Università
degli Studi
di Ferrara

DOTTORATO DI RICERCA IN
Scienze dell’Ingegneria

CICLO 31

COORDINATRICE/COORDINATORE Prof. Trillo Stefano

Device as a Service and Fog Computing Middleware for the Internet of Things

Settore Scientifico Disciplinare ING-INF/05

Dottorando
Dott. Venanzi Riccardo

Tutore
Prof. Stefanelli Cesare

Cesare Stefanelli

Anni 2015/2018

Contents

List of Tables

Abstract

Nowadays every person is carrying in the everyday life at least a smart object. A smart object is a simple everyday thing equipped with computation, storage, communication, and sensing capabilities. These things have a very large and worldwide usage, so vast and pervasive as to be considered ordinary. In this context the concept of Internet of Things (IoT) has taken hold, and it has changed from a set of connected computer devices, into a set of connected surrounding things of the human's living space. Smart objects, or namely IoT objects, are wearables, sensors, actuators, smartphones, smart medical equipment etc. IoT enables the communication among these objects and towards internet, making those smart things able to see, hear, think and perform jobs by having them "talk" together, to share information and to coordinate decisions. The IoT transforms these objects from being traditional to smart by exploiting its underlying technologies such as ubiquitous and pervasive computing, embedded devices, communication technologies, sensor networks, internet protocols and applications. IoT devices produce huge amounts of raw data and that introduces vulnerabilities in many aspects, just to name a few, security, privacy, and sustainability/energy efficiency. Furthermore, this huge amount of data would need to be promptly processed, stored, manipulated, and managed. All these actions cannot be performed from the IoT devices themselves. This because IoT objects have only a limited capabilities and they are not able to carry those onerous operations. IoT-cloud integration enables fully-personalized on demand platforms that are able to promptly react to the various needs and configuration changes that an IoT application can have, in a pay-per-use manner. Furthermore, the integration of IoT with the Cloud cuts the capital and operational expenditures for IoT applications. Although the Cloud computing paradigm overcomes most of IoT application needs, low latency, geo-distribution, location-awareness and mobility support, to efficiently collect and promptly process the IoT data remain open challenges. As acknowledged by the ongoing research, the widely distribution of IoT datasources and the low latency needs lead in-cloud data processing to fail of meeting IoT requirements in timely decision-making process. Indeed, this two-layered architecture fails in meeting the IoT tight constraints of low latency, cost effectively, and timely decision-making process. To overcome the above mentioned issues another quite novel networking paradigm can be used, fog computing. Fog computing has a distributed architecture targeting application and services with widely spread deployment analogous to the IoT. Fog computing is positioned as an intermediate layer between cloud computing infrastructure and IoT devices. Thus, fog nodes bridge application objects running in the Cloud and the edge. The benefits of fog computing is the support of the IoT environments with computing resources, communications protocol, location awareness, mobility support, low latency, geo-distribution, and enhanced Quality of Experience (QoE). Physically, fog nodes are industrial network routers, smart mobile access points, smart switches, deployed into the environments of interest; such as smart residential or business buildings, shopping centers, smart urban areas and so on. As a complementing concept to the cloud, fog computing has been

identified as a possible solution to ensure energy efficiency at the IoT devices and to overcome the main other issues. IoT is not a mature paradigm yet, and it has a plethora of open issues and very challenging future developments that fog paradigm can help to address. These open issues and challenges are drawing the attention of the research communities, those one that have most relevance are:

- Device Heterogeneity;
- Device Staticity;
- Object Availability;
- Network Scalability;
- Node Data Rawness.

There are many type of objects in our lives and in the market and, potentially, each of them might, carry a different type of hardware, communicate with a different protocol, have different amount of resources, etc. It is clear that a kind of standardization or homogenization is needed in order to be able to use and communicate with any type of IoT object. The IoT paradigm is very mobile, the nodes have an high dinamicity and they suddenly join or leave the network. Here, emerges the need of properly managing the objects network, the node connection loss, new smart node discovery, and the device data migration from a network to another. Directly related to device mobility is device availability. In IoT a device should always be capable of providing measurement information, because several actions might depend by it. It is clear that, a strategy of fault tolerance and device recovery is needed.

Under this preamble, this research project introduces the novel concept of Device as a Service (DaaS). The research effectively faces the above technical challenges and aims to raise the devices from physical layer to a higher level, making them integrated and available to the cloud. The basic idea is to virtualize sensors/actuators, more generally, IoT nodes, by making them available as a service to the Cloud. This virtualization will create an abstraction of the physical sensor. This abstraction will be available for Cloud applications and services, via advanced mechanisms, algorithms, protocols, and strategies for aggregation and virtualization. The project intends to solve the currently most challenging hot topics in the IoT scenarios.

Nodes Homogeneity Device homogenization cloud side through the sensors virtualization;

Advanced Smart Device Discovery Solutions Sensor's mobility introduced with new discovery protocol approach;

Device Sustainability The IoT node power efficiency is provided by fog middleware;

Complex Computation and Query on Smart Sensor Data Cloud side, the virtualized sensor data are made available for querying through Semantic Web.

First of all, this research introduces the definition of Device as a Service, and its all possible configurations as a services available to the cloud. As it is already stated, the two layered IoT-Cloud architecture fails in meeting IoT need in many scenarios. More precisely, when for example, an IoT application needs of low latency, or very quick decision-making process, the cloud is unable to timely satisfy the IoT application constraints. This research introduces a middle layer between IoT and cloud in order to leverage these problems. This research proposed a three layered architecture by interposing a fog middleware between IoT and cloud. Furthermore, this research project exploits a fog middleware to perform a smart IoT device node discovery improving the device discoverability and the power efficiency. This discovery protocol exploits the main features and characteristics provided by fog middleware such as location awareness, mobility support, and geo-distribution, in order to optimize the devices' discoverability of the traditional discovery protocols and reduce the power consumption, by increasing the node sustainability at the same time. The novel proposed protocols are smart and fog middleware driven. These protocols, indeed, exploit the support given by fog nodes to enhance the discoverability and the power efficiency of the IoT discovery protocols. More precisely, the proposed architecture and protocols will be implemented in a traditional Bluetooth Low Energy (BLE) discovery scenario. This scenario, normally composed by two entities, scanner, and advertiser, will be enriched with a third entity, the fog node. The fog node will alert the scanner via Message Queuing Telemetry Transport (MQTT) message when an advertiser is nearby. The alerting message sent by fog node, makes the scanner aware that a new advertiser is nearby, enabling the discovery process. With this novel fog middleware driven approach, the discovery process is activated only when a scanner and an advertiser are effectively in their proximity, increasing the device discoverability and nodes sustainability.

In addition, a new approach of querying data at cloud level through Semantic Web and its standard de-facto query language SPARQL will be presented. This research uses this mechanism to make virtual sensor abstract data available and accessible to end-user application and/or other services at cloud level. Finally this thesis presents and shows the feasibility of using DaaS in a telco application by exploiting the concept of Network Function Virtualization (NFV) in possible and future scenarios. In the last chapter will be presented a full use case of the vertical architecture cloud-fog-IoT. This use case is part of big European project powered by Artemis, named Arrowhead.

This dissertation concludes at chapter five with the conclusion that summarized the achieved results and the future directions of work.

Chapter 1

IoT and Fog Computing

In this chapter will be introduced the context in which this research project is placed. Furthermore this chapter will give a large overview on the two main computing paradigm this research addressed. Finally a briefly presentation of the key concepts this research project proposed.

