stores the IDs of the people currently in the room. If A has just entered the room, the RM triggers the embedded Policy Enforcement Point (PEP) to interact with the Usage Control (UCON) service to perform the authorization process. If A was already in the room, instead, the RM simply sends the update command to the proper PIP in order to update the value of the attribute that stores the timestamp representing the last time A has been seen in the room. In addition, the RM is also in charge of detecting when a user leaves the room to notify the Usage Control service. The PEP interacts with the UCON service through the *tryaccess, permitaccess, denyaccess, revokeaccess, endaccess* messages, which are derived from the *Usage Control actions* defined in [262]. The PEP sends a *tryaccess* message to the UCON service every time a new person enters the room. This message includes the data collected by the RM (i,e., user ID, user role, room ID, etc). The UCON service evaluates the policy, and sends back a *permitaccess* or *denyaccess* response to the PEP. This enforces the *denyaccess* response by triggering an alarm and starting the video recording. In case of *permitaccess*, instead, the PEP does not perform any action because the person has the right to enter the room. When a person leaves the room the PEP notifies the Usage Control service by sending the *endaccess* message. In addition, the PEP starts the video recording when it receives the *revokeaccess* message from the Usage Control service, which states that the person

[1]OpenCV, http://tiny.cc/t2la4x

in the room is no more allowed to stay there. Finally, the PEP is also in charge of enforcing the obligations included in the messages received from the Usage Control service, such as playing some announcements.

The Usage Control service consists of several components, namely: the *Context Handler*, the *Policy Information Points*, the *Policy Decision Point*, the *Session Manager*, and the *Access Table*.

The **Context Handler** (CH) is the front-end of the UCON service, and it is invoked by the PEP embedded in the RM. It manages the components of the UCON service to execute the *pre-evaluation* and the *ongoing-evaluation* of the policy. In particular, when the CH receives the *tryaccess* message, it performs the *pre-evaluation* of the policy. To this aim, it retrieves further attributes that could be exploited in the decision process by querying the **Policy Information Point**s (PIPs). These attributes are related to the user who entered the room, to the room itself, and to the environment. The PIPs, in turn, interact with the real providers of such attributes, called Attribute Managers (AMs), according to the specific protocols they provide. The CH can be configured to interact with any number of PIPs, in order to be able to manage all the AMs available in each specific scenario. The CH contacts the Policy Decision Point (PDP) sending the access request message enriched with the values of the attributes just collected. The PDP evaluates this request against the pre-policy (i.e., the policy including pre-authorization, pre-conditions, pre-updates and pre-obligations only), returning the access decision to the CH. If the access decision includes pre-updates of some attributes, the CH executes them by exploiting the *update* interface of the PIPs which manage these attributes. Finally, the CH forwards the access decision to the PEP. If the PDP decision is *allow*, i.e., the person is allowed to enter the room, the CH begins the *ongoing-evaluation* of the policy. Hence, the CH subscribes the attributes required for the decision process exploiting the *subscribe* interface of the PIPs. Consequently, the PIPs provide the current values of the attributes they manage to the CH, and will notify the CH as soon as one (or more) of these attributes has changed its value. Then, the CH sends the corresponding enriched access request to the PDP to evaluate the ongoing-policy (i.e., the policy including on-authorization, on-conditions, on-updates and on-obligations only), which, in principle, is different from the pre-policy. If the response of the PDP is *denyaccess*, the CH sends the *revokeaccess* message to the PEP, which starts the video recording. Otherwise, no action is taken. Due to the attribute subscriptions, a PIP could notify the CH that the value of an attribute is changed. Even in this case, the CH coordinates the components of the Usage Control service to perform the ongoing-policy re-evaluation exploiting the updated values of the mutable attributes. If the evaluation of the ongoing-policy returns *denyaccess*, i.e., the person in the room loses the right to stay there, the CH sends the *revokeaccess* message to the PEP that starts the video recording.

The **Policy Decision Point** (PDP) is a XACML evaluation engine following the format defined by the XACML standard [271]. Hence, given a policy and an access request, the PDP evaluates the policy for that request, and returns the decision: *allow* or *deny*. As previously

described, the CH invokes the PDP each time a pre-policy or an ongoing-policy must be evaluated.

The **Session Manager** (SM) is in charge of keeping trace of the ongoing usage sessions in order to implement the ongoing-evaluation of Usage Control policies as a consequence of an attribute update. In fact, each time an attribute changes its value, the SM is invoked to determine for which of the sessions that are in progress the ongoing-policy must be re-evaluated. In particular, the SM exploits the Access Table (AT) to store data about ongoing sessions. The CH invokes the SM to create a new entry in the AT for each access request (*tryaccess)* that is allowed by the pre-evaluation phase, setting the session status in the entry to "active". The SM is then invoked to modify the session status from "active" to "revoked" when the *revokeaccess* message is sent to the PEP because the ongoing-evaluation of the policy results in a policy violation, i.e., the right of the person to be in the room is no longer valid. Moreover, the SM deletes the entries linked to ended accesses due to an *endaccess* message. Finally, the CH invokes the SM when the values of an attribute is changed. In this case, the SM retrieves from the Access Table the list of the sessions that could be affected by this change, and sends it to the CH, which performs the re-evaluation of the ongoing-policy exploiting the updated attribute values for each of them.

The **Access Table** (AT) keeps meta-data about accesses in progress, i.e., the active usage sessions. The Access Table is implemented through a Database. Each entry refers to an active session, and it stores a set of data such as the session ID, the access request, and the session status (i.e., pending, active, ended, revoked). As previously explained, these entries are created by the SM when new sessions begin, are read by the SM when the value of attributes change, and are deleted when the related accesses are terminated.

6.5.1 Architecture Development

This section describes some details related to the implementation of the surveillance framework depicted in Figure 6.2. In our framework, we used a D-Link DCS-942L IP camera, which is able to capture up to 20 frames per second with a resolution of 640x480 pixels. The camera is accessible using the HTTP protocol, and also it provides a built-in night vision with sound and motion detection capabilities. The Video Stream Elaborator is a software component developed as a Java application, and it uses JavaCV[2] to implement the functions for real-time computer vision, to access the camera video stream and to perform face detection operations. Each face detected by the VSE is compared with the faces available in the *Faces Database* to find a match (if it exists). The Faces Database contains a jpg picture of about 200x200 pixels per each user, along with user attributes (i.e., name, role, etc.). The Faces Database is implemented using MySQL as DataBase Management System[3], and the face recognition operation is executed using SIFT (§6.3.2.1) included in *Fiji*[4], which is a Java image processing package for image

[2]JavaCV, https://github.com/bytedeco/javacv
[3]MySQL, https://www.mysql.com
[4]Fiji, http://fiji.sc/Fiji

transformation, registration and interpretation.

We adopt SIFT to identify faces due to its robustness to variation in scale and orientation. In our test we observed that SIFT achieves a true positive rate near to 100% when the pictures in the database have good lighting conditions and quality of the captured stream is high without shades. That percentage decreases proportionally with the variation of those conditions. However, it is worth noting that the number of false positive remains negligible even in different conditions.

After the face identification, the VSE shares the *userID* and the *role* attributes with the RM, and the latter interacts with the the UCON service through the embedded PEP. Both the RM and the UCON service are entirely developed in Java. The PEP exploits the remote procedure call mechanism provided by the Apache XML-RPC library (v.3) to interact with the UCON service, in particular with the CH. Communications between PEP and CH use the XML-RPC protocol over HTTP since in our testbed the RM and the UCON service are deployed on the same machine. However, HTTPS is also supported by our prototype, and it should be adopted when the RM and the UCON service are deployed on distinct machines.

The PDP implementation is based on the WSO2 Balana[5] open source software. Balana implements the XACML engine, while the CH invokes the PDP by means of its API that follows the XACML standard.

The SM is the component that manages the database to store the data related to the ongoing sessions. In particular, the SM implements the Access Table exploiting a Data Abstraction Layer based on the Oracle Java Persistence Architecture (JPA) v2.1 [272], which is an Object Relational Mapping (ORM) abstraction. This allows the SM to work with distinct DataBase Management System by properly changing a configuration file. In our reference implementation, the SM exploits the Apache Derby[6] DataBase Management System.

We implemented two PIPs in our prototype. The first PIP manages the number of employees present in each monitored room. The second PIP says whether the current time is within the working hours or not. Their interfaces provide remote methods to perform the *retrieve, subscribe/unsubscribe and update* operations. These PIP communicates with the CH through XML-RPC over HTTP calls, since they are located on the same machine of the CH.

6.5.2 Architecture Deployment and Evaluation

To validate our video surveillance framework, we protected a room of our department with the prototype we developed. When a person enters the room, the continuous face recognition is performed through the cameras, and the system governed by our Usage Control service enforces the policies shown in Table 6.1. Policies are enforced exploiting the following attributes:

1. the role of a person;

[5]Balana, http://xacmlinfo.org/category/balana/
[6]Apache Derby, http://db.apache.org/derby/

2. the number of employees inside the room;

3. if the current time is within the working time.

The role attribute is retrieved from the Face Database during the face recognition phase, while the remaining two attributes are retrieved using the following PIPs:

- *PIPNumEmployee*: this PIP is in charge of counting how many employees are currently within the room. The value is kept in a attribute that is incremented and decremented by the policy.

- *PIPWorkingTime*: this PIP is in charge of keeping a boolean value indicating whether the current time is within the working time or not.

To indicate the presence of users outside/inside the room, we developed a dashboard application, which is linked to the Usage Control module and video cameras. This application shows the position of the users inside or outside the room. When a user accesses the monitored room the camera detects his/her face and SIFT executes the face identification, by matching the face detected with those ones already available in the faces database. The configuration of SIFT that we set up in our testbed framework allows the Video Stream Elaborator to recognize faces with a number of false positive close to zero. This result is achieved by setting a high accuracy threshold in SIFT when it matches faces. Thus, when our Video Stream Elaborator component identifies a person, it gives a very high certainty that the user identified is that one registered in the faces database. However, this high level of accuracy requires that the quality of the video stream captured by the camera is very good and clean. So, blurry images are discarded by the Video Stream Elaborator during the face recognition phase.

The dashboard application that we developed is thought to be used by a security operator to monitor the activity in one or more rooms. The dashboard application represents each user with an icon that is automatically moved inside or outside the room by the Room Manager according to the position of the user. The state of a user is represented by our application using different colors for the icon:

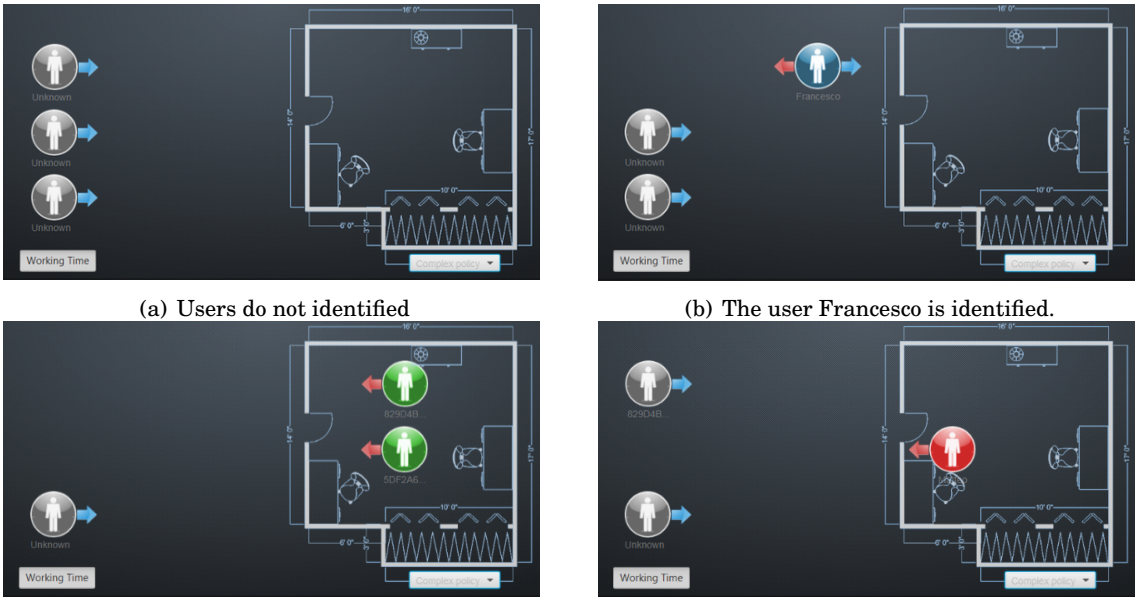
- grey: the user is outside the camera view;

- blue: the user face is being processed;

- green: the user is authorized to stay in the room;

- red: the user is not authorized to stay in the room.

If the Video Stream Elaborator identifies more than a single face in the stream, it is able to correctly identify the faces as done in a single face identification setting, and even in this



(a) Users do not identified

(b) The user Francesco is identified.

(c) A guest with an employee inside the room.

(d) The authorization for the guest is revoked.

Figure 6.3: Identification and authorization states recorded by our Video-Surveillance application.

situation, false positive percentage is negligible, provided that the video stream should present high quality of images without shades.

In the empirical tests we conducted in our laboratory, we involved two actors, i.e., an employee and a guest. Fig. 6.3a shows all users outside the room with no identification, so they are labelled as *unknown*. When a user gets in the room, the Video Stream Elaborator correctly identifies the user *Francesco*, see Fig. 6.3b. In parallel, the Room Manager sends a *tryAccess* with the related *userID* and *role* to the UCON; the UCON successfully returns a *Permit* allowing the user to stay in the room, see Fig. 6.3c. In this phase, the Room Manager obscures the identity of the employee with a randomly generated identifier and the room recording state is moved to OFF to protect employee's privacy.

Always in Fig. 6.3c, the Video Stream Elaborator highlights the presence of another user. He is identified as a guest and he is granted by the Room Manager to stay in the room only because an employee is already inside the room during the working time range, as policy number two states. When the employee leaves the room (Fig. 6.3d), the policy is not matched anymore and the guest is obliged to leave the room. To inform the guest of this condition, we use a speaker inside the room that notifies the guest about the new authorization state, and the guest must leave the room. For security reasons, the cameras inside the room start recording each action done by the guest.

6.6 Privacy-preserving face recognition

The use of video-surveillance into departments, offices, banks and so on, is seen as a very sensible aspect since employers cannot leverage on video-surveillance in any situations. Often, employers use cameras for security purposes but at the same time cameras are forbidden if used to monitor employees' activity[7]. In general, the rapid improvement and widespread deployment of video-surveillance technology raises strong concerns regarding the violation of individuals' privacy. In fact, biometric information can be collected and misused to profile and track individuals against their will. For these reasons, the interest in privacy-preserving face recognition systems is increasing and to tackle this aspect, we propose a privacy-preserving solution that aims to authenticate users/employees keeping private their identity during the recognition phase. Thus, we designed our privacy-preserving solution to work in compliance with the architecture defined in Figure 6.2. However, we force the Video Stream Elaborator to not match in clear the face recognized with the Faces Database, instead it runs a Secure Two-party Computation (2PC) session with a new component placed in the next to the Face Database. In this way, we build a client-server infrastructure in which the client is represented by the new sub-component installed in the VSE, while the server resides in a new component that is placed just before the Face Database and physically available far away from the VSE or outsourced to a third-party, see Fig. 6.4. We repeat saying that the goal of this two components is to run a 2PC in which an employee is authenticated without knowing his/her real identity. So, the VSE will know, after the 2PC session, the content of a boolean value saying whether the employee was identified or not. If the VSE receives a *true* value, then the Room Manager that will proceed (or deny) the authorization.

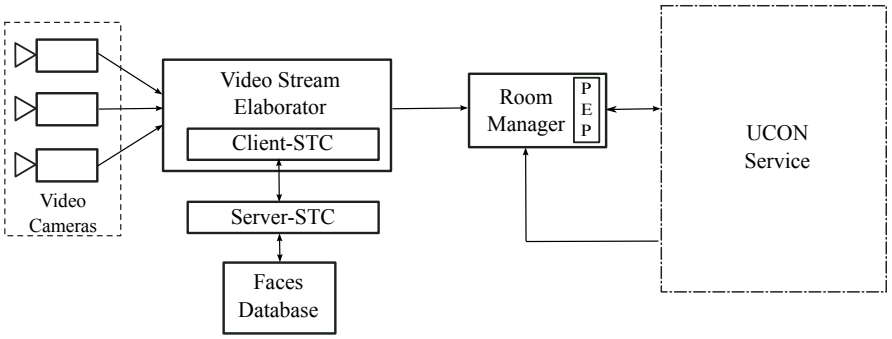


Figure 6.4: Architecture extended with 2PC components

[7] http://files.findlaw.com/pdf/employment/employment.findlaw.com_workplace-privacy_can-employers-use-video-cameras-to-monitor-workers.pdf

6.6.1 Secure Two-Party Computation

Secure Two-party Computation is a sub-field of cryptography that allows two users to jointly compute a function using their own inputs, and keeping those inputs private. The classical secure two-party computation example is the Millionaires' problem expressed in [273]. Here, two millionaires are interested in knowing which of them is richer without revealing their actual wealth. In order to accomplish this kind of tasks a number of protocols have been proposed in the past years. The first one is the Yao's protocol. The attacker model considered in this scenario is the "semi-honest". In this context, the two parties run the protocol exactly as specified (no deviations, malicious or otherwise), but they may try to learn as much as possible about the input of the other party from the protocol flow.