1.1 IoT: Internet of Things

In the recent few past years, the concept of Internet has changed from a set of connected computer devices into a set of connected surrounding things of the human's living space. These things, namely smart objects, are everyday life physical things equipped with computation, storage, communication, and sensing capabilities, i.e. wearables, sensors, actuators, and even smartphones. The communication among these smart objects is referred as Internet of Things (IoT). The IoT enables physical objects to see, hear, think and perform jobs by having them "talk" together, to share information and to coordinate decisions. The IoT transforms these objects from being traditional to smart by exploiting its underlying technologies such as ubiquitous and pervasive computing, embedded devices, communication technologies, sensor networks, Internet protocols and applications [1]. Already in 2010, the number of Internet connected objects had surpassed the earth's human population, and the trend is still growing [2]. According with the US National Intelligence council has stated, by 2025 the IoT will connect everything in our life [3]. IoT has been defined as new paradigm embracing all the wireless communication technologies such as wireless sensor network (WSN), and mobile notwork (MN). The term IoT was initially proposed to refer to uniquely identifiable interoperable connected objects with radio-frequency identification (RFID) technology [4]. Later on, researchers relate IoT with more technologies such as sensors, actuators, GPS devices, and mobile devices. Today, a commonly accepted definition for IoT is *a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual 'Things' have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network* [5]. The basic idea of this concept is the pervasive presence around us of a variety of things or objects – such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, mobile phones, etc. – which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals [3]. These new paradigm has paved the way toward ubiquitous and pervasive IoT services and applications [6]. Digital-health, smart utilities, environmental monitoring, smart-buildings/homes and public safety are just a few examples of IoT applications. [7, 8, 9].

1.1.1 IoT Applications

IoT environments and applications are playing a remarkable role in improving the quality of our lives. According to [10], IoT applications can be spitted in two macro categories, real-time reactive applications, and ambient data collection and analytics. Furthermore, inside these two categories, the IoT applications cover several domains. In fig 1.1 is depicted some IoT application domains.

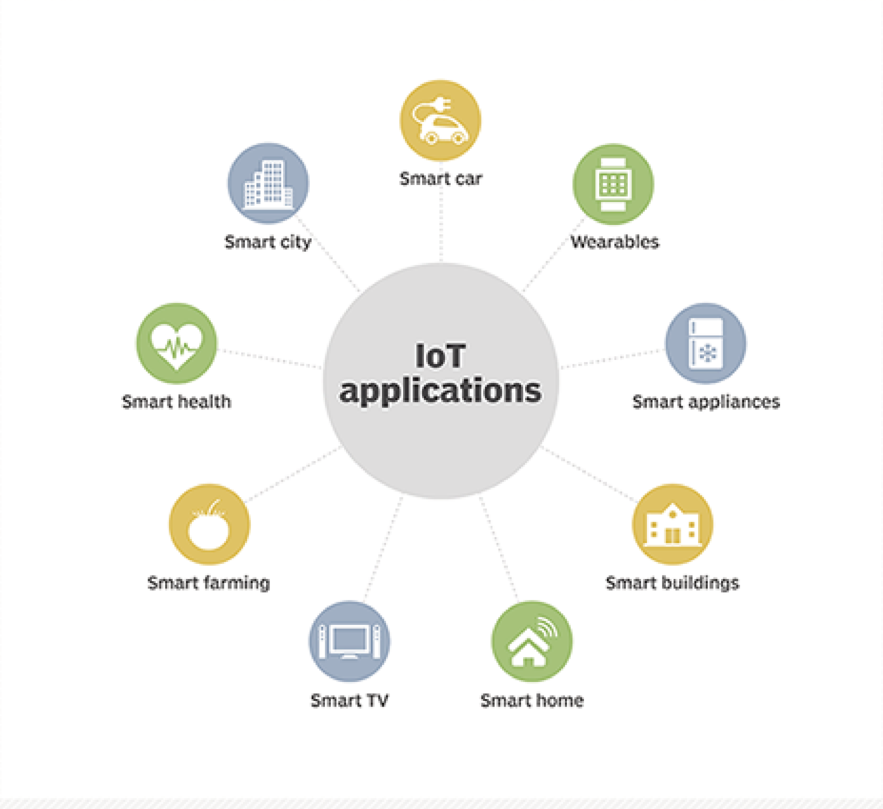


Figure 1.1: IoT Application Domains. Credits to "https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT"
.

Inside this plethora of domains, the most remarkable are:

- Smart healthcare;
- Smart Environments;
- Logistic and Transportation;
- Social Applications

In the ambient collection and analytics category falls all the applications that generally involve collecting sensor data from a variety of sensors in order to monitor a specific situation or to keep tracking the trends of particular activities. Then these data are processed offline to infer a model, and finally, run the model as a predictor for new data collected from the sensor in the future. An application for tracking user's sport activity improvements or a system for monitoring the environment pollution at home or along a user's everyday path [11], are two examples of real data collection and analytics applications. The real time reactive applications category groups all those applications that involves real-time reactive systems such as autonomous vehicle or manufacturing processes where the systems make real-time decisions based on observed sensor values. Another example of application belonging to this category might be a smart health application that monitors a patient life's values and promptly alerts the nearest medical center as soon as any of these values exceeds the regular threshold. The domains listed above are the most targeted by IoT applications, they address the most relevant spheres of our lives. In the transportation domain, cars, public transport, bikes , and even roads, can be equipped with sensors, actuators, and processing power. IoT applied to transport and logistic can help to send important information to traffic control sites and transportation vehicles to better route the traffic, help in the management of the depots, provide the tourist with appropriate transportation information, and monitor the status of the transported goods. The logistic field can also benefit of IoT application support; it is possible to realize a real-time monitoring application in order to follow the good along every step of the supply chain, just to name an example. Many more benefits are provided by IoT technologies to healthcare, real time life's signs monitoring application, patient-flow monitoring to improve workflow in hospitals, real time diagnosis, emergencies detection, medication regiment prediction and many more. The applications of IoT technologies that can have the biggest impact on the people everyday life, fall in Smart Environments domain. All these applications aim to make our lives more comfortable applying sensors, actuators and intelligence to our homes, office spaces, buildings, and leisure environments. Rooms heating can be turned ON/OFF or adjust according to weather or to our preferences remotely, the switching ON/OFF of the lights and their brightness can be adapted according to external luminosity, personal garden's irrigation can be monitored and triggered remotely, museums and gym can provide real time information to their visitors, just to name some example. Finally, the Social domain groups all those applications that enable the user to interact with other people to maintain and build social relationships. Indeed, things may automatically trigger the transmission of messages to friends to allow them to know what we are doing or what we have done in the past, such as moving from/to our house/office, travelling, meeting some common mates or playing soccer [4, 12]. Fig. 1.2 shows the projected market share of dominant IoT applications [13].

IoT is having a huge impact on the market in several sectors, and its projections is growing year by year. According with Gartner, IoT market value by 2020 will be $ 1.9 trillion [15]. While the whole annual economic impact caused by the IoT is estimated to be in range of $ 2.7 trillion to $ 6.2 trillion by 2025 [13].

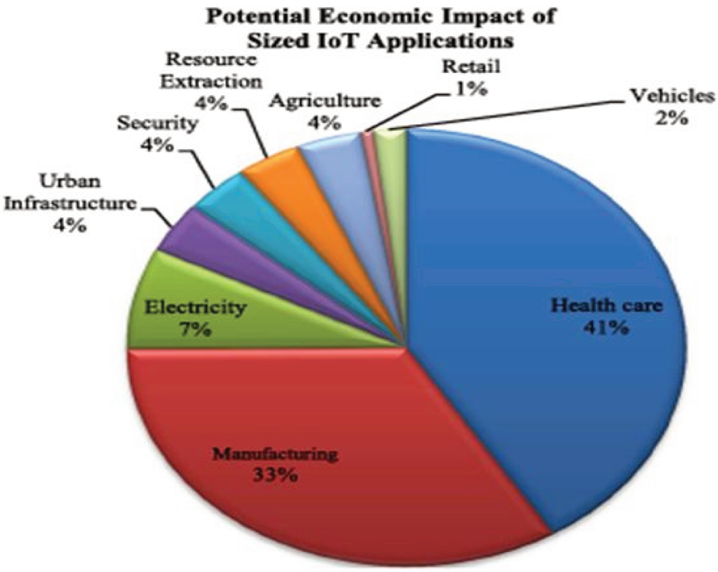


Figure 1.2: Projected Market Share of Dominant IoT Application by 2025. Credits to [14]
.