6.6.1.1 Garbled circuits

Garbled circuits are encrypted boolean circuits that play a central role in secure two-party computation. In these circuits the boolean values are replaced by encrypted signals, and the gates provide enough information to obliviously evaluate the circuits gate-by-gate to produce a garbled output. In a garbled circuit, each wire is associated with a pair of random binary strings called wire signals, corresponding to the 0 and 1 bit values in the boolean circuit. These values are known as the semantics of the signals since this reflects the meaning of the signal in the "cleartext" circuit. The correspondence between the wire signals and their semantics is random and is kept secret. Each two-input gate in the garbled circuit consists of four gate labels presented at random order that corresponds to the four possible input values. Such labels are stored in garbled computation tables. A garbled circuit is made by the garbled computation table of all its gates, the description of their interconnections, and the output description tables that map the random values on circuit-output wires to their corresponding real values. In order to evaluate a gate for a given pair of input wire signals it is necessary to locate the gate label corresponding to such signals, then some computations involving the gate label and the input signals are performed and the output wire signal is obtained. In more detail, the output wire signal is retrieved by applying two consecutive decryption processes on the correct gate label using the related wire signals. The garbled computation tables are designed in such a way that it is possible to get the gate label corresponding to a certain pair of wire signals without revealing anything about the value it is bounded to.

6.6.1.2 Oblivious Transfer

In cryptography, an oblivious transfer protocol (often abbreviated OT) is a type of protocol in which a sender transfers one of many pieces of information to a receiver, but remains oblivious as to what piece (if any) has been transferred. A useful form of oblivious transfer that finds application in the field of Secure Two-party Computation is the called 1-out-of-2 oblivious transfer.

Such a protocol is used in a scenario involving a sender, who has two messages, and a receiver, who can chooses which one of the messages he wishes to receive. By running this protocol the sender can send to the receiver the message selected by this one without knowing which of the messages is, and without letting know anything to the receiver about the other message.

6.6.1.3 Yao's protocol

The sender and the receiver have respective inputs x and y, and wish to compute the output f(x,y).

1. the sender constructs a garbled circuit and sends it to the receiver;
2. the sender and the receiver interact so that the receiver gets the input-wire keys that are associated with the inputs x and y. In particular, the sender simply sends to the receiver the keys that are associated to its own input and runs with him a 1-out-of-2 oblivious transfer protocol. During the OT protocol the receiver learns the garbled values of the input wires corresponding to its input;
3. finally, the receiver computes the garbled circuit as described above and concludes the protocol.

6.6.1.4 CBMC-GC

CBMC-GC [274] is a compiler for C programs in the context of Secure Two-party Computation (2PC). It compiles a C program that specifies a secure computation into circuits that can be read in by an 2PC platform, which then performs the secure computation between two parties A and B. At the beginning of the compilation, CBMC-GC translates the C program into an intermediate representation featuring a simplified set of control statements. This program is then translated into circuits by symbolically executing the program. To perform the symbolic execution CBMC-GC unrolls loops and recursive function calls up to a predetermined bound. This bound is either determined automatically via an internal static analysis of the program or is given by the user in case CBMC-GC is not able to discover the bound. During the symbolic execution each statement (i.e., "if" or "for" blocks) is translated into a subcircuit, which represents the semantics of the respective statement. In such a way, the input wires to these subcircuits represent the current program state, and the output wires represent the state of the next program after the application of the statement in the current program state.

6.6.1.5 Attacker Model

The attacker model that is considered in the Secure-Two party Computation framework CBMC-GC is *honest-but-curious*. In this model, the attacker follows all protocol steps deigned in the 2PC session, but he/she can try to learn additional information about the other party during message exchanging phase, with the purpose to acquire some details of the identified person.

Moreover, it is important to notify that when executing a 2PC session, there is an asymmetry on the provided security guarantees. In particular, it is not possible to prevent one party from quitting the protocol prematurely, and not sending the result of the computation to the other party. We know that this situation is a weakness but we cannot overtake it due to 2PC specifications flow.

6.6.2 The implementation

Our privacy-preserving face recognition solution, which integrates 2PC, leverages on the CBMC-GC framework and it is tested using both SIFT and Eigenfaces algorithm for face recognition. In particular, our SIFT implementation is developed using "mpicbg[8]", which is a JAVA open source collection of algorithms focusing on image transformation, registration and interpretation. Instead, for Eigenfaces, which involves a lot of computations on matrices, we used "colt[9]" to get a set of JAVA open source libraries for high scientific performance.

During the implementation of our face recognition solution with support of 2PC, we faced a number of issues related to the complexity of the integration of the Secure Two-party Computation technique in conjunction with the face recognition algorithms. Our first attempt of integration was the porting of both face recognition algorithms in C to compile them with CBMC-GC and generate the garbled circuits needed to run a 2PC session. However, the CBMC-GC compiler was not able to compile both algorithms due to their structure complexity. In fact, to compare two images using SIFT, it requires a feature-by-feature match and SIFT extracts about 500 features per image and each feature consists of an array of 128 values: so each matching operation involves about 250.000 comparisons between features, and each comparison consists in the calculation of the difference vector between 128-dimensional vectors. Slightly better is the Eigenfaces case, projecting a facial image of "P" pixels into the N-dimensional eigenfaces space, involves "N" scalar products between P-dimensional vectors (in our case, $P = 300x250 = 75.000$). Thus, in both cases it was not possible to generate the necessary circuits to run a secure computation session. To overcome these issues, we decided to use a simplified version of the two algorithms, and to reduce the portion of the algorithms run in the privacy-preserving mode. For SIFT, we reduced the number of features describing an image and the length of the feature descriptors, but this led to an unacceptable reduction of the algorithm performances in terms of accuracy. Due to the limitations occurred with the integration with SIFT, we decided to use Eigenfaces but leaving the part related to the face projection in the eigenfaces space not involved in the secure computation. So, we designed and implemented the following face recognition privacy-preserving flows:

1. *Pre*-2PC session:

[8] `http://fly.mpi-cbg.de/~saalfeld/Projects/`
[9] `https://dst.lbl.gov/ACSSoftware/colt/`

a) the Server shares with the Client the average face and the eigenfaces computed over the Face Database. The Client will use this to project its image into the eigenfaces space;

b) the Server projects all images contained in the Face Database into the eigenfaces space;

c) the Client projects its image, which it wants to recognizes, into the eigenfaces space;

2. *During*-2PC session:

a) the Server and the Client run a Secure Two-party Computation session.

b) for each image in the Face DataBase, the Server gives as input the projection of that image, the ID of the image, the corresponding Role and a threshold[10];

c) the Client gives as input the projection of the face it wants to authenticate and a random integer number;

d) When the 2PC session ends, two cases are possible:

i. *Match Found*: the 2PC session produces three integer numbers in the form:

$$\{0,1\};\{0,1\};123456 \tag{6.1}$$

The number “1" in the first position indicates that a matching is found, “0" otherwise. The integer in the second position indicates: “1" that he/she is an employee, “0" that he/she is a guest. This integer is got during the execution of the 2PC session and provided by the Server from the Face Database. The third integer is a fake ID number that is obtained by summing the real ID of the picture saved in the Face Database and the number sent by the Client. Even if the Server gets this sum, it does not know the number sent by the client since it is not unveiled in the 2PC. So, the sum result is meaningless for the Server. Instead, the Client gets the real ID by subtracting to the fake ID the random ID number that only it knows.

At this point, the Client stores the real ID number in a local table and associates a new random[11] ID to be communicated to the Room Manager. In this way, each time that the Client receives a fake ID from the 2PC session, it first calculates the real ID, checks in its table the associate random ID, and then communicates it to the Room Manger.

[10]This threshold establishes the accuracy of the identification by calculating the Euclidean distance between the two projections.

[11]This new ID is generated only once by the Client each time that it receives an ID not already stored in its local table. Then, the random ID is sent to the Room Manager that will work with this ID without knowing the real ID associated to the identified employee/guest.

ii. *Match NOT Found*: the 2PC session produces one simple integer in the form:

$$\{0, 1\} \tag{6.2}$$

Where, the number “0" confirms that no match is found.

6.6.2.1 Results

To test our solution, we used a Face Database of 20 images 125x150px, each represented as a linear combination of 11 eigenfaces. Since CMBC-GC can only produce circuits operating on integer values, we multiplied per 100 the vectors representing the faces in the database, and we considered only the integer part of the results for the inputs during the 2PC session without leading to any observable decrease of performances.

Once the VSE detects the face in the video stream, it extrapolates the position of the eyes[12] and use it to transform the face in a format compliant with the face stored in the Face Database. Then the 2PC session is run, and the boolean result is got in about 25 seconds in a privacy-preserving fashion compared to the 6sec. needed to run Eigenfaces without the privacy-preserving feature.

Concluding, it is worth noting that our tests highlights the sensibility issue of Eigenfaces about the variation in lighting and orientation reported in literature as come out from the Holistic approaches.

6.6.2.2 Considerations

From the results obtained in our tests, we observed that the time needed to run a face recognition session with the privacy-preserving technique is not negligible. However, the effort employed in obtained this such a feature is quite relevant since the use of a privacy-preserving approach can limit the criticality that face recognition has in some context, like offices and so on.

To get a proper integration with our Usage Control system, we put a lot of effort to avoid that the server achieves any information about the identified person. Although this property is kept by the 2PC technique, our Usage Control system needs some details to continuously monitor people in the room. In particular, the Room Manger must know from the VSE the “ID" of the employee and his/her role. But, if the Server knows the ID at the end of the 2PC session it may understand which person was identified. To overtake this issue, we use the integer number sent by the Client at the step 2.C that is summed to the real ID of the employee. At the end, only the Client, which knows the random number, is able to get the real ID of the employee, instead the Server receives a meaningless number.

[12] Specifically, the more aligned the eyes are, the more accurate the eigenfaces algorithm is.

6.7 Conclusion

In this chapter, we proposed the adoption of a Usage Control based authorization system to establish whether one o more people are authorized to access and stay in a monitored room. We validated our approach by developing a prototype of the proposed architecture and testing the enforcement of the policy described in Table 6.1. Components of our prototype where tested to verify that people who accessed the room were correctly authenticated and their behavior corresponded to what the policy declared. On the contrary, if the policy is not respected, we verified that the Usage Control system was able to detect such violations.

We presented a flexible architecture composed by two main parts: the first involves the authentication step, while the second the Usage Control. The choice to provide an architecture based on separate modules allows us to modify one part without altering the other. With this modular architecture we designed and deployed to privacy preserving solutions: in the first solution, we record the video stream only in case of policy violation to preserve the privacy of the monitored people. Instead, in the second solution, we added two additional components to make our face recognition phase in privacy preserving manner using the Secure-Two party Computation techniques.

Finally, we presented the results of our implementations showing that the 2PC technique helps to keep private employees' identification, although it adds a not negligible computational time.

CHAPTER 7

A Privacy-preserving Location-aware information service for mobile users

7.1 Introduction

Today's mobile technologies have dramatically increased the opportunities of keeping the mass informed. This is not limited to the news but expands to all types of information. In particular, it concerns information that raises people's interest depending on their location. For example, a satellite navigation system running on a user's mobile device will download the maps around the user's location. Similarly, it seems more appropriate to deliver dating information to a user about the user's surroundings than about another continent. Also the appropriateness of emergency information inherently depends on the location of the users.

The simplest protocol to regulate information delivery perhaps is the one that ships with Android. It sees the information provider, namely Google, receive the user's precise location upon the user's consent, and then deliver information that is appropriate to that location. Although this is very practical, it raises privacy concerns on the user's location, which qualifies as private for various reasons. One is that it is a clear market asset: the provider or third parties could deliver location-based shopping suggestions.

This chapter takes the challenge of studying ways to balance the appropriateness of information for a location with the location privacy of the receiver of that information, which are apparently contradicting requirements. One protocol only delivers the user's approximate location to the provider, and hence is termed AL protocol. With respect to Android's protocol, it is found to increase location privacy at the cost of decreasing information appropriateness. Another protocol resorts to a cryptographic technique to maximise both location privacy and information appropriateness. The cryptographic technique is secure two-party computation (2PC), which

enables two principals to compute a shared function out of their respective private inputs without disclosing the inputs to each other. Notably, this removes the need for a trusted third party to get the inputs privately and then compute the function.

The protocol that adopts 2PC ensures location-based and yet privacy-preserving delivery of information, and hence is termed LBPP protocol. All protocols are informally analysed in terms of security and privacy. In consequence, the LBPP protocol is implemented and exposed as the Getmewhere service for the reader to try. Obviously, it can be tailored to notifying all sorts of location-based information.

This chapter is organized as follows. (§7.2) reviews some related work before facing information delivery and the three protocols to realise it (§7.3). Then, it provides a security analysis (§7.4) and a privacy analysis (§7.5) of the protocols. (§7.6) concludes this chapter.

7.2 Related Work

Secure Two-Party Computation. 2PC concepts have been thoroughly discussed in section 6.6.1. In the following major nowadays 2PC implementations are presented.

FairPlay [275] is a well-know system that allows a user to define functions using the high-level language SFDL. The function is processed by a compiler that outputs garbled boolean circuits. FairPlay is shown secure in terms of privacy and integrity: principals cannot learn more details about each other's inputs than they could if the function were computed by a secure trusted third party; at the same time, an attacker cannot alter the output of the computed function. Newer versions of this system are FairplayMP [276], which is a multi-party extension, and MobileFairplay [277], which is the two-party version ported over Android.

A recent 2PC toolkit that is worth mentioning is MightBeEvil [278]. It allows a user to easily write functions that can be run privately to the user. Out of our empirical tests, MightBeEvil appears to be faster and less memory consuming than Fairplay.

The CBMC-GC tool, already presented in section 6.6.1.4, sports the desirable feature that the input functions can be conveniently written in the popular ANSI C programming language. This details brings at least two main advantages over competitors. One is that the size of the garbled circuits resulting from functions written in C generally is very small, hence the circuit evaluation is correspondingly fast. The other one is that one can check the correctness of the input function by appealing to the variety of software verification tools that are available for C. After experimenting with all technologies outlined above, these features of GBMC-GC determined our choice to adopt it as an 2PC tool for practical use in our implementations.

Location Privacy. Mascetti et al. use the location shifting technique and a centralized service to establish whether two users are in proximity [279]. Their protocol transforms the real location of a user into a new location using an encryption function that first encodes the real location and then applies the *modulus* operation. The distance between the two users is

calculated using the Euclidean formula, and proximity is easily defined by appealing a threshold.

Puttaswamy et al. use the same location shifting technique [280] as [279]. However, they first partition all users into groups of friends interested in each other's location and data. All members of a group know the transformation keys of everyone in the same group, and these are shared with a server. Every group member translates their real coordinates into encrypted ones and uploads them on the server, which computes the various distances.

Sedenka and Gasti advance novel distance computation protocols and perform proximity testing based on homomorphic comparison and garbled circuits [281]. The implementation of these protocols is at a prototype level.

7.3 Information Delivery

We term *information delivery service* or, for simplicity, *information service*, a service that delivers live information, such as for navigation, for dating and for handling emergencies. We term *event* each piece of live information that an information service provider delivers to its subscribing users. At present, information services are often tailored upon the user's location. The appropriateness of the event for the user must be assumed to depend on that location.

There are other assumptions about the service provider. Several operators work permanently (taking shifts) in a control room to manage events. When an event arrives through a news agency or dedicated sources, it gets assigned to a team of operators who evaluate it. Following the evaluation, the team decides the size of the geographical area, centred on the location of the event, within which the event is relevant.

The general scenario of information delivery also sees the subscribing user endowed with a mobile device that runs a dedicated app through which the provider delivers information. The app exploits its user's location.

Information delivery requires dedicated protocols, and three of these are presented in the following. One is the well-known Android's protocol, while the other two are the authors' novel designs aimed protecting the user's location privacy. Below, *SP* denotes the service provider, and *U* denotes the mobile device and at the same time its user who interacts with the service provider through the dedicated app.

7.3.1 Android's Information Delivery Protocol

The information delivery protocol that ships with Android is very simple. If the user discloses their precise location to Google, the user will then obtain information that is specific to that location. It consists of two phases:

Setup phase. The user chooses whether to disclose their location by setting the *consent* predicate. In practice, this is done through the system settings. Regardless of the predicate value, he sends it to the provider. Only if the predicate value is true, will he send also his location to the provider.

Execution phase. This phase begins when the provider has an event e to notify. Upon the basis of out-of-band information, the provider decides the area $area(e,m)$ within which e is relevant. The provider then verifies if the user's location $gps(U)$, which was received during the previous phase, lies within $area(e,m)$ and sets the *membership* predicate consequently. Only if the predicate value is true, will the provider notify the event to the user through the app.

7.3.2 The Approximate-Location (AL) Information Delivery Protocol

We design a new information delivery protocol based on an Approximate Location (AL) of the user. In the AL protocol, users select an area of interest on their device, namely an area they want to receive information about from the provider. The AL protocol is given in Figure 7.1, and consists of two phases too:

Setup phase. The user chooses some size parameter n and calculates their area of interest

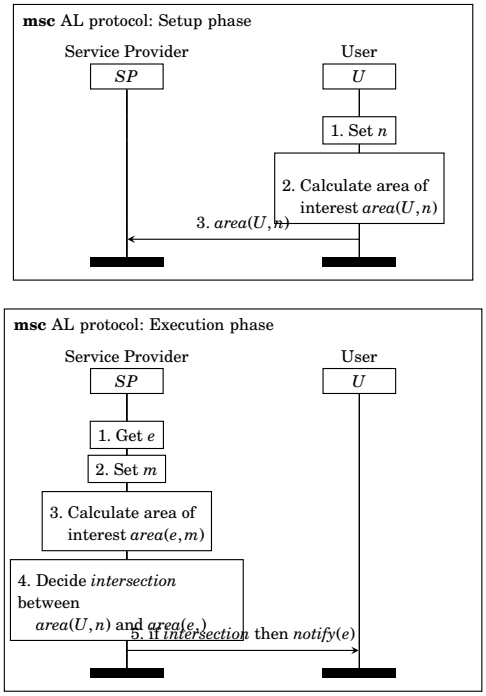


Figure 7.1: The AL Information Delivery protocol

based upon that parameter. Without loss of generality, the area of interest can be defined as a region of some shape based upon the size parameter n and containing the user location. Finally, the user sends that area to the provider. In practice, n can be thought of as a parameter stored in the settings of the app. The user can return to redefine it at a later time, causing the entire phase to run again. While the user moves inside the area, the protocol prescribes this area not to be changed; as soon as the user moves out of the set area of interest, the entire Setup phase must be run again so that the user may define a new area of interest with respect to their new location.

Execution phase. As with the previous protocol, this phase begins when the provider intends to notify an event e and decides that it is relevant over area $area(e,m)$. Using the user's area of interest $area(U,n)$ received during Setup, the provider checks whether the intersection of the two areas is non-empty, setting the *intersection* predicate correspondingly. Only if the predicate value is true, will the provider the event to the user through the app.

7.3.3 The Location-Based Privacy Preserving (LBPP) Information Delivery Protocol

We design another information delivery protocol that appeals to a cryptographic primitive to meet the requirement that information is delivered to a user only if it is certainly relevant to the user location; yet, at the same time, the protocol wants to meet the requirement of keeping the user location private. These logically contradicting requirements can be reconciled by accessing the user's location through an appeal to 2PC. The protocol therefore is location based and also privacy preserving (LBPP). It consists of the usual two phases (Figure 7.2)

Setup Phase. While the provider plays no role, the user customarily calculates their area of

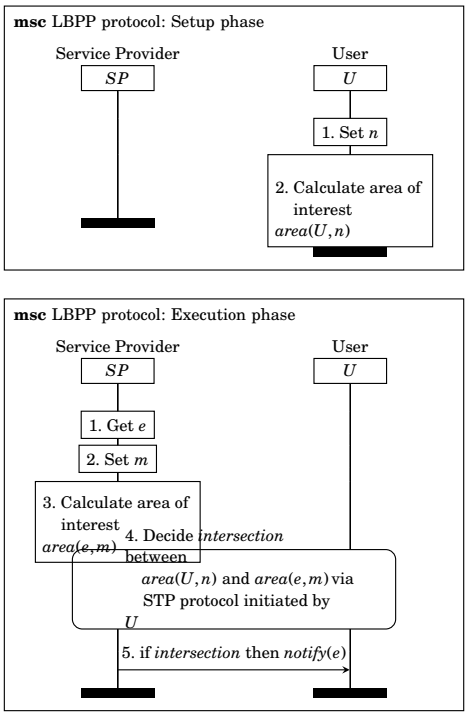


Figure 7.2: The LBPP Information Delivery protocol

interest. Therefore, this phase is the same as that of the AL protocol except for the fact that the user does not transmit her area of interest to the provider. He will do this in the next phase.

Execution phase. This phase begins when the provider gets an event and decides that it is worth being notified. As with the AL protocol, this event originates out of band and, having

evaluated it, the provider calculates the area that it deems relevant for it. At this point in time, the user begins an 2PC protocol with the provider to establish whether there is intersection between $area(U,n)$ and $area(e,m)$, then he sets the *intersection* predicate correspondingly. Only if the predicate value is true, will the provider notify the event to the user through the app. The 2PC protocol is taken off-the-shelf, hence its details are omitted from this presentation. It is used as a black box and its implementation is available, for the user, as Android app termed "Getmewhere" [282] and as service, for the provider, available here [283].

7.4 Security Analysis

The particular protocol notwithstanding, whether confidentiality is a requirement in our scenario depends on the specific type of information that the service provides. For example, while emergency events may be assumed to be public, dating events may be required to be confidential. Taking a security-by-design approach, it may be wise to generally state this requirement. Similar considerations apply to the authentication requirement.

These requirements could be met by establishing a secure channel between the provider and the app using standard technology such as SSL/TLS and by running the particular information delivery protocol of choice over it. The secure channel should be combined with an appropriate pushing system to enable the app to receive the live information as it comes. Various technologies implementing that combination are available today, such as Apple Push Notification Service [284], Microsoft Push Notification Service [285] and Google Cloud Messaging [286].

Due to the known asymmetry of 2PC, which favours the initiator, a specific security consideration pertains to the LBPP protocol only. Because the protocol design wants the user as initiator of the 2PC protocol, there is no way to prevent the user from terminating the protocol prematurely without sending the outcome of the computation to the other principal. A malicious user could then refuse to terminate the protocol with the provider after learning that their location is within the area of interest of an event that the provider could notify; however, this does not seem beneficial to the user, making this potential attack in fact impractical.

It is clear that, if the protocol had chosen the other principal as initiator, then a malicious provider would have had an advantage: the opportunity to arbitrarily decide whether to notify an event to a specific user despite their being in the area of interest set for the event.

7.5 Privacy analysis

Android's protocol makes the evident choice of privileging appropriateness of the information it delivers at the cost of user privacy (the user's location is sent to the provider).

As for the AL protocol, it is obvious that the bigger the area of interest, the less pertinent the information that the user receives ends up being. However, this approximation is paid back to the user in terms of their location privacy; the user in fact only discloses an area in which they may

be located, rather than their precise location. At an extreme, the user might (attempt hacking the app that implements the AL protocol in order to) select an area of interest and lie outside that area, but it is clear that they would receive information irrelevant to their actual location.

Another privacy consideration is worth making about the AL protocol. It prescribes the user's area of interest (of size parameter n) to randomly contain, not to be centred on, the user location. This choice enhances location privacy, otherwise the provider could calculate the user location with certainty from the user's area of interest. It seems wise that an app that implements this protocol should prevent users from centring their area of interest on their location.

Thanks to the use of 2PC, the LBPP protocol is the most clever: it can deliver information that is appropriate to the user's precise location while keeping that location private for the user. It seems fair to claim that the Android's protocol enjoys maximum information appropriateness and no location privacy, the AL protocol grants some information appropriateness and some location privacy and the LBPP protocol optimises both variables because it delivers information whose area of interest intersects the user's, and does so without disclosing the latter.

Let us consider location privacy alone. While Android's protocol is trivial to assess, it is more interesting to relate the other two. The AL protocol reveals the area in which the user is located, so the provider's probability to precisely locate the user also depends on the size of that area. One may observe that also the LBPP protocol enables the provider to decide whether an event is relevant to a user, and therefore the provider does learn something about the user's location. But it only learns that the user's area of interest, which normally contains the user's location, has some intersection, no matter how small, with the area of interest for the event. Therefore, the precise location is concealed both because the size of the user's area of interest could vary significantly as it is chosen by the user, and because it does not necessarily lie in the intersection. This appears to be a better privacy preserving result than that of the AL protocol.

7.6 Conclusions

This paper investigated solutions to mitigate a specific form of privacy loss: the location privacy loss that a subscriber to an information service risks to receive information that is appropriate to the subscriber's location. It designed the AL protocol, one that generalises Android's use of the user's precise location to an area that could include the user's location. It then advanced the use of 2PC to protect the user's precise location and yet deliver appropriate information through another protocol, the LBPP protocol.

The privacy analysis and the adoption of 2PC determined our choice of the LBPP protocol for implementation as a service, called Getmewhere. We believe that this work contributes to the important meta-goal of raising people's awareness on the value of their privacy, which perhaps is itself a central goal for the next decade.

CHAPTER 8

ENABLING STANDARD CERTIFICATE-BASED MUTUAL AUTHENTICATION IN CONSTRAINED IOT SCENARIOS

8.1 Introduction

The increasing number of smart objects surrounding people in their everyday lives is a clear sign that the envisioned IoT scenario is quickly becoming a reality. Such an ubiquitous emergence of connected and autonomous sensors and actuators is turning people's immediate environment into a cyber-physical system with whom they can continuously interact, by requesting and/or providing services, through personal mobile devices like smartphones, connected cars and wearable devices. It is therefore clear that the IoT paradigm is going to dramatically change the way people live their lives, seamlessly augmenting their activities by means of digital applications [4].

Classic IoT applications, however, are statically configured to rely on an a priori known set of smart objects implementing the functionalities they need to deliver their services. This lack of flexibility inevitably results in an inefficient use of resources and in closed applications which cannot integrate with the surrounding ever-evolving ecosystem to provide new services and dynamically meet the users' needs.

For this reason, and in order to fully exploit the potential of the IoT paradigm, nowadays IoT systems are striving to break down the boundaries between them, evolving from a static scenario characterized by a plethora of closed, vertically integrated applications, towards an open scenario of horizontally integrated applications which can take advantage of the services provided by any not predefined IoT infrastructure [287].

In this new scenario, for example, wearable devices for runners will be able to autonomously reach agreements and exchange information with wearables of passing by users and with

surrounding smart objects and sensors offering different services . Depending on the data they manage to collect leveraging the reached agreements, wearables could then provide new services to their owners, and lead them on demand through the less polluted, or less noisy, or less crowded paths. Likewise, the smart cameras of the systems presented in chapters 4 and 6 could offer their services to wide and dynamic set of applications and users, and softwarized IoT nodes such as the ones presented in chapter 5 could even offer to execute custom programs provided by third parties.

Security technologies dealing with authentication, confidentiality and integrity issues will play a key role in this new, dynamic and open context where no a priori knowledge of the involved parties can be guaranteed [288]. Indeed, the great relevance of such topics is reflected in the fact that the architecture proposed by oneM2M, the global standard initiative for the development of a comprehensive machine-to-machine and IoT communications and management system, also encompasses security aspects [104]. With regards to mutual authentication, the oneM2M architecture supports three Security Association Establishment Frameworks [289]:

- **Provisioned Symmetric Key**: using symmetric keys shared by devices for authentication and session key establishment;
- **Certificate-based**: using certificates provisioned to devices for authentication and session key establishment;
- **M2M Authentication Function (MAF)-based**: using a MAF hosted by a third party which authenticates the involved end-points by means of pre-shared keys and/or certificates and facilitates the establishment of a symmetric key.

The Provisioned Symmetric Key framework is clearly inadequate for the considered open and dynamic scenario since it requires both the parties involved in the communication to pre-share some cryptographic material.

The Certificate-based framework overcomes this issue by leveraging on public key certificates. However this approach also falls short in our context because of the inherent limitedness of smart objects in terms of computational and energetic resources, which often clashes with the heaviness of the operations they should be able to perform to take advantage of the authentication mechanisms provided by a Public Key Infrastructure (PKI). The use of Raw Public Key (RPK) certificates [290], introduced to overcome this issue, presents however the same limitations of the previous framework.

The MAF-based framework, in the end, gets around this problem by introducing a trusted third party which shares symmetric keys with constrained smart objects and helps them in the authentication process with other entities. This framework also has limitations since it is applicable only when both parties involved in the communication rely on the same MAF. If they do not, and none is able to mutually authenticate with the other's MAF by using certificates, one of

them must register to the other's MAF by carrying out a MAF client registration procedure based on the oneM2M Pre-Provisioned Symmetric Key Remote Security Provisioning Framework. This implies a laborious process involving another third party hosting a M2M Enrolment Function (MEF) with which the constrained node pre-shares a symmetric key and which helps it in establishing a symmetric key with the other's MAF. This process must be repeated for every new MAF, and then all MAFs cryptographic material must be stored into the node. Moreover, additional messages must be exchanged by the nodes and the MAF to disseminate the necessary information to identify and reach the target MAF, and to generate and retrieve the correct symmetric key. All these storage and transmission overheads may constitute a problem for constrained nodes.

In this chapter we propose the PKIoT architecture, a solution aimed at making PKI and certificate based authentication available in the forthcoming IoT scenarios and at overcoming the aforementioned shortcomings of the oneM2M Security Association Establishment Frameworks.

The rest of the chapter is organized as follows: section 8.2 presents the PKIoT architecture in all its components, its security characteristics and its relationships with the oneM2M architecture; section 8.3 describes the simulation settings and implementation; section 8.4 reports and discusses the experimental results; section 8.5 closes the chapter. A thorough discussion about related works can be found in section 2.5.2.

8.2 The PKIoT Architecture

In this section we present the proposed PKIoT architecture, a *Public Key Infrastructure for the Internet of Things* aimed at making the PKI technology an effective means through which provide with mutual authentication mechanisms the forthcoming IoT scenarios. In particular PKIoT has been designed to be:

- *flexible*, in order to dynamically meet the different constraints of the involved parties;
- *compatible*, as far as possible, with the standard PKI, in order to enable transparent integration with the already deployed security infrastructure;
- *extensible*, in order to ease the development and addition of new security functionalities.

The components and the overall schema of the PKIoT architecture are depicted in Fig. 8.1.

The PKIoT architecture is a classic client/server architecture, which therefore defines two types of nodes:

- *PKIoT clients*, nodes with limited capabilities which cannot afford some or all the operations required to establish a secure communication channel with certificate-authenticated nodes.
- *PKIoT servers*, powerful nodes offering different types of services to PKIoT clients to assist them in the establishment of secure communication channels with other nodes.

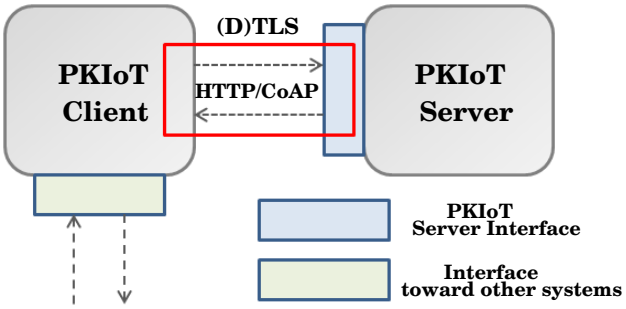


Figure 8.1: PKIoT architecture components.

PKIoT servers expose an interface which is compliant with the well-known RESTful paradigm, the extensively adopted approach in today's web service implementations due to its lightness and tight integration with commonly used application protocols. The REST resources exposed by the servers' interface, each with their own expected inputs and outputs, correspond to services that clients can request to get support in carrying out different operations.

PKIoT clients, by interacting with a certain PKIoT server through its interface and depending on their level of constrainedness and on the amount of trust they place on that server, can outsource to it various security related tasks characterized by different computational and privacy requirements. The decision about which are the services that a PKIoT server should offer, and therefore about the structure of its interface, has been taken accordingly to the outcomes of the experimental results detailed in the next section. However it is worth noting that the modularity of the defined REST interface allows to easily modify and extend it in order to integrate new security functionalities that may prove necessary, such as the ones related to the verification of attribute certificates [291].

While in theory a PKIoT client could ask for services from multiple PKIoT servers, in the typical configuration it relies on a single PKIoT server, its trust anchor. According to certain policies, a trust anchor could follow a security bootstrapping procedure and delegate to other PKIoT servers some of the functions needed by its served clients. However for the sake of simplicity and without loss of generality this chapter focuses on the most common configuration only. We also assume that PKIoT clients can always reach at least a valid PKIoT server.

In any case, the communication between PKIoT clients and their trusted servers must be secured through a cryptographic protocol which permits out-of-band mechanism for key exchange and validation, by resorting, for example, to Pre-Shared-Keys (PSK) or Raw Public Keys (RPK).