1.1.2 IoT Architecture

IoT is able to connect a vast number of different devices, to fulfill this purpose a flexible layered architecture has a crucial importance. Through the years, with the ever growing adoption of IoT technologies, many IoT architectures has been proposed. Despite the many architectures proposed, a common-accepted IoT reference architecture has not been reached yet [16]. One of the first architecture proposed is a coarse-grained three-layered one, composed by a Perception, Network and Application layer. According with this architecture, the Perception layer is accounted of detecting and recognizing the things or objects in the environment. In addition, It also is responsible for the sensing and acting actions. Indeed, this layer, contains all the sensors and the actuators. The Network layer is the core of this type of architecture, it is accounted of collecting, aggregating and transmitting the information from/to the other layers. All the actions of data manipulation are performed by the Network layer. It contains the software and hardware instrumentation of internet network in addition to the management and information centers. The last component, the Application layer, is a macro-layer containing all the applications and services that exploit the IoT data. The purpose of this layer is to perform the IOT applications billing, management and authentication operations. With the wide-spreading of IoT usage, new needs have emerged and more layered finer-grained type of architecture has become necessary. Therefore five layer architectures have been proposed. In figure 1.3. are depicted some example of IoT architectures.

As it is possible to see from fig. 1.3, the architectures have the most of layers in common and differ from each other only in few of them. Among the all architectures, and despite the different naming of the tiers, five layers are the most well recognized and wide spread.

The first layer from the bottom is the Object layer. This layer totally reflects the Perception layer of the three layer architecture model. This tier represents the physical sensors of the IoT that aim to collect and process information. This layer includes sensors and actu-

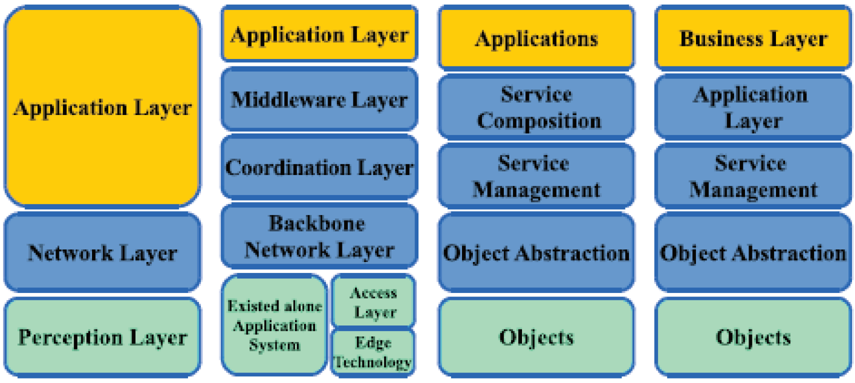


Figure 1.3: IoT Architecture examples

.

ators to perform different actions such as fetching location, temperature, weight, motion, vibration, acceleration, humidity, etc. The Object layer digitizes and send data to the next tier, the Object Abstraction layer, through secure channels. The big data created by the IoT are gathered from the real world by this layer.

Object Abstraction layer sends data provided by the Objects layer to the Service Management layer through secure channels. Data can be transferred through various technologies such as 3G, 4G, WiFi, Bluetooth Low Energy, infrared, ZigBee, etc. In addition, other functions like data management processes are handled at this layer [17].

Service Management layer works as a middleware between the objects world and the traditional computer paradigms. This layer is accounted of providing IoT services to the requester, this pairs is based on addresses and names. At this layer the IoT data is no more hardware specific and this allows the application developers to work with heterogeneous objects regardless from where that data is fetched or which type of communication protocol has been used to deliver it. Furthermore, Service Management layer processes received data, makes decisions, aggregates and delivers the required services over the network wire protocols [18].

The Application layer provides the services requested by users. Also, the Application layer, provides the IoT data with semantics to the final user. Indeed, this layer, enriches the IoT data from the previous layer with information that give to those data a proper meaning for the user. For instance, this layer provides temperature, air humidity and brightness measurements to the user who asks for information about its own smart home. This layer has a crucial importance for the IoT, it is able to provide high-quality smart services to meet users' needs. In the Application layer falls all the IoT application vertical domains described in the previous subsection, such as smart building/home, transportation, smart healthcare, industrial automation, social, etc [19].

The top tier of the IoT five layers architecture is Business layer. It basically manages the overall IoT system activities and services. indeed it is also accounted of orchestrating different IoT services or applications in order to create more complex one. The Business layer, also named Management layer, is capable to build business model, graphs, flowcharts, etc. based on the data provided by the previous layers. Business layer, being the top tier, has the visibility of all the IoT data and information from lower layer of the stack. Therefore, Business layer is accounted of analyzing, monitoring and evaluating actions, data and performance of the IoT system elements. This tier is also named Management tier because it can perform IoT service orchestration, and provide support to decision-making processes based on Big Data analysis. Moreover, this layer compares the output of each layer with the expected output to enhance services and maintain users' privacy.

The three layer architectures described above, borrows its layers and concepts from network stacks, but this kind structure does not cover all the real IoT environments and it is tailored for a specific types. Indeed, for example, the Network layer of this type of architecture, does not cover all underlying technologies that transfer data to an IoT platform. In addition, a coarse-grained architecture leads to have more duties and responsibilities on each single layer. It brings the whole system to be unbalanced, moreover because the lower layers are supposed to be run on resource-constrained devices. For example, all the actions and processes performed by Service composition layer would consume the most of the time and energy of the device to communicate with other devices and integrate the required services. On the other hand, a five layer architecture is more suited for covering the needs and constrains of all IoT environments. In this architecture, the Application layer is the entry point by which end-users can interact with a device and query for interesting data. In addition, it makes Business layer capable to produce high-level analysis and report, by providing an interface. The control mechanisms of accessing data in the application layer are also handled at this layer. This layer is hosted on powerful machine due to its complex and enormous computational needs, i.e in the Cloud. The five-layer architecture is the most applicable model for IoT applications.

1.1.3 IoT Open Issues and Challenges

Despite its widespread adoption and its continuously growing supporting community, IoT is not considered at mature level yet [20]. Indeed, IoT paradigm, still presents several open issues and very hot challenges to research community. The main open issues fall in two big category, Technological challenges, and Security and Privacy. Also, future efforts are needed to address these challenges and examine the characteristics of different industries to ensure a good fit of IoT devices in the industrial environments.

Technical and Technological Challenges

IoT include an incredibly huge number of different nodes, each of which produces an enormous quantity of raw data. In addition, this data would need to be accessed from everywhere regardless the underling technologies. IoT is a very complicated heterogeneous network, which includes the connection between various types of networks through various communication technologies. Currently, there is lack of widely accepted common platform that hides the heterogeneity of underlining networks/communication technologies and provides a transparent naming service to various applications [21]. Large amounts of data transmission across the network at the same time can also cause frequent delay, conflict, and communication issues. Indeed, with the huge number of things connected to the Internet, a massive amount of real-time data flow will be automatically produced by connected things [22]. IoT is typically developed based on a traditional ICT environment, hence another remarkable effort would be required to integrate IoT with already existing IT systems or legacy systems into a unified information infrastructure. Design a Service Oriented Architectures (SOA) for IoT is not trivial, indeed, service-based things might suffer from performance and cost limitations. In addition, scalability issues often arise as more and more physical objects are connected to the network. When the number of things is large, scalability is problematic at different levels including data transfer and networking, data processing and management, and service provisioning. From the viewpoint of service, a lack of a commonly accepted service description language makes the service development and integration of resources of physical objects into value-added services difficult. The developed services could be incompatible with different communication and implementation environments. In addition, powerful service discovery methods and object naming services need to be developed to spread the IoT technology [21], [23]. According with what is stated above, the crucial technical and technological IoT open issues, are summarized as following:

Device heterogeneity: There are many types of objects on the market, they differ each others by the hardware they carry on (quantity of sensors embedded, device's intelligence, etc.), the level of OSI stack they can support, the communication technology they can adopt and many others.

Device staticity: The thing's mobility is currently barely handled in IoT networks, indeed, an object moving among different IoT networks, is not properly managed without connection loss. This topic really opens a large amount of unexplored and very hot fields, such as IoT nodes discoverability, device networks hand-off managing, sensor's portability, network elasticity, data quality etc.

Objects Availability: Often the objects in IoT network could be unavailable, or for a networking problem or for a device fault; for the time being a fault tolerance and/or recovery is a big open issue for the IoT scenarios.