8.2.2 PKIoT server services

The Datagram Transport Layer Security (DTLS) protocol [121] is the UDP-based version of the well-know TLS protocol. Because of its lightness, and since it also supports PSK and RPK besides the traditional X.509 certificate-based key establishment mechanism, today it is considered the

de facto standard for communication security in the IoT. For this reason, DTLS has been adopted as underlying security protocol of the PKIoT architecture, and as reference point to identify the most demanding, and therefore the most beneficial to outsource, cryptographic operations.

DTLS defines a handshake protocol which must be run by the involved entities at the beginning of a new connection to initiate a secure session. Even if the handshake is performed only in the initial phase of the connection, it accounts for by far the greatest part of the overheads caused by the protocol. DTLS supports tens of cipher suites, sets of cryptographic primitives used during the handshake for key exchange, encryption and hashing. The Constrained Application Protocol (CoAP) [28], the most widely adopted application protocol in the IoT, mandates the use of the `TLS_PSK_WITH_AES_128_CCM_8` cipher suite for PSK mode and of the `TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8` cipher suite for RPK and certificate mode. This last one uses the ECDH and ECDSA algorithms for key exchange and digital signature, and the AES-based CCM algorithm with 128 bits keys and 8 bytes authentication tags for encryption and authentication. Experimental activities have been conducted on real IoT platforms to identify which are the most time and energy consuming tasks that a device has to carry out to perform a full DTLS handshake in certificate mode. The outcomes of such experiments, which are detailed in the following sections, highlighted that the followings are the DTLS operations that constrained nodes take the longest time to perform:

- **ephemeral Diffie-Hellman keys generation**, performed by the server and the client to include them in the ServerKeyExchange (flight 4) and ClientKeyExchange (flight 5) messages respectively;
- **digital signature generation**, performed by the server to sign its ephemeral Diffie-Hellman public key and the relative information to be included in the ServerKeyExchange (flight 4) message, and by the client to sign the content of the CertificateVerify message (flight 5);
- **digital signature verification**, performed by the server and the client to check the validity of the received CertificateVerify (flight 5) and ServerKeyExchange (flight 4) messages respectively;
- **certificate verification**, performed by the client and the server before starting flight 5 and flight 6 respectively;
- **pre-master secret generation**, performed by the server and the client after having received and verified the ClientKeyExchange (flight 5) and ServerKeyExchange (flight 4) messages respectively.

PKIoT servers should therefore offer services aimed at relieving PKIoT clients from the aforementioned tasks. On the other hand, nodes acting as PKIoT clients should be allowed to

Table 8.1: PKIoT Server Interface

Resource	Input	Output
/generateECDHkeys	the elliptic curve to be used to generate the keys [POST]	the generated keys
/generatePreMasterSecret	the node's ECDH private key and the other party's public key [POST]	the pre-master secret
/generateECDSASignature	the digest to sing and the signer's private key [POST]	the signed digest
/verifyECDSASignature	the signed digest to verify, the unsigned digest and the signer's public key [POST]	verified/not verified
/verifyCertificate	the certificate to verify [POST]	verified/not verified/RPK

request all and only the services they need to complete a DTLS handshake, which may vary depending on their workload and power status over time. In fact, for example, a node that is usually able to perform on his own all the operations required to carry out a DTLS handshake, might need in some periods of its operational life to resort to some or all the above listed services, either because in that moment it is involved in other intensive computations or because it is running low on batteries and wants to save energy.

With this in mind, we designed the PKIoT server RESTful interface, whose resources, one for each task, and the relative expected inputs and outputs are depicted in Tab. 8.1. In accordance with it, for example, a PKIoT client who wishes to check the validity of a given certificate, can perform a POST to the /verifyCertifcate resource of a PKIoT server providing the certificate to verify. The server then performs all the needed operations to validate the certificate, such as signature, temporal validity and revocation status check, and returns a verified/not verified answer depending on the state of the submitted certificate. In Fig. 8.2 a more complex interaction is displayed.

8.2.3 The PKIoT certificate

While the described services can effectively assist constrained nodes during the execution of security critical tasks, they leave unaddressed the problem of certificates transmission in a context where bandwidth is a scarce resource. To give an example, the IEEE 802.15.4 standard [292], the basis of the most widely adopted protocols in the IoT context, supports MAC frames of 127 bytes. So even without considering the headers of the upper layer protocols, which may significantly reduce the space available, the transmission of the google.com certificate, which is 2516 bytes, would require its fragmentation in 20 packets. This implies the introduction of overheads and delays, and great bandwidth and energy consumption exacerbated by the increased risk of packets retransmission. Moreover, in [293] the authors showed that fragmentation makes nodes vulnerable to certain types of attacks.

To overcome these issues, we propose a new type of compact certificate, the *PKIoT certificate*, which is meant to be used in conjunction with the certificate verification service offered by PKIoT servers. The idea behind this new type of certificate is that, as long as constrained nodes are not the ones who perform certificate validation, there is no need to have the full certificate sent to them. In fact, they would receive the certificates just to forward them to PKIoT servers for verification, doubling the already highlighted transmission overheads.

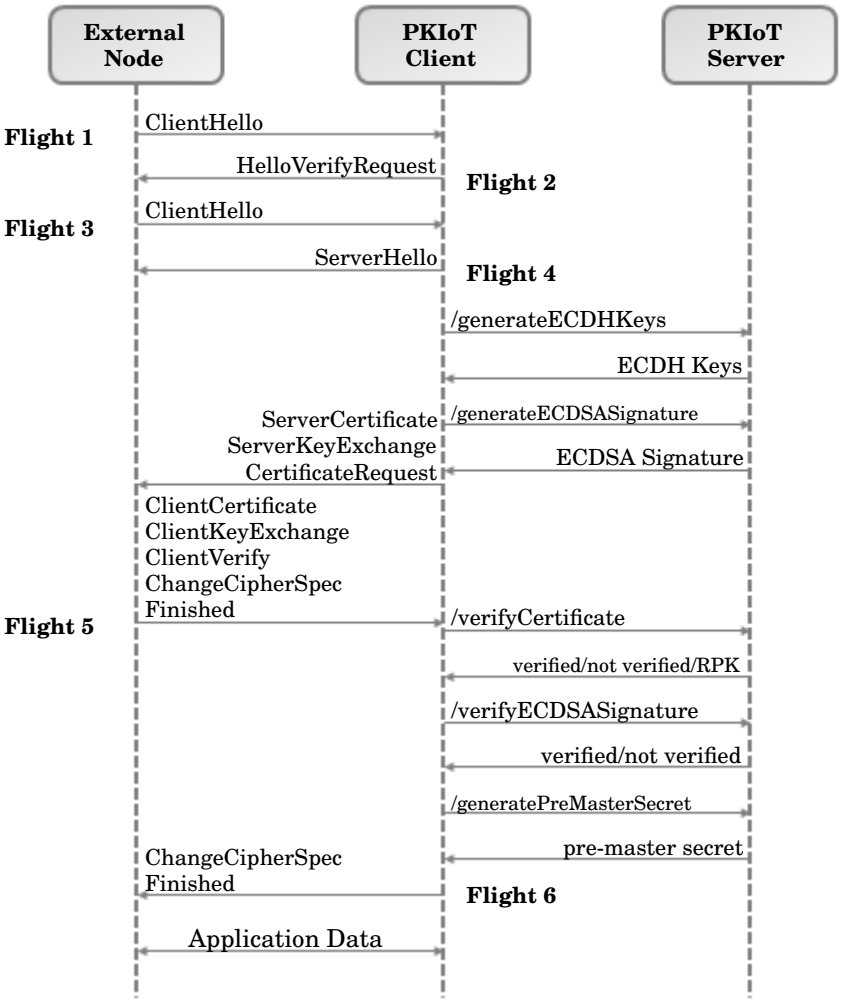


Figure 8.2: Message sequence chart depicting the interaction between a PKIoT Client and a PKIoT Server when the former uses all the services offered by the latter to perform a full DTLS handshake with an external node.

In order to avoid this, the PKIoT certificate only contains a link to the certificate, an URI through which the PKIoT server can get the full certificate for verification. Since PKIoT servers usually do not operate in environments with the same constraints of their clients' ones, the use of PKIoT certificates can considerably reduce the amount of data transmitted inside the constrained network segments.

To use the PKIoT certificate in a DTLS handshake, the Certificate message payload has been used as a container for the *CertiLink* structure. The new Certificate payload format is an extension of the one presented in [290], and it is defined as:

```
struct {
    select(certificate_type) {

        //the proposed certificate
        case PKIoTCertificate:
            CertiLink certificate;

        //RawPublicKey certificate defined
        //in [RFC7250]
        case RawPublicKey:
            opaque ASN.1_subjectPublicKey
        Info<1...2^24 - 1>;

        //X.509 certificate defined in
        //[RFC5246]
        case X.509:
            ASN.1Cert certificate_list<
            0...2^24-1>;

       //Additional certificate type based
       //on "TLS Certificate Types"
       //subregistry
    };
}Certificate;
```

The *CertiLink* structure is defined as:

```
struct {
    opaque link<1...100>;
}CertiLink;
```

When PKIoT servers are requested to validate a PKIoT certificate, they must first retrieve the full certificate following the specified link and then perform the usual verification procedure. The output of the certificate verification service cannot simply be a verified/not verified answer in this case, since the requesting clients have only access to the PKIoT certificate, which does not contain any cryptographic information, and so the clients would not know what the verified/not verified answer refers to. The service should instead return the certificate public key information if the certificate is verified, and an error message if it is not. In this context Raw Public Key certificates have been adopted as a convenient means to encapsulate and transmit public key information.

Table 8.2: Potential Security Threats when using PKIoT Services

	DoS	Eavesdropping	Tampering	Masquerade	Digital Identity Theft
ECDH keys generation	✓	✓	✓		
Pre-master secret generation	✓	✓	✓		
ECDSA signature generation	✓				✓
ECDSA signature verification	✓			✓	
Certificate verification	✓			✓	

It is worth noting that, while the adoption of PKIoT certificates can significantly reduce the overheads related to certificates transmission, their use is not mandatory for the PKIoT architecture to operate correctly. One of the main strengths of the PKIoT architecture is actually that it is completely transparent to the nodes which are not aware of it. From a not-PKIoT-enabled node point of view, in fact, there is no difference in the interactions with PKIoT-enabled and not-PKIoT-enabled nodes, since all the PKIoT operations take place in background and their outcomes are indistinguishable from the ones the nodes would get acting on their own. This feature enables the seamless integration of the PKIoT architecture into already established security infrastructures, and at the same time allows PKIoT-enabled nodes to make the most of it through the use of PKIoT certificates.

8.2.4 Security considerations on the PKIoT server services

The possibility to have a fine-grained control on which task demand support for allow nodes to select the computations to outsource taking into account also aspects such as the level of trust they place in the various PKIoT servers. In delivering their services, in fact, PKIoT servers gain knowledge of PKIoT clients' cryptographic material, either because they produce it or because they need to be provided with it to carry out their work. Such material is characterized by different levels of sensitivity depending on the potential impact that its accidental or malicious disclosure could have on their owners' security.

The risks that may arise in presence of misbehaving PKIoT servers are reported in the following grouped by service, and summarized in Tab. 8.2.

- **ephemeral Diffie-Hellman keys generation:** PKIoT clients do not need to disclose any previously held private cryptographic information to make use of this service. However they delegate to the PKIoT server, and thus disclose to it, the generation of new cryptographic material, namely the ephemeral Diffie-Hellman keys, which can be misused to derive the pre-master secret used to secure a certain connection. This would allow a malicious server to eavesdrop and tamper the messages exchanged on that connection undetected. A sever could also launch a Denial-of-Service (DoS) attack by returning invalid keys to the requester;

- **pre-master secret generation:** the risks related to this service are the same of the previous one, and are connected to the fact that in this case too the server becomes aware of the client's private ephemeral Diffie-Hellman key, because the client disclose it in order to request the service. Therefore the server becomes able to compute the pre-master secret of the connection, with all the already mentioned threats that this implies. Moreover it is interesting to note that, from a security perspective, it does not make much sense for a PKIoT client to request the first service to a PKIoT server without requesting to it also the second one and vice versa;

- **digital signature verification:** PKIoT clients do not need to disclose any secret information to request this service, since the PKIoT server just need to be fed with the signed digest to check and the relative public key. However it could still interfere with the normal client operations by maliciously claiming that a not valid signature is verified to carry out or give support to a masquerade attack. On the other hand, by claiming that valid signatures are not verified would cause the interruption of legitimate DTLS handshakes, and this would result in DoS attacks;

- **certificate verification:** the use of this service too does not entail the disclosure of any secret cryptographic material. Just like in the previous case, misbehaving PKIoT servers could launch masquerade and DoS attacks by stating that not valid certificates are valid and vice versa;

- **digital signature generation:** this service requires access to the private key of the PKIoT client, so it should be requested only to completely trusted PKIoT servers. Along with the possibility to launch DoS attacks by generating erroneous signatures, in fact, malicious servers can also steal clients' digital identity by misusing their private keys to impersonate them.

8.2.5 The PKIoT architecture in the oneM2M framework

The oneM2M architecture [104] is a major point of reference in the path towards the standardization of the IoT ecosystem. It proposes a functional architecture , depicted in Fig. 8.3, which is based on a three-layered model including an Application Layer, a Common Services Layer and a Network Services Layer. In the Application Layer reside the Application Entities (AEs), which represent application service logics. The Common Services Layer hosts the Common Service Entities (CSEs), instantiations of a set of service functions that can be invoked by the AEs or other CSEs. In the end, the Network Services Layer comprises the Network Service Entities (NSEs), which encapsulate and offer to the hosted CSEs the services provided by the underlying network such as device management, location and triggering.

All the entities can interact with each other through specific reference points, which vary depending on the type of the interacting entities. However, before any interaction can actually

occur between a CSE and other CSEs and/or AEs hosted on remote nodes, connectivity has to be established in the underlying network by following the network-specific service registration procedures specified by the Network Services Layer. After such procedures have taken place, and in general independently from them, the Security Association Establishment procedures defined by the already described Security Association Establishment Frameworks (SAEF) can be carried out.

As already stated, the SAEF allow entities to mutually authenticate, as well as enabling the encryption and integrity protection of the exchanged messages, and these represent mandatory prerequisites before any AE or CSE can use services offered by other CSEs. The security association establishment, according to the oneM2M security specifications [289], translates into a (D)TLS session protected either via PSK, RPK or X.509 certificate depending on the used framework. In an open environment where inter-M2M Service provider interactions are likely to occur, however, only the SAEFs relying on X.509 certificates, namely the certificate-based and the MAF-based ones, are able to guarantee the necessary security, scalability and flexibility levels. But, as extensively highlighted, not all the nodes operating in this context can afford to deal with this type of certificate.

It is therefore in this context that the PKIoT architecture can play a meaningful role. The oneM2M architecture can in fact leverage on the functionalities that the PKIoT architecture provides to enable the adoption of X.509 certificates in the certificate-based and MAF-based SAEFs in nodes with a wide range of characteristics, including the most constrained ones.

The services offered by the PKIoT architecture can be easily and fruitfully embedded into the oneM2M CSEs by adding them to the Security Common Service Functions (SEC CSFs) related to security association establishment. Their integration into the oneM2M architecture allows AEs and CSEs to access them just like any other oneM2M common service, transparently improving the architecture functionalities and effectively giving support to standard, secure communications between heterogeneous oneM2M nodes.

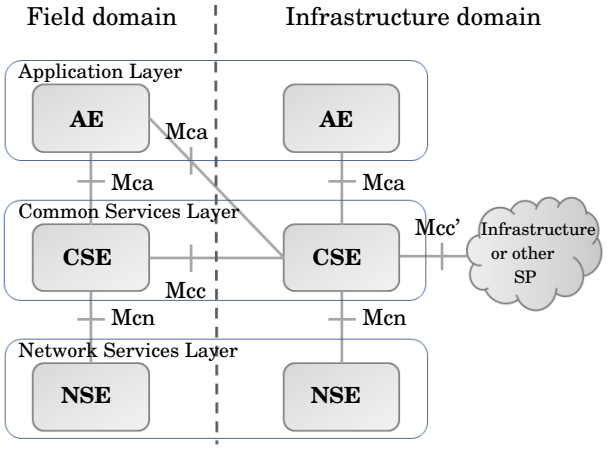


Figure 8.3: oneM2M functional architecture.

8.3 Simulation settings and implementation

The decisions regarding the selection of the services delivered by the PKIoT architecture, as well as the assessment of its performances, have been made in the light of data obtained from experimental activities conducted on real IoT platforms. In the fulfillment of such activities open-source software tools widely used in the IoT context have been taken into account, conveniently amended and extended to implement the PKIoT specific functionalities.

In more detail, the test-bed set up to evaluate the proposed solution consisted in:

- an embedded IoT platform, playing the role of the constrained node acting as a PKIoT client;
- a PC, playing the the role of the PKIoT server. The PC also hosted for convenience a desktop application acting as an external node wishing to interact with the constrained node and an online certificate directory;
- a multi-interface gateway, connected wirelessly to the constrained node and via the serial interface to the PC, providing connectivity between the two domains.

8.3.1 PKIoT client implementation

The selected embedded IoT platform was a device targeted to battery powered wireless sensor networks based on the CC2538 microcontroller, a System-on-Chip (SoC) by Texas Instruments which provides, among the other things, a low-power 32MHz ARM Cortex-M3, 512kB of flash memory, 32kB of RAM and a 2.4-GHz IEEE 802.15.4 compliant transceiver.

Contiki [294] has been selected as operating system of the IoT platform. Contiki is an open-source operating system written in C, developed and supported by a wide range of industrial and academic bodies, which is targeted to constrained wireless IoT devices. Contiki comes with a full IPv6 network stack, and it provides implementations of protocols for low-power IPv6 networking such as the 6LoWPAN adaptation layer [295] and the CoAP RESTful application-layer protocol. In its current release, however, Contiki does not comprises any implementation of the DTLS protocol. In order to overcome this issue we resorted to tinydtls [296], a lightweight C implementation of the DTLS protocol supporting the two mandatory-to-implement CoAP cipher suites, the already mentioned `TLS_PSK_WITH_AES_128_CCM_8` and `TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8`. We integrated tinydtls in Contiki and extended it with micro-ecc [297], a small and fast C implementation of the ECDH and ECDSA algorithms which outperforms the tinydtls built-in cryptographic libraries.

8.3.2 PKIoT Server implementation

The PC was an Intel Core i5-6500 with a 3.20 GHz quad-core processor running Ubuntu 14.04 and 15.6 GiB of RAM.

Table 8.3: Experimental Results

	Time (ms)		**Generated traffic (bytes)**		**Energy (mJ)**	
	Without PKIoT	With PKIoT	Without PKIoT	With PKIoT	Without PKIoT	With PKIoT
Handshake with the PKIoT Server	0	1446	0	1164	0	121.16
ECDH Keys Generation	4067	348	0	282	362.37	12.30
Pre-master Secret Generation	4075	437	0	340	363.08	15.30
ECDSA Signature Generation	4111	434	0	338	366.29	15.05
ECDSA Signature Verification	5237	429	0	312	466.62	14.17
Certificate Verfication (without/with PKIoT certificate)	5330 / -	1534 / 469	0 / -	1221 / 373	474.90 / -	57.37 / 16.57
PKIoT tasks	22820 / -	4628 / 3563	0 / -	3657 / 2809	2033.26 / -	235.35 / 194.55
Full Handshake	27254 / -	9517 / 6132	3888 / -	7547 / 4879	2531.42 / -	405.57 / 284.52

The PKIoT server and the desktop application have been realized using Californium and Scandium [298], Java implementations of the CoAP and DTLS protocols. Both tinydtls and Scandium have been modified to work also with PKIoT certificates. For the generation of standard X.509 certificates we used OpenSSL [299], an open-source implementation of the SSL and TLS protocols which also provides various utility functions, including those related to certificate management.

In particular we generated two X.509 certificates, one for the constrained node and one for the desktop application, using ECC as certificate key algorithm and secp256r1 as elliptic curve for the generation of the ECC keys. Public keys generated from the selected curve consist in two values of 32 bytes each. The size of the two full X.509 certificates was about 800 kb. The certificates have been stored at the node and application sides, and they have also been made retrievable through a web server running on the same PC hosting the PKIoT server and the desktop application. The URLs at where they could be downloaded have been included into the relative PKIoT certificates, which have been handed out to the two parties as well.

8.3.3 Gateway implementation

The gateway was a BeagleBone Black board running Linux and equipped with a Wireless Connectivity Cape. The Contiki border router application has been used on the gateway during the experiments.

8.4 Experimental results and discussion

With the aforedescribed layout drawn up, we conducted our experiments taking into account three scenarios. In the baseline scenario neither the constrained node nor the desktop application took advantage of the PKIoT architecture, so they performed the usual DTLS handshake. In the second scenario only the constrained node was a PKIoT-enabled node. In that case it could leverage on the services provided by the PKIoT server, but it could not use PKIoT certificates. In the third scenario both the constrained node and the desktop application were PKIoT-enabled, so they could also resort to PKIoT certificates. In all the experiments carried out the desktop application

was the party starting the connection, and the communication between the constrained node and the PKIoT server, when present, was secured by using DTLS in PSK mode.

The parameters we took into account while performing our tests were the execution times related to the DTLS handshake and the corresponding consumed energy and generated traffic in the constrained network segment. The experiments conducted in the baseline scenario allowed us to identify the most time and energy consuming tasks, which therefore have been selected for outsourcing. In Tab. 8.3 the average values of the measurements related to the three considered scenarios are recorded. In particular, we reported the values for the PKIoT tasks taken individually, the sum of such values and the values for the full DTLS handshake. Since the second and third scenarios just differ for the values pertaining to the certificate verification task, the related columns of the table have been collapsed into one for the sake of better readability, highlighting the only changed value.

It is important to stress that the PKIoT tasks values do not sum up to the full handshake values, since there are also other computations and transmissions carried out by the node during the DTLS handshake which are not covered by the PKIoT tasks. Nevertheless, as can be seen from Tab. 8.3, the PKIoT tasks in the baseline scenario account for about 84% of the full handshake execution time and for about 80% of the total consumed energy.

The achieved results show that the adoption of the PKIoT architecture brings significant benefits in terms of execution time, which can be reduced up to 78% when outsourcing all the PKIoT tasks and resorting to PKIoT certificates. Also when PKIoT certificates are not used, thus enabling transparent interoperability with not-PKIoT-enabled devices, the advantages brought by the PKIoT architecture are still remarkable, since it allows to reduce by 65% the execution time.

Of course, outsourcing the PKIoT tasks to a remote server cause overheads in terms of traffic generated by the consequent interactions with it, and this is an aspect to take into account when talking about constrained network scenarios. From an energy consumption point of view, however, the introduced transmission and reception operations only partially obfuscate the benefits resulting from the outsourcing of energy hungry tasks. In fact, even with these overheads, it is possible by leveraging the PKIoT architecture to reduce by 89% the consumed energy when all the PKIoT tasks are outsourced and PKIoT certificates are used, and by 84% when PKIoT certificates are not used.

It is worth noting that, in order to evaluate the time and energy requirements of the certificate validation process, in our tests we used for convenience X.509 certificates directly signed by a root Certification Authority trusted by both the desktop application and the constrained node, reducing the validation process to a simple digital signature verification. In real scenarios, however, this is not applicable: nodes have to send certificate chains, which are clearly bigger than single certificates, and they have to perform the already mentioned certification path discovery and validation procedures, which have been omitted in our tests and that could bring

unacceptable overheads for a wide range of constrained devices. The benefits brought by the PKIoT architecture are therefore greater than the ones reported above, since it prevents the transmission of big certificate chains and relieve constrained nodes of performing procedures that would be too burdensome for most of them.

While the reported improvements are comparable with the ones obtained by other approaches following a similar delegation-based philosophy, such as the aforementioned [140] and [120], our solution introduces a higher level of flexibility that will allow next generation IoT objects to autonomously decide which security tasks outsource on a case-by-case basis, depending on a wide range of internal and external conditions. Moreover already deployed infrastructures and services will not need any modification to interact with PKIoT enabled devices. In the end, the PKIoT architecture has been designed to be naturally integrated in a wide-supported standard framework, namely oneM2M. This is in our opinion a crucial aspect promoting the adoption of our architecture that is not addressed by any other proposed solution.

8.5 Conclusions

The IoT world is undergoing a paradigm shift that will soon turn today's closed and static IoT applications into open, dynamic, horizontally integrated ones. The entities participating in this new not pre-configured and ever-evolving scenario will need standard mechanisms to authenticate each other and protect their communications. Public Key Infrastructure and certificates, extensively used in the Internet for this purpose, cannot be employed right away in this emerging context because of the heterogeneity of the involved entities, which may not be able to perform all the required operations.

In this chapter we presented PKIoT, an architecture which aims at making certificate-based authentication affordable also for constrained IoT devices. The PKIoT architecture allows IoT nodes to outsource security related demanding tasks to a remote server, which supports them in the establishment of a secure connection with a third party in a completely transparent way as regards the latter. Nodes can flexibly select the tasks to outsource in accordance with their current status and with the trust they place on the server. Such features, in addition to the ease with which it is possible to integrate the architecture with new security services by simply modifying the REST interface of PKIoT servers, make the PKIoT architecture a flexible, compatible and extensible solution.

We also introduced a new type of compact certificate, the PKIoT certificate, whose use instead of usual X.509 certificates allows to further reduce the transmission overheads, but require both the involved parties to be PKIoT-enabled.

Bearing in mind the importance that standard architecture will have in the depicted forthcoming scenario, we also highlighted how the PKIoT architecture can be integrated into oneM2M, the widely promoted architecture for standard machine-2-machine communications, to support

its SAEFs.

In the end, we reported the results of our experimental activities, which showed that it is possible to significantly reduce the execution times and the nodes energy consumption by leveraging on the PKIoT architecture functionalities.

The PKIoT architecture has therefore proved to be an effective means to enable standard and secure communications in the IoT world of the future.

CHAPTER 9

CONCLUSIONS

The main motivation that has governed this thesis was the delineation and study of forthcoming IoT scenarios and their enabling technologies, and the development of approaches paving the way to the softwarized and secure IoT ecosystems of the future.

Chapter 2 and chapter 3 gave a comprehensive overview of the trends which are affecting distributed systems in general and IoT systems in particular. The techno-economical drivers which are leading to an increasing decentralization, dynamicity and openness of such systems, and to the unprecedented programmability, flexibility and pervasivity of forthcoming 5G network infrastructures, are extensively analyzed. In this respect, the thesis reports a survey of the enabling technologies of those emerging scenarios, namely automatic contract management, service discovery and mutual authentication technologies, and proposes the concept of a 5G Operating System, i.e. a distributed platform, to enable an automated and efficient management and orchestration of 5G network infrastructures.

Chapter 4 presented a softwarized IoT-based system to monitor urban mobility in real time. The visual sensor nodes of the system run a specifically developed middleware which includes a management module enabling dynamic allocation, management and tuning of high-level tasks on IoT nodes, in line with softwarization trends. Moreover, in order to promote the exploitation of this component as a means to incorporate highly-programmable IoT systems into the 5G network fabric, chapter 5 proposes its integration into the oneM2M architecture to implement specific oneM2M common service functions.

In the end, chapters from 6 to 8 addressed security aspects of next generation smart environments. In more detail, chapter 6 deals with usage control frameworks working in synergy with smart cameras based video surveillance systems to monitor users behaviour. In this context the chapter proposed a technique limiting the disclosure of users' attributes only to those

which are relevant to the framework for the evaluation of usage control policies, while keeping secret users' identities as long as they act correctly. Chapter 7 focused on geolocation-aware applications running on mobile devices. Here an approach has been proposed to preserve users' location privacy while they are using applications requiring access to their position. In the end, chapter 8 presented an architecture to enable certificate-based mutual authentication in constrained IoT scenarios. The proposed architecture allows to flexibly delegate the most burdensome cryptographic tasks related to certificate validation and to reduce the consumed bandwidth by leveraging a new type of compact certificate. The chapter also highlights the role that the PKIoT architecture can play in the context of the oneM2M security framework in enabling standard secure machine-2-machine interactions.

PUBLICATIONS

- E. Carniani, G. Costantino, **F. Marino**, F. Martinelli, P. Mori, "Enhancing video surveillance with usage control and privacy-preserving solutions", *Journal of Wireless Mobile Networks, Ubiqitous Computing and Dependable Applications*, vol. 3, n. 1, pp. 20-40, 2016.

- **F. Marino**, L. Maggiani, L. Nao, P. Pagano, M. Petracca, "Towards softwarization in the IoT: Integration and evaluation of t-res in the oneM2M architecture", *2017 IEEE Conference on Network Softwarization (NetSoft)*, pp. 1-5, 2017.

- G. R. Leone, D. Moroni, G. Pieri, M. Petracca, O. Salvetti, A. Azzarà, **F. Marino**, "An intelligent cooperative visual sensor network for urban mobility", *Sensors*, vol. 17, n. 1, article no. 2588, 2017.

- G. Bella, G. Costantino, **F. Marino**, F. Martinelli, "Getmewhere: A Location-Based Privacy-Preserving Information Service", *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pp. 529-532, 2018.

- A. Manzalini, **F. Marino**, "Operating Systems for 5G Services Infrastructures: Convergence between IT and Telecommunications industry structures", *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pp 1-5, 2018.

- **F. Marino**, C. Moiso, M. Petracca, "Automatic contract negotiation, service discovery and mutual authentication solutions: a survey on the enabling technologies of the forthcoming IoT ecosystems", *Computer Networks*, vol. 148C, pp. 176-195, 2019.

- **F. Marino**, C. Moiso, M. Petracca, "PKIoT: a Public Key Infrastructure for the Internet of Things", paper submitted to *Ad Hoc Networks* on 16 May 2018.

BIBLIOGRAPHY

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, pp. 2347–2376, Fourthquarter 2015.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645 – 1660, 2013.

[5] S. Sicari, A. Rizzardi, L. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, pp. 146 – 164, 2015.

[6] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for Internet of Things," *Journal of Network and Computer Applications*, vol. 42, pp. 120 – 134, 2014.

[7] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," *IEEE Communications Surveys Tutorials*, vol. 17, pp. 1294–1312, thirdquarter 2015.

[8] C. W. Tsai, C. F. Lai, M. C. Chiang, and L. T. Yang, "Data Mining for Internet of Things: A Survey," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 77–97, First 2014.

[9] Y. Qin, Q. Z. Sheng, N. J. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: A survey on data-centric internet of things," *Journal of Network and Computer Applications*, vol. 64, pp. 137 – 153, 2016.

[10] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 414–454, First 2014.

[11] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 3, pp. 70–95, Feb 2016.

[12] O. Hahm, E. Baccelli, H. Petersen, and N. Tsiftes, "Operating Systems for Low-End Devices in the Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 3, pp. 720–734, Oct 2016.

[13] "TerraSwarm Project." `https://ptolemy.berkeley.edu/projects/terraswarm/index.html`.

[14] R. Inam, A. Karapantelakis, K. Vandikas, L. Mokrushin, A. V. Feljan, and E. Fersman, "Towards automated service-oriented lifecycle management for 5G networks," in *IEEE Conference on Emerging Technologies & Factory Automation*, (Luxembourg), pp. 1–8, Sept. 8–1, 2015.

[15] C. Sierra, V. Botti, and S. Ossowski, "Agreement computing," *KI - Künstliche Intelligenz*, vol. 25, pp. 57–61, Mar. 2011.

[16] L. Paliulionienė, "On description of contracts and agreements in the context of SOA," *Computational Science Techniques*, vol. 1, pp. 171–183, 2013.

[17] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, 2005.

[18] Open Grid Forum, "Service-Oriented Architecture Ontology," tech. rep., Open Group, 2010.

[19] M. Wright and A. Reynolds, *Oracle SOA Suite 11g R1 Developer's Guide*. Packt Publishing, 2010.

[20] W3C, "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language," 2007. `https://www.w3.org/TR/wsdl20/`.

[21] W. Emmerichand, S. Shrivastava, F. Panzieri, J. Crowcroft, and W. Beckmann, "TAPAS - Trusted and Quality-of-Service Aware Provision of Application Services," 2004. `http://www.ercim.eu/publication/Ercim_News/enw58/emmerich.html`.

[22] A. Paschke, "RBSLA A declarative Rule-based Service Level Agreement Language based on RuleML," in *International Conference on Computational Intelligence for Modelling, Control and Automation & International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, (Vienna), pp. 308–314, Nov. 28-30, 2005.

[23] H. Ludwig and A. Keller and A. Dan and R. P. King and R. Franck, "Web Service Level Agreement (WSLA) Language Specification," tech. rep., IBM Corporation, 2003.

[24] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, "Web Services Agreement Specification (WS-Agreement)," tech. rep., Open Grid Forum, 2007.

[25] W3C, "SOAP Version 1.2," 2007. `https://www.w3.org/TR/soap/`.

[26] R. T. Fielding and R. N. Taylor, *Architectural styles and the design of network-based software architectures*.
PhD thesis, University of California, Irvine, 2000.

[27] M. Laine, "RESTful Web Services for the Internet of Things," 2012.
`http://media.tkk.fi/webservices/personnel/markku_laine/restful_web_services_for_the_internet_of_things.pdf`.

[28] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)." RFC 7252, June 2014.

[29] W3C, "Web Application Description Language," 2009.
`https://www.w3.org/Submission/wadl/`.

[30] R. Kübert, G. Katsaros, and T. Wang, "A RESTful implementation of the WS-agreement specification," in *International Workshop on RESTful Design*, (Hyderabad), pp. 67–72, Mar. 28, 2011.