Network Scalability: With the increasing of number of the devices connected to the network a single router device could be no longer enough to manage the massive amount of connected devices and all the communications these require. In other words, when the number of devices boosts up, they can saturate the network resources, and this represents a big ceiling, moreover with the continuously growing of IoT diffusion.

Node Data Rawness: Devices are a big source of raw data that need to be managed in order to optimize the connection with the Cloud, the battery life and the quality of these data. The manipulation of data could cover a very large scope from security and privacy up to data awareness. The majority of these scenario aren't well explored, analyzed and/or standardized yet.

Security and Privacy

The acceptance and widespread of new IoT technologies and services will largely rely on the information security and data privacy protection, which are two difficult issues in IoT

[24]. People use to resist the IoT as long as there is no public confidence that it will not cause serious threats to privacy. IoT allows many daily things to be tracked, monitored, and connected, and a lot of personal and private information can be collected automatically. IoT is extremely vulnerable to attacks, the majority of the communications are wireless, which makes eavesdropping extremely simple. Moreover, IoT components are characterized by low capabilities in terms of both energy and computing resources and thus, they cannot implement complex schemes supporting security. In addition, authentication and data integrity also represents two crucial weak points for IoT. Some existing technologies are available for consumer use, but are not suitable for industrial applications that have strict safety and security requirements. To secure the information, existing encryption technology borrowed from the WSNs or other networks need to be carefully reviewed, when they are used to build IoT. Protecting privacy in the IoT environment becomes more serious than the traditional ICT environment because the number of attack vectors on IoT entities is apparently much larger [25], [26]. For example, in a smart health application, a health monitor continuously fetches patient's data, such as heart rate and blood sugar level and then delivers the information directly to the hospital or the doctor's office over the network. When the information is transferred over the network, patient's personal data might be stolen or compromised. Another example is a smart home where smart sensors can be use to monitor the occupancy, to turn ON/OFF the lights accordingly with the external brightness, to turn ON/OFF the heating, etc. The sensors in a smart home can produce sensible information regarding the people living in the house. This information, if stolen or altered, can be a serious problem. If altered a completely outsider might take the control of our home. In addition, the theft of a such personal information is a severe privacy violation. This information can be used by a thief to know when nobody is inside. It should be noticed that some issues, such as the definition of privacy and legal interpretation are still vague and are not clearly defined in IoT. Although the existing network security technologies provide a basis for privacy and security in IoT, more work still needs to be done. A reliable security protection mechanism for IoT needs to be researched from the following aspects:

- the definition of security and privacy from the viewpoint of social, legal, and culture
- trust and reputation mechanism
- communication security such as end-to-end encryption
- privacy of communication and user data
- security on services and applications.

Accordingly with what has been stated in these two subsections, IoT is clearly not at a mature level and it still presents several open issues and hot challenges to the research community. From the aforementioned highlighted problems and open issues clearly emerge the necessity of a middleware or middle tier to bridge the IoT world and open Internet-Cloud.

1.2 Fog Computing Paradigm

IoT devices produce huge amounts of raw data and that introduces vulnerabilities in many aspects, just to name a few, security, privacy, and sustainability/energy efficiency [27]. IoT-cloud integration enables fully-personalized on demand platforms that are able to promptly react to the various needs and configuration changes that an IoT application can have, in a pay-per-use manner. Furthermore, the integration of IoT with the Cloud cuts the capital and operational expenditures for IoT applications [28]. Although the Cloud computing paradigm overcomes most of IoT application needs, low latency, geo-distribution, location-awareness and mobility support in order to efficiently collect and promptly process the IoT data remain open challenges [29]. Cloud computing suffers from substantial yet unsolved challenges such as large end-to-end delay, traffic congestion, processing of massive amount of data, and communication cost. Some of these issues are caused mainly due to large physical distance between cloud service provider's Data Centers (DCs) [30]. As acknowledged by the ongoing research, the widely distribution of IoT datasources and the low latency needs lead in-cloud data processing to fail of meeting IoT requirements in timely decision-making process [31, 28]. To overcome these issues fog computing is defined as an extension to cloud computing that brings Cloud facilities close to the edge, hence to IoT data sources, as well as the end users. Cisco defines the concept of fog computing as a bridge between the IoT devices and large-scale cloud computing and storage services [31]. The term "*fog*" is used simply because "*fog is a cloud close to ground*", [32], i.e., *F*rom c*O*re to ed*G*e computing enabling refined and better applications or services. Fog computing is a highly virtualized platform that provides computing, storage, and networking services between end user and DC of the traditional cloud computing [33].

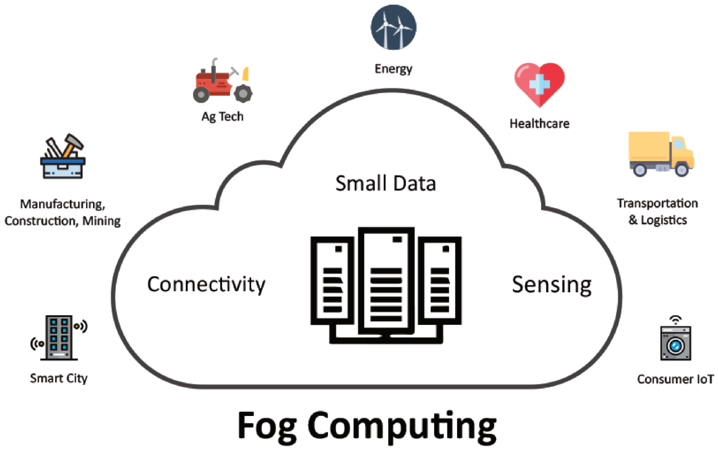
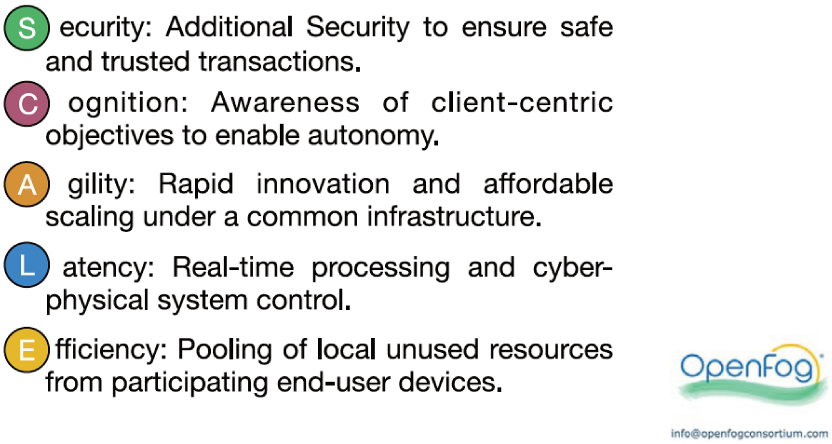


Figure 1.4: Fog and IoT. Credits to [34]

.

Fog computing has a distributed architecture targeting application and services with widely spread deployment analogous to the IoT [29]. Fog computing is positioned as an intermediate layer between Cloud computing infrastructure and IoT devices. Thus, fog nodes bridge application objects running in the Cloud and the edge. The benefits of fog computing is the support of the IoT environments with computing resources, communications protocol, location awareness, mobility support, low latency, geo-distribution, and enhanced Quality of Experience (QoE) [35]. OpenFog consortium [36], defines the main capabilities of fog computing as SCALE, an acronym stands for Security, Cognition, Agility, Latency and Efficiency, as depicted in fig 1.5.



Ⓢecurity: Additional Security to ensure safe and trusted transactions.

Ⓒognition: Awareness of client-centric objectives to enable autonomy.

Ⓐgility: Rapid innovation and affordable scaling under a common infrastructure.

Ⓛatency: Real-time processing and cyber-physical system control.

Ⓔfficiency: Pooling of local unused resources from participating end-user devices.

Figure 1.5: OpenFog: SCALE characteristics

.