[31] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, M. J. Wooldridge, and C. Sierra, "Automated negotiation: prospects, methods and challenges," *Group Decision and Negotiation*, vol. 10, pp. 199–215, 2001.

[32] O. Waeldrich, D. Battrè, F. Brazier, K. Clark, M. Oey, A. Papaspyrou, P. Wieder, and W. Ziegler, "WS-Agreement Negotiation Version 1.0," tech. rep., Open Grid Forum, 2011.

[33] E. Mingozzi, G. Tanganelli, and C. Vallati, "A framework for QoS negotiation in things-as-a-service oriented architectures," in *International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems*, (Aalborg), pp. 1–5, May 11–14, 2014.

[34] WSAG4J.
`http://wsag4j.sourceforge.net/site/`.

[35] V. Casola, A. D. Benedictis, M. Rak, G. Aversano, and U. Villano, "An SLA-Based Approach to Manage Sensor Networks as-a-Service," in *International Conference on Cloud Computing Technology and Science*, (Bristol), pp. 191–197, Dec. 2-5, 2013.

[36] K. Mišura and M. Žagar, "Negotiation in Internet of Things," *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, vol. 57, no. 2, pp. 304–318, 2017.

[37] R. Vigne, J. Mangler, E. Schikuta, and S. Rinderle-Ma, "WS-Agreement based service negotiation in a heterogeneous service environment," in *IEEE International Conference on Service-Oriented Computing and Applications*, (Taipei), pp. 1–8, Dec. 17-19, 2012.

[38] R. Liu, W. Wu, H. Zhu, and D. Yang, "M2M-Oriented QoS Categorization in Cellular Network," in *International Conference on Wireless Communications, Networking and Mobile Computing*, (Wuhan), pp. 1–5, Sept. 23–25, 2011.

[39] M.-A. Nef, L. Perlepes, S. Karagiorgou, G. I. Stamoulis, and P. K. Kikiras, "Enabling QoS in the Internet of Things," in *International Conference on Communications, Theory, Reliability, and Quality of Service*, pp. 33–38, 2012.

[40] Z. Ming and M. Yan, "A modeling and computational method for QoS in IOT," in *IEEE International Conference on Computer Science and Automation Engineering*, (Beijing), pp. 275–279, June 22–24, 2012.

[41] R. Duan, X. Chen, and T. Xing, "A QoS Architecture for IOT," in *International Conference on Internet of Things and International Conference on Cyber, Physical and Social Computing*, (Dalian), pp. 717–720, Oct. 19–22, 2011.

[42] X. Zheng, P. Martin, K. Brohman, and L. Da Xu, "Cloud service negotiation in internet of things environment: A mixed approach," *IEEE Transactions on Industrial Informatics*, vol. 10, pp. 1506–1515, 2014.

[43] O. Rana, M. Warnier, T. B. Quillinan, and F. Brazier, "Monitoring and Reputation Mechanisms for Service Level Agreements," in *International workshop on Grid Economics and Business Models*, (Las Palmas De Gran Canaria), pp. 125–139, Aug. 26, 2008.

[44] S. Venticinque, R. Aversa, B. D. Martino, M. Rak, and D. Petcu, "A Cloud Agency for SLA Negotiation and Management," in *Euro-Par 2010 Parallel Processing Workshops* (M. R. Guarracino, F. Vivien, J. L. Träff, M. Cannatoro, M. Danelutto, A. Hast, F. Perla, A. Knüpfer, B. D. Martino, and M. Alexander, eds.), pp. 587–594, Springer Berlin Heidelberg, 2011.

[45] H. Zhang, L. Ye, J. Shi, X. Du, and M. Guizani, "Verifying cloud service-level agreement by a third-party auditor," *Security and Communication Networks*, vol. 7, pp. 492–502, 2013.

[46] A. Maarouf, A. Marzouk, and A. Haqiq, "Towards a Trusted third party based on Multi-agent systems for automatic control of the quality of service contract in the Cloud Computing," in *2015 International Conference on Electrical and Information Technologies (ICEIT)*, pp. 311–315, March 2015.

[47] D. Khader, J. Padget, and M. Warnier, "Reactive Monitoring of Service Level Agreements," in *Grids and Service-Oriented Architectures for Service Level Agreements* (P. Wieder, R. Yahyapour, and W. Ziegler, eds.), pp. 13–22, Springer US, 2010.

[48] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. `http://bitcoin.org/bitcoin.pdf`.

[49] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[50] Protocol Labs, "Filecoin: a Decentralized Storage Network," 2018. `https://filecoin.io/filecoin.pdf`.

[51] "EtherAPIs," 2018.
https://etherapis.io/.

[52] K. Noyen, D. Volland, D. Wörner, and E. Fleisch, "When Money Learns to Fly: Towards Sensing as a Service Applications Using Bitcoin," *CoRR*, vol. abs/1409.5841, 2014.

[53] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[54] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.

[55] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," in *2017 19th International Conference on Advanced Communication Technology (ICACT)*, pp. 464–467, Feb 2017.

[56] P. Holl, E. Scepankova, and F. Matthes, "Smart Contract based API usage tracking on the Ethereum Blockchain," in *Software Engineering und Software Management 2018* (M. Tichy, E. Bodden, M. Kuhrmann, S. Wagner, and J.-P. Steghöfer, eds.), (Bonn), pp. 237–239, Gesellschaft für Informatik, 2018.

[57] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an Optimized BlockChain for IoT," in *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 173–178, April 2017.

[58] M. Samaniego and R. Deters, "Blockchain as a Service for IoT," in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 433–436, Dec 2016.

[59] "The Tangle," 2017.
https://iotatoken.com/IOTA_Whitepaper.pdf.

[60] C. N. Ververidis and G. C. Polyzos, "Service discovery for mobile Ad Hoc networks: a survey of issues and techniques," *IEEE Commununications Surveys & Tutorials*, vol. 10, pp. 30–45, Sept. 2008.

[61] D. G. Reina, S. L. Toral, F. Barrero, N. Bessis, and E. Asimakopoulou, "The Role of Ad Hoc Networks in the Internet of Things: A Case Scenario for Smart Environments," in *Internet of Things and Inter-cooperative Computational Technologies for Collective Intelligence* (N. Bessis, F. Xhafa, D. Varvarigou, R. Hill, and M. Li, eds.), pp. 89–113, Springer Berlin Heidelberg, 2013.

[62] S. K. Datta, R. P. F. Da Costa, and C. Bonnet, "Resource discovery in Internet of Things: Current trends and future standardization aspects," in *IEEE World Forum on Internet of Things*, (Milan), pp. 542–547, Dec. 14–16, 2015.

[63] UDDI XML.org, "UDDI technical white paper," 2000.
http://www.uddi.org/pubs/Iru_UDDI_Technic\al_White_Paper.pdf.

[64] Sun Microsystem, "JINI architecture specification," 1999.
https://river.apache.org/doc/specs/html/jini-spec.html.

[65] GS1, "Draft Standard Specification GS1 Object Name Service (ONS)," 2013.
http://www.gs1.org/sites/default/files/docs/epc/ons_2_0_1-standard-20130131.pdf.

[66] J. Mitsugi, Y. Sato, M. Ozawa, and S. Suzuki, "An integrated device and service discovery with UPnP and ONS to facilitate the composition of smart home applications," in *IEEE World Forum on Internet of Things*, (Seoul), pp. 400–404, Mar. 6–8, 2014.

[67] A. J. Jara, P. Lopez, D. Fernandez, J. F. Castillo, M. A. Zamora, and A. F. Skarmeta, "Mobile Digcovery: A Global Service Discovery for the Internet of Things," in *International Conference on Advanced Information Networking and Applications Workshops*, (Barcelona), pp. 1325–1330, Mar. 25–28, 2013.

[68] S. K. Datta and C. Bonnet, "Search engine based resource discovery framework for Internet of Things," in *IEEE Global Conference on Consumer Electronics*, (Osaka), pp. 83–85, Oct. 27–30, 2015.

[69] Z. Shelby, "Constrained RESTful Environments (CoRE) Link Format." RFC 6690, Aug. 2012.

[70] I. Ishaq, J. Hoebeke, J. Rossey, E. De Poorter, I. Moerman, and P. Demeester, "Facilitating Sensor Deployment, Discovery and Resource Access Using Embedded Web Services," in *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, (Palermo), pp. 717–724, July 4–6, 2012.

[71] T. A. Butt, I. Phillips, L. Guan, and G. Oikonomou, "TRENDY: an adaptive and context-aware service discovery protocol for 6LoWPANs," in *International workshop on the Web of Things*, (Newcastle), pp. 1–6, June 19, 2012.

[72] Salutation Consortium, "Salutation Architecture Specification," 1999.
http://web.archive.org/web/20030623193812/www.salutation.org/.

[73] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol, Version 2." RFC 2608, June 1999.

[74] U. C. Kozat and L. Tassiulas, "Service discovery in mobile ad hoc networks: an overall perspective on architectural choices and network layer support issues," *Ad Hoc Networks*, vol. 2, pp. 2–44, Jan. 2004.

[75] F. Sailhan and V. Issarny, "Scalable Service Discovery for MANET," in *International Conference on Pervasive Computing and Communications*, (Kawai Island), pp. 235–244, 2005.

[76] M. Liu, T. Leppänen, E. Harjula, Z. Ou, M. Ylianttila, and T. Ojala, "Distributed Resource Discovery in the Machine-to-Machine Applications," in *International Conference on Mobile Ad-Hoc and Sensor Systems*, (Hangzhou), pp. 411–412, Oct. 14–16, 2013.

[77] M. Klein, B. Konig-Ries, and P. Obreiter, "Service rings - a semantic overlay for service discovery in ad hoc networks," in *International Workshop on Database and Expert Systems Applications, 2003. Proceedings*, pp. 180–185, Sept. 1–5, 2003.

[78] G. Schiele, C. Becker, and K. Rothermel, "Energy-efficient cluster-based service discovery for Ubiquitous Computing," in *European workshop on ACM SIGOPS*, (Leuven), Sept. 19-22, 2004.

[79] S. Cirani, L. Davoli, G. Ferrari, R. Léone, P. Medagliani, M. Picone, and L. Veltri, "A Scalable and Self-Configuring Architecture for Service Discovery in the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, pp. 508–521, Oct. 2014.

[80] F. Paganelli and D. Parlanti, "A DHT-Based Discovery Service for the Internet of Things," *Journal of Computer Networks and Communications*, vol. 2012, 2012.

[81] Microsoft Corporation, "Universal Plug and Play: Background," 1999. `http://www.upnp-hacks.org/upnp.html`.

[82] Bluetooth SIG, "Specification of the Bluetooth® System," 2014. `https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?docid=286439`.

[83] D. Chakraborty, A. Joshi, Y. Yesha, and T. Finin, "Toward Distributed service discovery in pervasive computing environments," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 97–112, Feb. 2006.

[84] O. Ratsimor, D. Chakraborty, A. Joshi, and T. Finin, "Allia: alliance-based service discovery for ad-hoc environments," in *International Workshop on Mobile Commerce*, (Atlanta), pp. 1–9, Sept. 2002.

[85] Z. Gao, X. Yang, T. Ma, and S. Cai, "RICFFP: An Efficient Service Discovery Protocol for MANETs," in *Embedded and Ubiquitous Computing* (L. T. Yang, M. Guo, G. R. Gao, and N. K. Jha, eds.), pp. 786–795, Springer Berlin Heidelberg, 2004.

[86] C. Lee, S. Helal, and W. Lee, "Gossip-Based Service Discovery in Mobile Ad Hoc Networks," *IEICE-Transactions on Communications*, vol. E89-B, pp. 2621–2624, Sept. 2006.

[87] S. Motegi, K. Yoshihara, and H. Horiuchi, "Service discovery for wireless ad hoc networks," in *International Symposium on Wireless Personal Multimedia Communications*, pp. 232–236, Oct. 27–30, 2002.

[88] P. T. Nguyen and A. Aggarwal, "Enhanced DNS-based service discovery in an internet of things (IoT) environment," Feb. 27 2018.
US Patent 9,906,605.

[89] P. Vandwalle, T. Thomas, and C. F. Dominguez, "Efficient service advertisement and discovery in a peer-to-peer networking environment with cooperative advertisement," Apr. 5 2016.
US Patent 9,306,813.

[90] S. K. Datta, "Towards securing discovery services in internet of things," in *Consumer Electronics (ICCE), 2016 IEEE International Conference on*, pp. 506–507, IEEE, 2016.

[91] D. J. Wu, A. Taly, A. Shankar, and D. Boneh, "Privacy, discovery, and authentication for the internet of things," in *European Symposium on Research in Computer Security*, pp. 301–319, Springer, 2016.

[92] G. Cherian and S. P. Abraham, "Efficient infrastructure service discovery with security," June 6 2017.
US Patent 9,674,048.

[93] P. E. Engelstad, Y. Zheng, R. Koodli, and C. Perkins, "Service discovery architectures for on-demand ad hoc networks," *International Journal of Ad Hoc and Sensor Networks*, vol. 2, pp. 27–58, Mar. 2006.

[94] Microformats.org, "Microformats."
`http://microformats.org/`.

[95] M. Zhou and Y. Ma, "A web service discovery computational method for IOT system," in *IEEE International Conference on Cloud Computing and Intelligence Systems*, (Hangzhou), pp. 1009–1012, Oct. 30–Nov. 1, 2012.

[96] J. Quevedo, M. Antunes, D. Corujo, D. Gomes, and R. L. Aguiar, "On the application of contextual iot service discovery in information centric networks," *Computer Communications*, vol. 89, pp. 117–127, 2016.

[97] S. Mayer and D. Guinard, "An extensible discovery service for smart things," in *International Workshop on Web of Things*, (San Francisco), June 12, 2011.

[98] S. Alam and J. Noll, "A Semantic Enhanced Service Proxy Framework for Internet of Things," in *International Conference on Green Computing and Communications & International Conference on Cyber, Physical and Social Computing*, (Hangzhou), pp. 488–495, Dec. 18–20, 2010.

[99] C. Perera and A. V. Vasilakos, "A knowledge-based resource discovery for Internet of Things," *Knowledge-Based Systems*, vol. 109, pp. 122 – 136, 2016.

[100] F. E. Castillo-Barrera and H. A. Duran-Limon, "Knowledge Capitalization in a Component-Based Software Factory: a Semantic Viewpoint," in *LA-NMR*, pp. 105–114, 2011.

[101] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. L. Phuoc,

L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25 – 32, 2012.

[102] S. Huber, R. Seiger, A. Kühnert, and T. Schlegel, "Using semantic queries to enable dynamic service invocation for processes in the internet of things," in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, pp. 214–221, Feb 2016.

[103] H. Li, D. Seed, B. Flynn, C. Mladin, and R. D. Girolamo, "Enabling Semantics in an M2M/IoT Service Delivery Platform," in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, pp. 206–213, Feb 2016.

[104] "ETSI TS 118 001 V.2.10.0: Functional Architecture," 2016. `http://www.onem2m.org/images/files/deliverables/Release2/TS-0001-%20Functional_Architecture-V2_10_0.pdf`.

[105] Z. Li and G. Gong, "A Survey on Security in Wireless Sensor Networks," *Journal of the Korea Institute of Information Security and Cryptology*, vol. 18-6B, pp. 233–248, 2008.

[106] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *ACM conference on Computer and communications security*, (Washington), pp. 41–47, Nov. 2002.

[107] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hungar. Acad. Sci*, vol. 5, pp. 17–61, 1960.

[108] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Symposium on Security and Privacy*, pp. 197–213, May 11–14, 2003.

[109] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Transactions on Information and System Security*, vol. 8, pp. 41–77, Feb. 2005.

[110] O. Garcia-Morchon, S. L. Keoh, S. Kumar, P. Moreno-Sanchez, F. Vidal-Meca, and J. H. Ziegeldorf, "Securing the IP-based internet of things with HIP and DTLS," in *ACM conference on Security and privacy in wireless and mobile networks*, (Budapest), pp. 119–124, Apr. 17–19, 2013.

[111] P. Loree and K. Nygard, "Post Deployment Secure Key Management in Wireless Ad hoc Networks," in *International Conference on Future Computational Technologies and Applications*, (Rome), pp. 47–53, Mar. 20–24, 2016.

[112] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences," in *Advances in Cryptology - CRYPTO 92* (E. F. Brickell, ed.), pp. 471–486, Springer Berlin Heidelberg, 1993.

[113] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security*, vol. 8, pp. 228–258, May 2005.

[114] R. Blom, "An optimal class of symmetric key generation systems," in *Workshop on the Theory and Application of Cryptographic Techniques*, pp. 335–338, 1984.

[115] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," in *ACM conference on Computer and communications security*, (Washington), pp. 62–72, Oct. 27–30, 2003.