Fog computing has many characteristics targeting the IoT support, the most widespread accepted and recognized are [30, 32]:

- Low Latency and Location Awareness
- Supports geographic distribution
- End Device Mobility
- Capacity of Processing High Number of Nodes
- Wireless Access
- Real-Time Application
- Heterogeneity

Physically, fog nodes are industrial network routers, smart mobile access points, smart switches cabable of providing resources for services at the edge of the network [37]. Fog nodes are typically deployed into the environments of interest such as smart residential or business buildings, shopping centers, smart urban areas and so on [38, 39]. As a complementing concept to the cloud, fog computing has been identified as a possible solution to overcome the most of the IoT still open issues and to improve the energy efficiency of the IoT devices [40, 41]. In the next subsection a plethora of more formal definitions of fog computing are given.

1.2.1 Fog Definition

The first definition of fog computing was given by CISCO, that initiates it as an extension of the cloud computing paradigm from the core of network to the edge of the network.

In the perspective of Cisco, It is a highly virtualized platform that provides computation, storage, and networking services between end devices and traditional cloud servers [32]. According to Cisco [42], the Fog computing paradigm provides an ideal place to analyze most data near the devices that produce and act on that data instantaneously. The Fog is located near things that are able to process and act on the data generated. The devices that are within the Fog environment are known as fog devices or fog nodes. Fog nodes can be resource-poor devices such as set-top-boxes, access points, routers, switches, base stations, and end devices, or resource-rich machines such as Cloudlet and IOx. Cloudlet is a resource-rich computer like "cloud in a box", which is available for use by nearby mobile devices. Satyanarayanan et al. [43] build Cloudlet, which is ahead of fog computing but coincides the concept of fog computing. IOx is a fog device product from Cisco, whose architecture is shown in fig. 1.6, works by hosting applications in a Guest Operating System (GOS) running in a hypervisor directly on the Connected Grid Router (CGR)" [44, 45]. On IOx platform, developers can run python scripts, compile their own code, and even replace the operation system with their own.

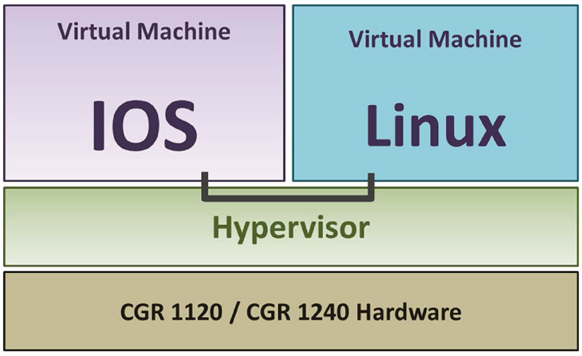


Figure 1.6: IOx architecture

.

Cisco Systems is not the only one to have provided a definition of fog computing. Vaquero and Rodero-Merino has defined this paradigm as "a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralized devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third parties. These tasks can be for supporting basic network functions or new services and applications that run in a sandboxed environment. Users leasing part of their devices to host these services get incentives for doing so" [46]. Several consortium and importan company has address the emerging fog computing topic. For example, IBM has define this paradigm as "the term Fog computing or Edge Computing means that rather than hosting and working from a centralized cloud, Fog systems operate on network ends. It is a term for placing some processes and resources at the edge of the cloud, instead of establishing channels for cloud storage and utilization" [47], while the OpenFon consortium has stated that "fog is a system-level horizontal architecture that distributes resources and services of computing, storage, control and networking anywhere along the continuum from cloud to Things" [48]. According to Yi et al. in [44], the definition given by Vaquero and Rodero-Merino is debatable and a definition that can distinguish clearly between Fog computing and

other related computing paradigms is still required. The definition given by IBM represents Edge and Fog computing as the same computing paradigm. While, from the Shi et al. point of view, in [49], fog computing focuses more on the infrastructure side while edge computing focuses more on the things' side. Furthermore, Edge computing is not spontaneously associated with any cloud-based services such as SaaS, IaaS, and PaaS [50]. On the other hand, the definitions given by Cisco, OpenFog consortium, Bononi, and many others, see the fog computing as a paradigm main focused on supporting IoT and end users' devices.
In the next subsection an overview of the fog computing architecture is provided.

1.2.2 Fog Computing Architecture

Many types of fog computing architectures has been proposed in literature, especially since fog is a not-trivial extension of cloud towards IoT world. Furthermore, fog is not in a mature stage as cloud yet, hence all the architectures proposed do not converge in a standard one. Among the several architectures proposed the most common used, widely accepted and adopted is three tiers model, depicted in fig. 1.7.

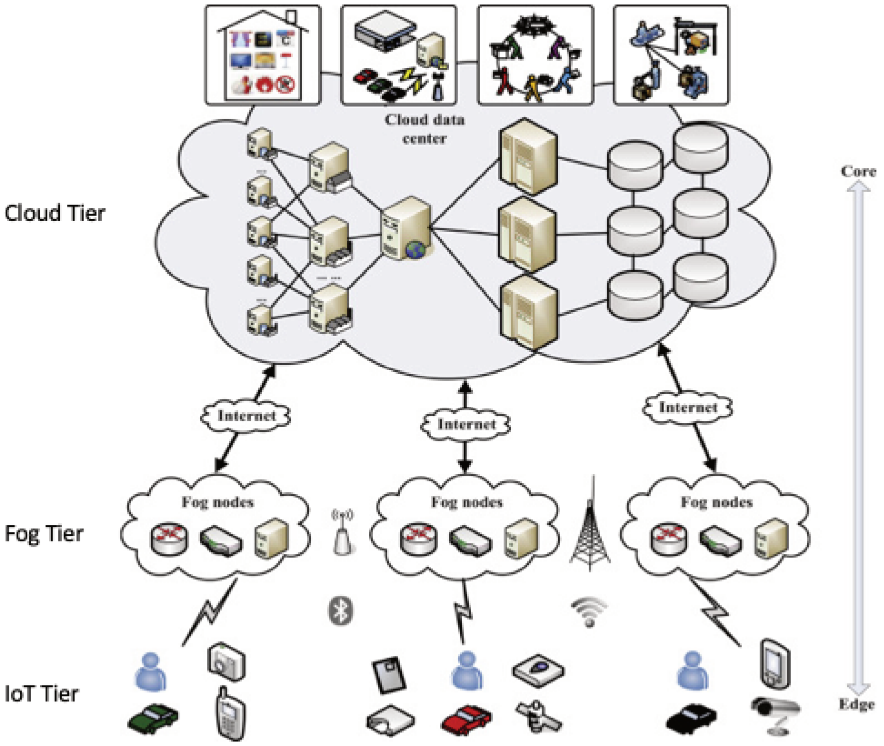


Figure 1.7: Three leyered fog computing architecture.

The three layers are, from top to the bottom, Cloud Tier, Fog Tier, and IoT Objects/Things Tier. These layers are described below [51, 30]:

Cloud Tier The uppermost tier, commonly known as, cloud computing tier. Traditional cloud servers and cloud DC reside in the top-most tier. This tier has sufficient storage and computing resources

Fog Tier The Fog Tier, or the middle layer, is also known as the fog computing layer. The primary components of this tier are intelligent intermediate devices such as a router, gateway, switch, and Access Points (APs). These component are named fog nodes and possess the ability of data storage, computation, routing, and packet forwarding. These fog nodes can also collaboratively share storage and computing facilities.

IoT Objects/Things Tier This is the bottom-most tier of the architecture. This layer consists of IoT enabled objects including sensor nodes, EU's smart devices such as smartphones, tablets, smart cards, smart vehicles, smartwatch, and others. These IoT nodes are often termed as Terminal Nodes (TNs), they sense heterogeneous physical parameters and transmit the same to the immediate upper tier.