[116] B. Dutertre, S. Cheung, and J. Levy, "Lightweight key management in wireless sensor networks by leveraging initial trust," tech. rep., Technical Report SRI-SDL-04-02, SRI International, 2004.

[117] R. Anderson, H. Chan, and A. Perrig, "Key infection: smart trust for smart dust," in *IEEE International Conference on Network Protocols*, pp. 206–215, Oct. 5–8, 2004.

[118] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The Kerberos Network Authentication Service (V5)." RFC 4120, July 2005.

[119] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, pp. 521–534, Sept. 2002.

[120] S. Raza, L. Seitz, D. Sitenkov, and G. Selander, "S3k: Scalable security with symmetric keys - dtls key establishment for the internet of things," *IEEE Transactions on Automation Science and Engineering*, vol. 13, pp. 1270–1280, July 2016.

[121] E. Rescorla and N. Modadugu, " Datagram Transport Layer Security Version 1.2." RFC 6347, Jan. 2012.

[122] M. Sethi, P. Kortoçi, M. D. Francesco, and T. Aura, "Secure and low-power authentication for resource-constrained devices," in *2015 5th International Conference on the Internet of Things (IOT)*, pp. 30–36, Oct 2015.

[123] Fan Wu and Lili Xu and Saru Kumari and Xiong Li and Jian Shen and Kim-Kwang Raymond Choo and Mohammad Wazid and Ashok Kumar Das, "An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment," *Journal of Network and Computer Applications*, vol. 89, pp. 72 – 85, 2017.

[124] N. Park and N. Kang, "Mutual Authentication Scheme in Secure Internet of Things Technology for Comfortable Lifestyle," *Sensors*, vol. 16, no. 1, 2016.

[125] M. A. Jan, P. Nanda, X. He, Z. Tan, and R. P. Liu, "A Robust Authentication Scheme for Observing Resources in the Internet of Things Environment," in *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 205–211, Sept 2014.

[126] S. Yoon and J. Kim, "Mutual Authentication Scheme for Lightweight IoT Devices," in *SECURWARE 2017 : The Eleventh International Conference on Emerging Security Information, Systems and Technologies*, pp. 77–78, Sep 2017.

[127] 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, May 2008.

[128] S. Frankel and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap." RFC 6071, Feb. 2011.

[129] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2." RFC 5246, Aug. 2008.

[130] E. Rescorla, "HTTP Over TLS." RFC 2818, May 2000.

[131] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, and U. Roedig, "Securing communication in 6LoWPAN with compressed IPsec," in *International Conference on Distributed Computing in Sensor Systems and Workshops*, (Barcelona), pp. 1–8, June 27–29, 2011.

[132] J. Hui and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks." RFC 6282, Sept. 2011.

[133] S. Kent, "IP Authentication Header." RFC 4302, Dec. 2005.

[134] S. Kent, "IP Encapsulating Security Payload (ESP)." RFC 4303, Dec. 2005.

[135] S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt, "Lithe: Lightweight Secure CoAP for the Internet of Things," *IEEE Sensors Journal*, vol. 7, pp. 3711–3720, Aug. 2013.

[136] S. Raza, T. Voigt, and V. Jutvik, "Lightweight IKEv2: A Key Management Solution for both the Compressed IPsec and the IEEE 802.15.4 Security," in *IETF Workshop on Smart Object Security*, 2012.

[137] R. Falk and S. Fries, "Managed Certificate Whitelisting - A Basis for Internet of Things Security in Industrial Automation Applications," in *International Conference on Emerging Security Information, Systems and Technologies*, Nov. 2014.

[138] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, "A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication," in *IEEE Conference on Local Computer Networks Workshops*, (Clearwater), pp. 956–963, Oct. 22–25, 2012.

[139] H. Shafagh and A. Hithnawi, "Poster Abstract: Security Comes First, a Public-key Cryptography Framework for the Internet of Things," in *IEEE International Conference on Distributed Computing in Sensor Systems*, (Marina Del Rey), May 26–28, 2014.

[140] R. Hummen, J. H. Ziegeldorf, H. Shafagh, S. Raza, and K. Wehrle, "Towards viable certificate-based authentication for the internet of things," in *ACM workshop on Hot topics on wireless network security and privacy*, (Budapest), pp. 37–42, Apr. 19, 2013.

[141] K. Maletsky, "RSA vs ECC Comparison for Embedded Systems," tech. rep., Atmel Corporation, 2015.

[142] Certicom, "Explaining Implicit Certificates," tech. rep., Certicom Corporation, 2004.

[143] N. Li, D. Liu, and S. Nepal, "Lightweight Mutual Authentication for IoT and Its Applications," *IEEE Transactions on Sustainable Computing*, vol. 2, pp. 359–370, Oct 2017.

[144] S. Sciancalepore, A. Capossele, G. Piro, G. Boggia, and G. Bianchi, "Key Management Protocol with Implicit Certificates for IoT systems," in *Workshop on IoT challenges in Mobile and Industrial Systems*, (Florence), pp. 37–42, May 18, 2015.

[145] M. Campagna, "SEC 4: Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV)," tech. rep., Certicom Corporation, 2013.

[146] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "PAuthKey: A Pervasive Authentication Protocol and Key Establishment Scheme for Wireless Sensor Networks in Distributed IoT Applications," *International Journal of Distributed Sensor Networks*, vol. 10, no. 7, p. 357430, 2014.

[147] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, "Public Key Authentication and Key Agreement in IoT Devices With Minimal Airtime Consumption," *IEEE Embedded Systems Letters*, vol. 9, pp. 1–4, March 2017.

[148] O. Salman, S. Abdallah, I. H. Elhajj, A. Chehab, and A. Kayssi, "Identity-based authentication scheme for the Internet of Things," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, pp. 1109–1111, June 2016.

[149] J. Ni, X. Lin, and X. S. Shen, "Efficient and Secure Service-Oriented Authentication Supporting Network Slicing for 5G-Enabled IoT," *IEEE Journal on Selected Areas in Communications*, vol. 36, pp. 644–657, March 2018.

[150] D. Crockford, "The application/json Media Type for JavaScript Object Notation (JSON)." RFC 4627, July 2006.

[151] W3C, "Efficient XML Interchange (EXI) Format 1.0 (Second Edition)," 2014. `https://www.w3.org/TR/exi/`.

[152] F. Pacini, F.A., Aderohunmu, P. Pagano, A. Azzarà, M. Petracca, and S. Bocchino, "Performance Analysis of Data Serialization Formats in M2M Wireless Sensor Networks," in *European Conference on Wireless Sensor Networks*, (Porto), pp. 9–11, Feb. 2015.

[153] D. M and N. B. Biradar, "IOTA-Next Generation Block chain," *International Journal Of Engineering And Computer Science*, vol. 7, pp. 23823–23826, 04 2018.

[154] V. Atanasovski and L. Gavrilovska, "Efficient Service Discovery Schemes in Wireless Ad Hoc Networks Implementing Cross-Layer System Design," in *International Conference on Information Technology Interfaces*, (Cavtat), June 20–23, 2005.

[155] G. P. Halkes, A. Baggio, and K. G. Langendoen, "A Simulation Study of Integrated Service Discovery," in *Smart Sensing and Context* (P. Havinga, M. Lijding, N. Meratnia, and M. Wegdam, eds.), pp. 39–53, Springer Berlin Heidelberg, 2006.

[156] W3C, "OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)," 2012.
https://www.w3.org/TR/owl2-syntax/.

[157] B. A. A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 1617–1634, Third 2014.

[158] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 53, pp. 90–97, Feb 2015.

[159] A. Manzalini, R. Minerva, F. Callegati, W. Cerroni, and A. Campi, "Clouds of virtual machines in edge networks," *IEEE Communications Magazine*, vol. 51, pp. 63–70, July 2013.

[160] "FROG."
https://github.com/netgroup-polito/frog4.

[161] D. Soldani and A. Manzalini, "Horizon 2020 and Beyond: On the 5G Operating System for a True Digital Society," *IEEE Vehicular Technology Magazine*, vol. 10, pp. 32–42, March 2015.

[162] "Open Network Automation Platform."
http://www.onap.org/.

[163] AT&T White Paper, "Towards an Open, Disaggregated Network Operating System," 2017.
https://about.att.com/content/dam/innovationblogdocs/
att-routing-nos-open-architecture_FINAL%20whitepaper.pdf.

[164] D. C. Shoup, "Cruising for parking," *Transport Policy*, vol. 13, no. 6, pp. 479–486, 2006.

[165] J. N. Van Ommeren, D. Wentink, and P. Rietveld, "Empirical evidence on cruising for parking," *Transportation Research Part A: Policy and Practice*, vol. 46, no. 1, pp. 123–130, 2012.

[166] N. Zheng and N. Geroliminis, "Modeling and optimization of multimodal urban networks with limited parking and dynamic pricing," *Transportation Research Part B: Methodological*, vol. 83, pp. 36–58, 2016.

[167] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *Internet of Things Journal, IEEE*, vol. 1, no. 2, pp. 112–121, 2014.

[168] H. Liu, S. Chen, and N. Kubota, "Intelligent video systems and analytics: a survey," *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 3, pp. 1222–1233, 2013.

[169] M. Goldhammer, E. Strigel, D. Meissner, U. Brunsmann, K. Doll, and K. Dietmayer, "Cooperative multi sensor network for traffic safety applications at intersections," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pp. 1178–1183, IEEE, 2012.

[170] A. Redondi, M. Cesana, M. Tagliasacchi, I. Filippini, G. Dán, and V. Fodor, "Cooperative image analysis in visual sensor networks," *Ad Hoc Networks*, vol. 28, pp. 38–51, 2015.

[171] ICSI, "Intelligent Cooperative Sensing for Improved traffic efficiency." `http://www.ict-icsi.eu/`, 2016.
Last retrieved December 5, 2018.

[172] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K. K. R. Choo, and M. Dlodlo, "From cloud to fog computing: A review and a conceptual live vm migration framework," *IEEE Access*, vol. 5, pp. 8284–8300, 2017.

[173] S. Banerjee, P. Choudekar, and M. K. Muju, "Real time car parking system using image processing," in *2011 3rd International Conference on Electronics Computer Technology*, vol. 2, pp. 99–103, April 2011.

[174] H. Al-Kharusi and I. Al-Bahadly, "Intelligent parking management system based on image processing," *World Journal of Engineering and Technology*, vol. 2, pp. 55–67, 2014.

[175] D. Alessandrelli, A. Azzarà, M. Petracca, C. Nastasi, and P. Pagano, *ScanTraffic: Smart Camera Network for Traffic Information Collection*, pp. 196–211.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[176] L. Baroffio, L. Bondi, M. Cesana, A. E. Redondi, and M. Tagliasacchi, "A visual sensor network for parking lot occupancy detection in smart cities," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp. 745–750, Dec 2015.

[177] E. Eriksson, G. Dan, and V. Fodor, "Radio and computational resource management for fog computing enabled wireless camera networks," in *2016 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, Dec 2016.

[178] L. Mainetti, L. Patrono, M. L. Stefanizzi, and R. Vergallo, "A smart parking system based on iot protocols and emerging enabling technologies," in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pp. 764–769, IEEE, 2015.

[179] A. Bielsa, "Smart city project in santander to monitor parking free slots," *Dostopno na: http://www. libelium. com/smart_santander_parking_smart_city*, 2013.

[180] C.-C. Huang and S.-J. Wang, "A hierarchical bayesian generation framework for vacant parking space detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 12, pp. 1770–1785, 2010.

[181] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.

[182] M. Magrini, D. Moroni, C. Nastasi, P. Pagano, M. Petracca, G. Pieri, C. Salvadori, and O. Salvetti, "Visual sensor networks for infomobility," *Pattern Recognition and Image Analysis*, vol. 21, no. 1, pp. 20–29, 2011.

[183] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, pp. 246–252, IEEE, 1999.

[184] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground–background segmentation using codebook model," *Real-time imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[185] R. L. Sastre, P. G. Jimenez, F. J. Acevedo, and S. M. Bascon, "Computer algebra algorithms applied to computer vision in a parking management system," in *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, pp. 1675–1680, IEEE, 2007.

[186] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[187] V. Ojansivu and J. Heikkilä, "Blur insensitive texture classification using local phase quantization," in *International conference on image and signal processing*, pp. 236–243, Springer, 2008.

[188] P. R. De Almeida, L. S. Oliveira, A. S. Britto, E. J. Silva, and A. L. Koerich, "Pklot–a robust dataset for parking lot classification," *Expert Systems with Applications*, vol. 42, no. 11, pp. 4937–4949, 2015.

[189] V. N. Vapnik and V. Vapnik, *Statistical learning theory*, vol. 1.
Wiley New York, 1998.

[190] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, "Car parking occupancy detection using smart camera networks and deep learning," in *Computers and Communication (ISCC), 2016 IEEE Symposium on*, pp. 1212–1217, IEEE, 2016.

[191] G. Grassi, K. Jamieson, V. Bahl, and G. Pau, "Parkmaster: An in‚Äìvehicle, edge‚Äìbased video analytics service for detecting open parking spaces in urban environments," in *Proceedings of the 2nd ACM/IEEE Symposium on Edge Computing*, October 2017.

[192] X. Wang, X. Zheng, Q. Zhang, T. Wang, and D. Shen, "Crowdsourcing in its: The state of the work and the networking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1596–1605, 2016.

[193] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz, "Location-based crowdsourcing: extending crowdsourcing to the real world," in *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, pp. 13–22, ACM, 2010.

[194] A. Grazioli, M. Picone, F. Zanichelli, and M. Amoretti, "Collaborative mobile application and advanced services for smart parking," in *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, vol. 2, pp. 39–44, IEEE, 2013.

[195] R. P. Loce, E. A. Bernal, W. Wu, and R. Bala, "Computer vision in roadway transportation systems: a survey," *Journal of Electronic Imaging*, vol. 22, no. 4, pp. 041121–041121, 2013.

[196] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 3, pp. 920–939, 2011.

[197] L. Unzueta, M. Nieto, A. Cortés, J. Barandiaran, O. Otaegui, and P. Sánchez, "Adaptive multicue background subtraction for robust vehicle counting and classification," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 2, pp. 527–540, 2012.

[198] S. Messelodi, C. M. Modena, and M. Zanin, "A computer vision system for the detection and classification of vehicles at urban road intersections," *Pattern analysis and applications*, vol. 8, no. 1-2, pp. 17–31, 2005.

[199] A. Ottlik and H.-H. Nagel, "Initialization of model-based vehicle tracking in video sequences of inner-city intersections," *International Journal of Computer Vision*, vol. 80, no. 2, pp. 211–225, 2008.

[200] N. Saunier and T. Sayed, "A feature-based tracking algorithm for vehicles in intersections," in *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*, pp. 59–59, IEEE, 2006.

[201] Y.-L. Chen, B.-F. Wu, H.-Y. Huang, and C.-J. Fan, "A real-time vision system for nighttime vehicle detection and traffic surveillance," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 2030–2044, 2011.

[202] R. Rad and M. Jamzad, "Real time classification and tracking of multiple vehicles in highways," *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1597–1607, 2005.

[203] A.-N. Lai, H. Yoon, and G. Lee, "Robust background extraction scheme using histogram-wise for real-time tracking in urban traffic video," in *Computer and Information Technology, 2008. CIT 2008. 8th IEEE International Conference on*, pp. 845–850, IEEE, 2008.

[204] T. Semertzidis, K. Dimitropoulos, A. Koutsia, and N. Grammalidis, "Video sensor network for real-time traffic monitoring and surveillance," *IET intelligent transport systems*, vol. 4, no. 2, pp. 103–112, 2010.

[205] Z. Chen, T. Ellis, and S. A. Velastin, "Vehicle detection, tracking and classification in urban traffic," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pp. 951–956, IEEE, 2012.

[206] G. Wang, L. Tao, H. Di, X. Ye, and Y. Shi, "A scalable distributed architecture for intelligent vision system," *Industrial Informatics, IEEE Transactions on*, vol. 8, no. 1, pp. 91–99, 2012.

[207] M. Magrini, D. Moroni, G. Pieri, and O. Salvetti, *Intelligent Transport Systems: Technologies and Applications*, ch. Smart Cameras for ITS in Urban Environment, pp. 167–188. John Wiley & Sons, 2015.

[208] M. Magrini, D. Moroni, G. Palazzese, G. Pieri, G. Leone, and O. Salvetti, "Computer vision on embedded sensors for traffic flow monitoring," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pp. 161–166, IEEE, 2015.

[209] I. Chatzigiannakis, A. Vitaletti, and A. Pyrgelis, "A privacy-preserving smart parking system using an iot elliptic curve based security platform," *Computer Communications*, vol. 89, pp. 165–177, 2016.

[210] "SEED-EYE board." http://www.evidence.eu.com/products/seed-eye.html.

[211] D.-S. Lee, "Effective gaussian mixture learning for video background subtraction," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 5, pp. 827–832, 2005.