The principle of fog computing architecture is based on bridging IoT world with cloud computing bringing the cloud capabilities from the core to the edge. The IoT smart objects of the IoT Tier are assumed to be grouped in a location-based logical clusters. These logical clusters are named virtual clusters (VCs). The VCs together form an edge virtual private network (EVPN) that transmits data to multiple fog instances (FIs). An FI is conceptualized specific to a geographic location. The mobility of a IoT objects makes the mapping of a thing to an FI flexible and non-static. While the data are transmitted upwards (towards the fog tier) they are processed within the intermediate fog devices. The fog devices can stretch from different networking components, such as routers, switches, gateways, and access points to high-end proxy servers and computing machines. As stated by Bonomi et al. in [52], the fog computing architecture can be classified into two sub-parts:

- fog abstraction layer
- fog orchestration layer

While the first manages the fog resources, enables virtualization, and preserves tenant-privacy, the second one accounts the exclusive fog properties. The fog orchestration layer comprises of a small software agent—foglet which monitors the state of the devices, a distributed database to account for scalability and fault tolerance, and a service orchestration module which is responsible for policy-based routing of application requests. Within the FIs, the data are processed and analyzed to decide whether it needs to be transmitted to the cloud DCs. Application requests which require storage or historical data based analytics are redirected to the cloud, else, the data are processed within the fog units. The fog devices possess limited semi-permanent storage that allow temporary data storage and serve the latency-sensitive applications in real-time. The cloud computing tier is commonly responsible for permanent storage of huge, voluminous data chunks within its powerful DCs. The DCs are equipped with massive computational ability. However, unlike conventional cloud architecture, the core cloud DCs are not bombarded for every single query. Fog computing enables the cloud tier to be accessed and utilized in an efficient and controlled manner [51]. Another example of fog architecture has been proposed in [33]. This architecture has six layer, and it is structures as following, from top to the bottom:

Transport Layer Sends data to the cloud

Security Layer Handles security related issues

Temporary Storage Layer Stores the data temporally

Preprocessing Layer Perform data filtering and trimming

Monitoring Layer Handles services requests and energy consumption issues

Physical and Virtualization Layer Contains IoT smart objects and virtual sensors

description
Physical and virtualization layer mainly contains physical IoT nodes and virtual sensor nodes. The Monitoring layer handles the requested task and observes the energy consumption issues of underlying physical devices. Data management related tasks such as data filtering and data trimming are performed in Preprocessing layer. The Temporary storage layer stores the data for a limited time only. The security-related issues are handled in the Security layer. Finally, the Transport layer is responsible for sending data to the Cloud [30].

1.2.3 Issues and open challenges of Fog

Fog, being a not mature technology yet, presents several open issues and challenges that attract the research community focus. The main challenges and open issues of fog computing are presented as follow.

Fog Networking

As it is already stated, fog is located close to the edge, and this drives it to be heterogeneous. Fog network aims to connect every fog element. Nonetheless, handling such a network, maintaining connectivity and providing services upon that, especially in a large scale scenario as IoT, is not trivial. In order to create an easy to maintain and flexible network environment, two emerging techniques has been proposed, software-defined networking (SDN) and network function virtualization (NFV). The adoption of SDN and NFV can improve fog computing under many aspects. NFV and SDN can increase the network scalability, make easier the the management and the implementation, and reduce costs. The improvements due to SDN and NFV benefit the resource allocation, VM migration, traffic monitoring, application-aware control and programmable interfaces.

SDN "When SDN concept is implemented with physically (not just logically) centralized control, it resembles the fog computing concepts, with fog device acting as the centralized controller." [53]. In the fog, each node should be able to act as a router for nearby nodes and resilient to node mobility and churn, which means controller can also be put on the end nodes in fog network. The challenges of integrating SDN into fog network is to accommodate dynamic conditions as mobility and unreliable wireless link.

NFV NFV replaces the network functions with VM instances. Since the key enabler of fog computing is virtualization and those VMs can be dynamically created, destroyed and offloaded, NFV will benefit fog computing in many aspects by virtualizing gateways, switches, load balancers, firewalls and intrusion detection devices and placing those instances on fog nodes [54].

Although NFV and SDN brings a remarkable advantages, there still are unsolved problems with these two emerging techniques. Most of the researches in SDN address how to proper design distributed SDN system that meet the harsh requirement of fog computing such as latency, scalability and mobility. For NFV in fog computing, the performance of virtualized network appliances is still the first concern [55]. This problem has two aspects: one is the throughput or latency of virtualized network appliances (middlebox) in fog network, and the other is how to achieve efficient instantiation, placement and migration of virtual appliances in a dynamic network, together to meet low latency and high throughput requirements.

Application Offloading

By computation and storage offloading facilities, fog computing benefits the applications that are not effective while running them on resource constraint EU's devices. However, in certain instances, application offloading is not always efficient regarding the delay, bandwidth, and energy consumption due to lack of available resources in fog layer [56, 57]. Thus, before offloading any application, it is equally important to find the parameters that have an impact on the offloading performance. In addition, it would be useful if we can accurately predict the performance of the offloaded applications. If the predicted performance does not gain any significant advantages, then offloading to the fog is not efficient.

Resource Management

Compared to cloud computing, the fog computing does not enough computing and storage resources. Thus, efficient resource allocation is an important research issue in fog computing. Due to dynamic nature, the IoT devices and fog nodes leave (join) from (to) a fog layer arbitrarily. Thus, the dynamic load balancing and task assignment to mobile end device become very difficult due to the scalability issue in fog layer. Therefore, how to balance the load towards cost-effective concerning delay, power consumption, and bandwidth resource distribution is one of the open research directions. Since fog computing is localized, understanding the mobility pattern of end devices may be helpful for task assignment and resource management in fog computing.

Heterogeneity

As the bottom-most layer in fog computing architecture consists of various end devices such as a smartphone, smartwatch, virtual sensor node, intelligent devices including autonomous car, smart home devices, the heterogeneity issue arises regarding data collection, data format, and data processing capability. Nevertheless, fog node in fog cluster comprises by routers, switches, gateway and other devices with different computing and communication facilities. Thus heterogeneity becomes an important design factor in fog computing architecture. Handling of different data formats and various communication protocols for managing semi- or unstructured data becomes major issues.

Standardization

The working group of OpenFog consortium is working towards standardization of fog computing architecture in terms of communication; security, testbeds, manageability, and software Infrastructure. Standardization is required so that *various IoT systems can securely interact each other and cloud service*. This also enables computing, networking, and storage across multiple edges, fog, and end devices.

Billing and Incentives

To obtain high QoS, such as low service latency, sufficient storage, high bandwidth, and fast computation for the service sometimes require a cost in terms of money. Nevertheless, pay-as-you-go seems to be an essential aspect in terms of incentives for the users who share their computing and storage resources [58]. Proper cost management and how to properly bid the computing and storage price towards a ‘trustful’ collaboration are worthwhile to be studied in fog computing.

1.2.4 Differences between Fog and Cloud Computing

Although fog and cloud are strictly related in terms of providing resources, and application support, they differ in several aspects. As it is already stated, fog computing extends cloud computation, storage, communication and networking capabilities to the edge, the IoT world. At this scope, fog has a distributed architectures and approach based local geographical orchestration, while, on the other hand, cloud is more centralized and based on DCs [59]. This lead the IoT operations with the fog to have lower cost, lower power consumption end extremely lower latency than with cloud. On the other hand, cloud offers more stability, reliability and fault tolerance. Indeed, fog computing has an higher failure rate due to its decentralized managements, wireless connectivity, and power failures [60, 61]. Fog computing, thanks to its vicinity to the IoT, allows almost real-time interaction. This type of interactions are not possible with the cloud due to its very high distance and latency. The distance between IoT and fog is estimated to be one or two hops in wireless connectivity, while the distance between IoT and cloud surely requires multi-hops in wireless and wired connectivity [59, 62, 63]. On the other hand, the large distance allows the cloud to provide a global resource optimization and to have an overall view of the underlying layers and application, while fog organizes and manages the only the local environments. Fog is definitely suited for supporting, enhancing and improving the efficiency, performances, and the critical aspects of cyber physical systems [64]. In the table 1.1, are summarized the main technical differences between fog and cloud

Table 1.1: Main Differences between Fog and Cloud Computing

Features	**Cloud**	**Fog**
Computing Model	Centralized	Distributed, close to the edge
Deployment Cost	High	low
Resources optimization/managing	Global/Centralized	Local/ Distributed and Centralized
Fault Tolerance	High	Basic
Latency	High	very Low
Reliability	High	Low
Computation Cost and Capacity	High	Low
Storage Capacity	High	Low
Node Mobility and Mobility Management	Very Low/Easy	High/Hard
Maintenance	Expensive	Cheap
Power Consumption	High	Low
Power Source	Direct Power	Battery/Direct Power/Green Energy
Distance from IoT	Multi Hops	Few Hops
Real Time Application	Very Difficult	Easy Possible

Finally, fog and cloud are two different paradigms tailored for two different purposes. They have different characteristics and one cannot replace the other, but they can work perfectly together. Both contribute to fulfill distinct goals.

1.3 Building a Fog-Oriented Middleware for the IoT

As it is already emerged in the previous section, fog paradigm is particularly suited for overcoming the IoT issues. This project aims to solve the above addressed IoT open challenges with a particular focus on the aspects of: device heterogeneity, IoT objects mobility, and data quality. The project intends to solve the currently most challenging hot topics in the IoT scenarios by presenting an hardware/software middleware infrastructure that exploits fog paradigm and its key features. Among the all still open issues of IoT, the proposed middleware aims to fulfill the most challenging ones, according to the research community. The key features the proposed middleware target to provide are listed below:

Device Homogeneity: Device homogenization the Cloud by devices' virtualization

Advanced Discovery Solutions: Object's mobility introduced with new and smart device discovery protocol approaches

Node Network Elasticity: With device mobility the sensor network is able to re-size itself on demand

Complex Computation on Smart Sensor Data: Exploiting all points above, it is possible to make complex data computations.

This project is based on the idea of effectively dealing with the above technical challenges by raising the device from physical layer to a higher level through the middleware. In this way the proposed infrastructure would be able to make the physical devices integrated and available to the Cloud. The basic idea, on which the project is based, is to virtualize the IoT devices, by making them available as a service for Cloud applications and services. The proposed solution to achieve this goal, based its core on the introduction of **smart discovery protocol** cabable of overcoming the device heterogeneity. In addition, a proper **ad-hoc middleware architecture** based on fog computing principle provides support for the discovery and make the new devices available to be queried at cloud level as virtual device. The designing of this architecture and the development of such heterogeneous device discovery protocol, are the core of this project. They are the effective core elements for realizing and enabling the concept of virtual sensor. As first step, I start from the valuable seminal work described in [65] and plan to work on significantly extending its scope on other hard technical challenges, such as re/shaping of the sensors network (Elastic WSN), efficiently virtual sensors provisioning and aggregated data manipulation. My primary idea is to virtualize sensors by creating an abstraction, or virtual device between the physical devices and the resources seen by the Cloud; over the virtual device will be totally separated from a physical implementation or even a single real device, thus reducing and managing the associated heterogeneity issues. Like the already well known cloud paradigms Infrastructure, Platform and Software as a Service (PaaS, IaaS, SaaS), the cloudification of the devices would introduce the innovative concept of DaaS - Device as a Service. In addition to homogeneity via virtualization, DaaS allows many other interesting and disruptive features, for instance a virtual sensor could abstract a set of physical devices acting as data manipulator/handler toward the Cloud. A virtual sensor could represent an entire physical sensor network into the Cloud, creating, in this way, a cloud sensors network of networks; all of this seen as a single virtual device from the Cloud side. The DaaS concept has the potential to become very relevant for industrial developments as well, because all existing IoT applications and realizations are isolated and based on a vertical domain, hence based on a single technology. With sensors virtualization, or rather, with the devices seen as a service, the application will no longer see the physical layer (physical devices) and they will loose the tight binding with a specific hardware or communication technology. For instance, if a user application, made to monitor a zigBee sensors network, can just work over a zigBee network, using that specific communication technology; an application (or service) interacting with a virtual sensors network, it is completely uncoupled from which communication technology the sensors use to communicate each others or how the data is provided to the Cloud. The DaaS concept can introduce other interesting benefits as the constant sensor availability: the device cloud abstraction allows the applications to be completely hardware independent, this makes the IoT application automatically hardware fault tolerant. A virtual sensor can continue to work even if a physical sensor is faulty by simply switching the real device the DaaS support transparently takes data from. The continuous service uptime also allows a transparent sensor's recoverability; the final user does not realize anything about what happens on the device side of the system's architecture, that turns out totally transparent to the application.

1.3.1 QoD: Quality of Device

Along with the introduction of virtual sensor, another related concept has been introduced, the Quality of Device(QoD). This new term is stemmed from DaaS, where QoD includes all the aspects regarding Quality of Service for virtual sensor. DaaS aims to guarantee the following list of QoD aspects:

Availability: Virtual Sensor guarantee the continuous data availability through the possibility of gathering data from several real devices. When an object becomes unavailable, for example because the battery is discharged or it leaves the the network, the virtual sensor replaces the sensor with another one in the network. A Virtual Sensor can obtain the same data type of unavailable sensor by deriving and manipulating other sensors data type.

Scalability: DaaS can resize the physical sensors network in case of application load peaks. For instance, a virtual sensor can enlarge the corresponding physical sensors network under an incoming load peak and reduces it when the load disappears.

Recoverability: Similarly to the availability, A Virtual sensor dynamically can switch data source when the real devices is faulted. DaaS can recognize a faulted device from one that has left the network, and it alerts the final user.

QoD is a combination of much more other aspects, it also include user experience, sampling cost and all the aspects regarding the Quality of Service (QoS). The cloud side device virtualization requires the introduction of high dynamicity on the sensors network side, more specifically, there's the need to be able to add devices to the sensors network on demand.

1.3.2 Discovery Protocol

As it is already introduced before, discovery protocol is one of the two core part of this project. An efficient and smart discovery protocol has a crucila relevance in order to meet the heterogeneity and the battery life constrains belonging to IoT. furthermore, the device mobility support does not exist in every IoT scenario or application/realization and, where it is provided, it is typically not fully managed. Normally it is bound to a vertical domain and technology, and it is not inter-operable at all. The key for filling this gap is the introduction of a new device discovery protocol. A discovery protocol in IoT scenario has to deal with connection devices typically battery powered and with limited memory and computational power, for this reason the way in which a new device approaches to a sensor network and discovers others nearby devices, assumes a main role, and must be dynamic. Another important aspects to face is the heterogeneity of the embedded equipment carried by devices. For this reason an approach based on REST and Coap over HTTP like that one proposed in [66], in some cases could not result as the best solution. This project aims to provide an advanced and fully-horizontal discovery protocols able to add any type of IoT device to the network. The proposed protocol targets the adoption of a totally new fog-integrated and smart location-driven approach. Such smart protocol has to adopt a multi-layer approach in order to overcome the heterogeneity of devices at IoT layer.

Following a multi-layered approach for discovery, the proposed protocol can discover any type of smart object regardless its type, the hardware is carries and the communication technology it uses. There are many project aspects that a new discovery protocol has to consider; according with [67], discovery protocols in ad-hoc systems can be mainly divided in Request-Based Discovery and Direct Discovery. The first one includes all the protocols which attempt to discover the nearby devices by broadcasting a discover request, most common protocol included in this class are Bluetooth, Wi-Fi Direct and IrDA. On the other hand, Direct Discovery category includes all the schemes where a device periodically transmits its discovery signal in order to advertise its presence or listens for others' devices discovery signal (every certain period of time); FlashLinQ and Wi-Fi ad-hoc are both part of this class. All the literature are pushing toward the using a Device-To-Device (D2D) communications and Proximity-Based discovery protocol, even more in new LTE technology [68]. The introduction of mobility in IoT sensors network field requires a quite groundbreakingly new approach for the discovery protocols, a more efficient and smarter one. A smart location-driven fog initiated protocol is what this project is going to introduce. This protocol is based on the following steps:

1. the smart device (for instance a smartphone) periodically pings the fog with its position
2. the fog node alerts the in-range device closest network gateway
3. the gateway initiates the direct communication
4. the device establishes the connection and advertises the device
5. the device became part of the network

With a higher degree of details, we intend to propose a hybrid protocol based on D2D communication following the Direct Discovery pattern, optimized through fog interaction so to enable the beacon only in the network's neighborhood, according with the device position/direction. When the fog will alert the WSN that a new device approaches, the gateway awakes the closest device (closest to the approaching one) that switches the beacon mode on, it acts as a gateway, and tries to establish a direct connection. Once the device is connected to the network, it will expose its type of device and the services offered. The protocol is inspired and then extended and improved, by multi-hop cellular device discovery for LTE application mentioned in [67] for the direct connection, from [69] and [68] for energy saving and proximity principle.