[212] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.

[213] A. Azzara, M. Petracca, and P. Pagano, "The ICSI M2M middleware for IoT-based intelligent transportation systems," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 155–160, IEEE, 2015.

[214] E. T. . 690, "Machine-to-Machine communications (M2M); Functional architecture," October 2013.

[215] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *IEEE International Conference on Local Computer Networks*, November 2004.

[216] N. Kushalnagar, G. Montenegro, C. Schumacher, *et al.*, "Ipv6 over low-power wireless personal area networks (6lowpans): overview, assumptions, problem statement, and goals," tech. rep., RFC 4919 (Informational), Internet Engineering Task Force, 2007.

[217] Z. Shelby and C. Bormann, *6LoWPAN: The wireless embedded Internet*, vol. 43. John Wiley & Sons, 2011.

[218] T. Winter, "Rpl: Ipv6 routing protocol for low-power and lossy networks," 2012.

[219] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (coap)," 2014.

[220] L. Richardson and S. Ruby, *RESTful web services*. " O'Reilly Media, Inc.", 2008.

[221] D. Hall, "PyMite: A Flyweight Python Interpreter for 8-bit Architectures," March 2003.

[222] F. Pacini, F. Aderohunmu, A. Azzarà, S. Bocchino, P. Pagano, and M. Petracca, "Performance analysis of data serialization formats in m2m wireless sensor networks," in *European Conference on Wireless Sensor Networks*, February 2015.

[223] D. Alessandrelli, M. Petracca, and P. Pagano, "T-res: Enabling reconfigurable in-network processing in IoT-based WSNs," in *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*, IEEE, 2013.

[224] Q. Wu, C. Huang, S. yu Wang, W. Chiu, and T. Chen, "Robust parking space detection considering inter-space correlation," in *IEEE international conference on multimedia and expo 2007*, pp. 659–662, IEEE, 2007.

[225] D. B. L. Bong, K. C. Ting, and K. C. Lai, "Integrated approach in the design of car park occupancy information system," *IAENG International Journal of Computer Science*, vol. 35, pp. 1–8, 2008.

[226] H. Ichihashi, A. Notsu, K. Honda, T. Katada, and M. Fujiyoshi, "Vacant parking space detector for outdoor parking lot by using surveillance camera and fcm classifier," in *Proceedings of the 18th International Conference on Fuzzy Systems*, FUZZ-IEEE'09, (Piscataway, NJ, USA), pp. 127–134, IEEE Press, 2009.

[227] E. Borgia, "The internet of things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, pp. 1 – 31, Dec. 2014.

[228] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart objects as building blocks for the internet of things," *IEEE Internet Computing*, vol. 14, pp. 44–51, Jan. 2010.

[229] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, pp. 22–32, Feb. 2014.

[230] M. Darianian and M. P. Michael, "Smart home mobile rfid-based internet-of-things systems and services," in *International Conference on Advanced Computer Theory and Engineering*, pp. 116–120, Dec. 2008.

[231] Z. Bi, L. D. Xu, and C. Wang, "Internet of things for enterprise systems of modern manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 10, pp. 1537–1546, May 2014.

[232] M. Yannuzzi, R. Milito, R. Serral-Gracià, D. Montero, and M. Nemirovsky, "Key ingredients in an iot recipe: Fog computing, cloud computing, and more fog computing," in *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks*, pp. 325–329, Dec. 2014.

[233] A. Manzalini, "Softwarization of telecommunications," *it - Information Technology*, vol. 57, pp. 321–329, Oct. 2015.

[234] A. P. Castellani, M. Gheda, N. Bui, M. Rossi, and M. Zorzi, "Web services for the internet of things through coap and exi," in *IEEE International Conference on Communications*, pp. 1–6, June 2011.

[235] S. Bocchino, S. Fedor, and M. Petracca, "Pyfuns: A python framework for ubiquitous networking sensors," in *European Conference on Wireless Sensor Networks*, pp. 1–18, Feb., 2015.

[236] T. Luo, H. P. Tan, and T. Q. S. Quek, "Sensor openflow: Enabling software-defined wireless sensor networks," *IEEE Communications Letters*, vol. 16, pp. 1896–1899, Nov. 2012.

[237] D. Alessandrelli, M. Petracca, and P. Pagano, "T-Res: enabling reconfigurable in-network processing in IoT-based WSNs," in *IEEE International Conference on Distributed Computing in Sensor Systems and Workshops*, pp. 337–344, May 2013.

[238] A. Azzarà, M. Petracca, and P. Pagano, "The icsi m2m middleware for iot-based intelligent transportation systems," in *IEEE International Conference on Intelligent Transportation Systems*, pp. 155–160, Sept. 2015.

[239] oneM2M, "Technical Specification: Reference Architecture," 2016.
Available at `http://www.onem2m.org/images/files/deliverables/Release2/TS-0001-%20Functional_Architecture-V2_10_0.pdf`.

[240] N. G. S. S.r.l., "SensOne board."
Available at `https://ngs-sensors.it/portfolio/sensone`.

[241] L. de Conception et d'Intégration des Systémes, "WiSMote platform."
Available at `http://wismote.org/doku.php`.

[242] dada's perl lab, "Computer Language Shootout Scorecard."
Available at `http://dada.perl.it/shootout/craps.html`.

[243] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *IEEE International Conference on Local Computer Networks*, pp. 455–462, Nov. 2004.

[244] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y.-L. Tian, A. Ekin, J. Connell, C. F. Shu, and M. Lu, "Enabling video privacy through computer vision," *Security & Privacy, IEEE*, vol. 3, pp. 50–57, May 2005.

[245] D. A. Fidaleo, H.-A. Nguyen, and M. Trivedi, "The networked sensor tapestry (nest): A privacy enhanced software architecture for interactive analysis of data in video-sensor networks," in *Proc. of the 2Nd ACM International Workshop on Video Surveillance &Amp; Sensor Networks*, VSSN '04, (New York, NY, USA), pp. 46–53, ACM, 2004.

[246] P. Korshunov and T. Ebrahimi, "Using Warping for Privacy Protection in Video Surveillance," in *Proc of the 18th International Conference on Digital Signal Processing (DSP)*, 2013.

[247] H. Sohn, W. De Neve, and Y. M. Ro, "Privacy protection in video surveillance systems: Analysis of subband-adaptive scrambling in jpeg xr," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, pp. 170–177, Feb 2011.

[248] S. M. Mukesh Saini, Pradeep Atrey and M. Kankanhalli, "W3-privacy: understanding what, when, and where inference channels in multi-camera surveillance video," *Multimedia Tools and Applications*, vol. 68, no. 1, pp. 135–158, 2014.

[249] S. M. Mukesh Saini, Pradeep Atrey and M. Kankanhalli, "Anonymous surveillance," in *Proc. of the IEEE International Conference on Multimedia and Expo (ICME 2011), Barcelona, Spain*, pp. 1–6, 2011.

[250] D. N. Serpanos and A. Papalambrou, "Security and privacy in distributed smart cameras," *Proceedings of the IEEE*, vol. 96, pp. 1678–1687, Oct 2008.

[251] T. Winkler and B. Rinner, "Security and privacy protection in visual sensor networks: A survey," *ACM Comput. Surv.*, vol. 47, pp. 2:1–2:42, May 2014.

[252] M. Colombo, A. Lazouski, F. Martinelli, and P. Mori, *A Proposal on Enhancing XACML with Continuous Usage Control Features*, pp. 133–146.
Boston, MA: Springer US, 2010.

[253] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *Information, Security and Cryptology, Proc. of the 12th International Conference (ICISC 2009), Seoul, Korea, December 2-4, 2009, Revised Selected Papers* (D. Lee and S. Hong, eds.), (Berlin, Heidelberg), pp. 229–244, Springer Berlin Heidelberg, 2010.

[254] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. of the 17th International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT'99),Prague, Czech Republic*, (Berlin, Heidelberg), pp. 223–238, Springer-Verlag, 1999.

[255] I. Damgård, M. Geisler, and M. Krøigaard, "Efficient and secure comparison for on-line auctions," in *Proc of the 12th Australasian Conference on Information Security and Privacy*, ACISP'07, (Berlin, Heidelberg), pp. 416–430, Springer-Verlag, 2007.

[256] I. Damgard, M. Geisler, and M. Kroigard, "A Correction to "Efficient and Secure Comparison for on-Line Auctions"," *Int. J. Appl. Cryptol.*, vol. 1, pp. 323–324, Aug. 2009.

[257] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich, "Scifi - a system for secure face identification," in *Provìc. of the 31st IEEE Symposium on Security and Privacy (S & P), California, USA*, pp. 239–254, May 2010.

[258] C.-S. L. Chao-Yung Hsu and S.-C. Pei, "Image feature extraction in encrypted domain with privacy-preserving sift," vol. 21, pp. 4593–4607, IEEE, Nov 2012.

[259] J. Wickramasuriya, M. Datt, S. Mehrotra, and N. Venkatasubramanian, "Privacy protecting data collection in media spaces," in *Proc. of the 12th Annual ACM International Conference on Multimedia (MULTIMEDIA '04), New York, USA*, (New York, NY, USA), pp. 48–55, ACM, 2004.

[260] P. Birnstill and A. Pretschner, "Enforcing privacy through usage-controlled video surveillance," in *Advanced Video and Signal Based Surveillance , Proc. of the 10th IEEE International Conference on (AVSS), Kraków, Poland*, pp. 318–323, Aug 2013.

[261] J. Park and R. Sandhu, "The $UCON_{ABC}$ usage control model," *ACM Transactions on Information and System Security*, vol. 7, pp. 128–174, 2004.

[262] X. Zhang, F. Parisi-Presicce, R. Sandhu, and J. Park, "Formal model and policy specification of usage control," *ACM Transactions on Information and System Security*, vol. 8, pp. 351–387, Nov. 2005.

[263] A. Pretschner, M. Hilty, and D. Basin, "Distributed usage control," *Commununcations of the ACM*, vol. 49, pp. 39–44, Sept. 2006.

[264] A. Lazouski, F. Martinelli, and P. Mori, "Usage control in computer security: A survey," *Computer Science Review*, vol. 4, no. 2, pp. 81 – 99, 2010.

[265] J. Park, X. Zhang, and R. Sandhu, "Attribute mutability in usage control," in *Research Directions in Data and Applications Security XVIII: Proc. of the 18th Annual Conference on Data and Applications Security (IFIP TC11/WG11.3), Catalonia, Spain* (C. Farkas and P. Samarati, eds.), (Boston, MA), pp. 15–29, Springer US, 2004.

[266] X. Zhang, M. Nakae, M. J. Covington, and R. Sandhu, "Toward a usage-based security framework for collaborative computing systems," *ACM Transactions on Information and System Security*, vol. 11, pp. 3:1–3:36, Feb. 2008.

[267] P. Kumari, A. Pretschner, J. Peschla, and J. Kuhn, "Distributed data usage control for web applications: a social network implementation," in *Proc. of the First ACM Conference on Data and Application Security and Privacy CODASPY 2011, San Antonio, TX, USA*, pp. 85–96, 2011.

[268] P. M. Aliaksandr Lazouski, Gaetano Mancini Fabio Martinelli, "Usage control in cloud systems," in *Proc. of The 7th International Conference for Internet Technology and Secured Transactions (ICITST-2012), Infonomics Society (2012), London, United Kingdom*, pp. 202–207, "December" 2012.

[269] A. Lazouski, G. Mancini, F. Martinelli, and P. Mori, "Architecture, workflows, and prototype for stateful data usage control in cloud," in *Proc. of the Security and Privacy Workshops (SPW 2014), San Jose, California*, pp. 23–30, May 2014.

[270] V. K. Singh and M. S. Kankanhalli, "Adversary aware surveillance systems," *IEEE Transactions on Information Forensics and Security*, vol. 4, pp. 552–563, Sept. 2009.

[271] OASIS, "eXtensible Access Control Markup Language (XACML) version 3.0," January 2013.

[272] R. Biswas and E. Ort, "The java persistence api-a simpler programming model for entity persistence," *Sun, May*, 2006.

[273] C. Andrew and C. Yao, "Protocols for secure computations," in *23rd IEEE Symposium on FOCS*, pp. 160 –164, 1982.

[274] A. Holzer, M. Franz, S. Katzenbeisser, and H. Veith, "Secure two-party computations in ansi c," in *Proc. of the 2012 ACM CCS*, pp. 772–783, 2012.

[275] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella, "Fairplay — a secure two-party computation system," in *Proc. of the 13th conference on USENIX Security Symposium*, pp. 20–20, 2004.

[276] A. Ben-David, N. Nisan, and B. Pinkas, "Fairplaymp: a system for secure multi-party computation," in *Proc. of the 15th ACM CCS*, pp. 257–266, 2008.

[277] G. Costantino, F. Martinelli, P. Santi, and D. Amoruso, "An implementation of secure two-party computation for smartphones with application to privacy-preserving interest-cast," in *Proc. of the 18th Mobicom*, pp. 447–450, 2012.

[278] Y. Huang, P. Chapman, and D. Evans, "Privacy-preserving applications on smartphones," in *Proc. of the 6th USENIX conference on HotSec'11*, 2011.

[279] S. Mascetti, C. Bettini, and D. Freni, "Longitude: Centralized privacy-preserving computation of users' proximity," in *Proc. of the 6th VLDB Workshop on SDM*, pp. 142–157, 2009.

[280] K. P. N. Puttaswamy, S. Wang, T. Steinbauer, D. Agrawal, A. E. Abbadi, C. Kruegel, and B. Y. Zhao, "Preserving location privacy in geosocial applications," *IEEE Transactions on Mobile Computing*, vol. 13, no. 1, pp. 159–173, 2014.

[281] J. Šeděnka and P. Gasti, "Privacy-preserving distance computation and proximity testing on earth, done right," pp. 99–110, 2014.

[282] "Getmewhere Android app download." `https://mega.nz/#!DoIw2YLS!MLiNO6apMmYaOQ-RXTajKk1dGvNAFTTqkC1ksdH2JxY`, 2016.

[283] "Service provider web-app." `http://secure.iit.cnr.it:8080/LocalizerSTCServer`, 2016.

[284] "Apple iOS notifications." https://developer.apple.com/notifications/, 2016.

[285] "Windows Phone notifications." https://msdn.microsoft.com/en-us/library/hh221549.aspx, 2016.

[286] "Google Cloud notifications." https://developers.google.com/cloud-messaging/, 2016.

[287] J. Soldatos, N. Kefalakis, M. Hauswirth, M. Serrano, J.-P. Calbimonte, M. Riahi, K. Aberer, P. P. Jayaraman, A. Zaslavsky, I. P. Žarko, L. Skorin-Kapov, and R. Herzog, "Openiot: Open source internet-of-things in the cloud," in *Interoperability and Open-Source Solutions for the Internet of Things* (I. Podnar Žarko, K. Pripužić, and M. Serrano, eds.), pp. 13–25, Springer International Publishing, 2015.

[288] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the internet of things: perspectives and challenges," *Wireless Networks*, vol. 20, Nov 2014.

[289] "ETSI TS 118 103 V2.4.1: Security Solutions." http://www.onem2m.org/images/files/deliverables/Release2/TS-0003_Security_Solutions-v2_4_1.pdf, 2016.

[290] P. Wouters, H. Tschofenig, J. Gilmore, S. Weiler, and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)." RFC 7250, June 2014.

[291] S. Farrell, R. Housley, and S. Turner, "An Internet Attribute Certificate Profile for Authorization." RFC 5755, Jan. 2010.

[292] "802.15.4-2011 - IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)." IEEE Standard 802.15.4-2011, 2011.

[293] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle, "6LoWPAN Fragmentation Attacks and Mitigation Mechanisms," in *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, (Budapest), pp. 55–66, Apr. 17–19, 2013.

[294] "Contiki: The Open Source OS for the Internet of Things." http://www.contiki-os.org/.

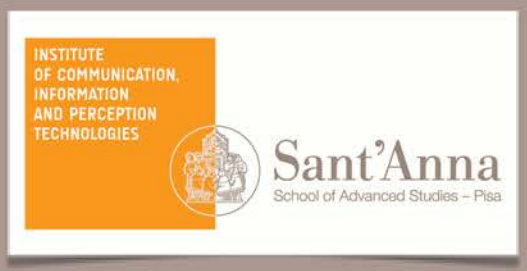
[295] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, " Transmission of IPv6 Packets over IEEE 802.15.4 Networks." RFC 4944, Sept. 2007.

[296] "tinydtls." https://projects.eclipse.org/projects/iot.tinydtls.

[297] "micro-ecc." https://github.com/kmackay/micro-ecc.

[298] "Californium." https://www.eclipse.org/californium/.

[299] "OpenSSL." https://www.openssl.org/.


INSTITUTE
OF COMMUNICATION,
INFORMATION
AND PERCEPTION
TECHNOLOGIES
Sant'Anna
School of Advanced Studies – Pisa