Chapter 2

Related Works

After having introduced the context in which this research project has been placed and the goals it aimed to achieve, this section provides a large overview on the state of the art and similar researches.

2.1 Virtual Sensors

The large wide spreading adoption of sensors and, more in general, the IoT, have paved the way among research communities that the concept of virtual sensor might be the main actor in addressing the hot open challenges and issues of IoT. The virtual sensor concept is very large scoped and among the several research fields, it has found a quite different adoption. For example in [70], the authors have been moved by the same motivation of mine, and they have proposed the concept of virtual sensor, but they proposed a different architecture and scopes. While, in [71], the focus has been led on creating a network of virtual sensor rather than to direct provide the abstraction on physical real sensor at cloud level for its service or application exploitation. This application of virtual sensor network (vsn) has been thought to primary target the mobility issues that affects the traditional sensors network. The authors in [72], define and describe a new paradigm of computation for wireless sensor networks, the sensor cloud, which decouples the network owner and the user and allows multiple WSN to interoperate at the same time for a single or multiple applications that are transparent to users. The paper provide a first sketch of the principle addressed in this research project defining sensor cloud as a heterogeneous computing environment spread in a wide geographical area that brings together multiple WSN s consisting of different sensors. Each WSN can have a different owner. The sensor cloud then virtualizes the wireless sensors and provides sensing as a service to users. Because users buy sensing services on demand from the sensor cloud, use of large-scale sensor networks becomes affordable with ease of use. The widely diffusion and adoption of sensors has also opened the door to new type of network, moreover with the emerging of IoT. Body Sensor Networks (BSNs) represent an emerging technology which has received much attention recently due to its enormous potential to enable remote, real-time, continuous and non-invasive monitoring of people in health-care, entertainment, fitness, sport, social interaction. This paper [73], introduces multi-layer task model based on the concept of virtual sensors to improve architecture modularity and design reusability. The concept of virtual sensor, in this paper, is defined as abstraction of components of BSN systems that include sensor sampling and processing tasks and provide data upon external requests. The virtual sensor model implementation relies on SPINE2, an open source domain-specific framework that is designed to support distributed sensing operations and signal processing for wireless sensor networks and enables code reusability, efficiency, and application interoperability. The proposed model is applied in the context of gait analysis through wearable sensors. As it is already stated the virtual sensor concept has a very large scope and it can be applied to many fields. One of them, vehicular network and application result to be particularly rich of virtual sensor's application. In [74], where the virtual sensor have been used as observer to monitor the sideslip angles of cars with three different levels of speed. The authors were particularly concerned with the stability of observers and models when the vehicle approaches the linear dynamic limits. Results for three different sets of sensors: yaw rate; vehicle speed; yaw rate and vehicle speed together are presented in their paper. Results concerning observability have been also included in their research work. Another remarkable application in the vehicular field is presented in [75]. This work resulted to be a patent, the virtual device is used to monitor the NO_x emission of a target engine. The virtual sensor is exploited to predict NOx values based on a model reflecting a predetermined relationship between control parameters and NO_x emissions, wherein the control parameters include ambient humidity, manifold pressure, manifold temperature, fuel rate, and engine speed associated with the engine. Another patent has been granted by US government to the application of virtual sensor robot navigation fields. In fact, in [76], the virtual sensor concept has been used for improving the navigation system of a mobile robot. The navigation method using a virtual sensor includes:

- generating information on positions of obstacles, the information which is estimated to be generated by virtual sensors that are virtually present, based on information on positions of the obstacles, which is generated by physical sensors;
- controlling a movement of a robot according to the information on the positions of the obstacles.

An application of virtual sensor in smart traffic field is presented in [77]. Where, more precisely, virtual sensors have been used to enable fault detection and isolation flight control system. The virtual sensor, in this case, is an virtual acceleration sensor used in the flight control system of a small commercial aircraft. Another application of virtual sensor in navigation system is presented in [78]. The authors, here, have adopted a virtual sensor for abstracting several sonar sensors and and odometric information, with the goal of improving the robot navigation system. More precisely, the virtual sensor has the main objective of detecting new room space in a indoor environment. Among the other IoT field benefiting of the virtual sensor concept is smart manufacturing. In [79], a predictive Fuzzy ARTMAP neural system and two hybrid networks, each combining a dynamic unsupervised classifier with a different kind of supervised mechanism, were applied to develop virtual sensor systems capable of inferring the properties of manufactured products from real process variables. A new method to construct dynamically the unsupervised layer was developed. A sensitivity analysis was carried out by means of self-organizing maps to select the most relevant process features and to reduce the number of input variables into the model. The IoT application field that, perhaps, benefits the most of the virtual sensor concept is smart environment. Smart environment is a big field in which home, building, factory, farm, more generally any environment equipped with IoT devices (sensors/actuators), falls into this category. In this field virtual sensor can abstract the several types of physical real sensor by addressing, in this way, all the mobility and heterogeneity issues of a typical and traditional IoT smart environment. Context information is the key to producing self-adaptive applications in IoT smart environments. However, the supporting infrastructure that generates context information can be made dynamically responsive to the environment. Therefore, for the development of self-adaptive applications, it is necessary to demonstrate that valid context information can be created by virtual sensors instead of physical sensors. In [80], the authors present a context aware simulation system called CASS. In particular, it generates the context information associated with virtual sensors and virtual devices in a smart home domain. By using CASS, the self-adaptive application developer can immediately detect rule conflict in context information and determine optimal sensors and devices in a smart home. Another simulation system for smart home is presented in [81], here the authors present a solution for physical sensor deployment in a real scenario based on virtual sensor. Virtual sensors in smart environment have not been just used for developing simulation systems. In [82], the smart home environment presented, adopts the virtual sensors to obtain information from the Internet, local network or local computer. The use of virtual sensors broadens the scale of context that is available in the environment and allows to develop more complex applications. The virtual sensors developed in this research project, aim to obtain music listening preferences by parsing iTunes music databases, news from online RSS feeds, concert timetables, concert tickets, TV listings, and stock quote information. Furthermore, the authors have also developed a virtual weather sensor to illustrate the ease in which Internet data can be incorporated into a smart home application. This sensor uses web-scraping techniques to read weather data that is published online. The using of virtual sensor in this context has brought many advantages. For example, the virtual weather forecast sensor, makes available a weather service without the need of having a physical personal weather forecasting infrastructure installed on the roof. It also directly provide the final weather information to the final user, avoiding the end user to be able to read the raw weather data provided by a physical weather infrastructure. It is ever more evident, even from the cited examples, that there is the necessity of a IoT middleware. This middleware would leverages the principal issues of IoT, such as the following:

Device Heterogeneity: There are many types of sensors on the market, they differ each others by the hardware they carry on (quantity of sensors embedded, device's intelligence, etc.), the level of OSI stack they support, the communication technology they adopt and many others.

Device Staticity: The sensor's mobility is not currently handled in IoT networks, in other words a sensor cannot move to another networks without connection loss. This topic really opens a large amount of unexplored and very hot fields, such as IoT sensors discoverability, sensor networks hand-off managing, sensor's portability, data security and privacy, network elasticity, data quality etc.

Device Availability: Often the devices in IoT network could be unavailable, or for a networking problem or for a device fault; for the time being a fault tolerance and/or recovery is a big open issue for the IoT scenarios.

Network Scalability: With the increasing of number of the devices connected to the network a single router device could be no longer enough to manage the massive amount of connected devices and all the communications these require. In other words, when the number of devices boost up, they can saturate the network resources, and this represents a big ceiling, moreover with the continuously growing of IoT diffusion.

Sensor Data Rawness: Sensors are a big source of raw data that need to be managed in order to optimize the connection with the Cloud, the battery life and the quality of these data. The manipulation of data could cover a very large scope from security and privacy up to data awareness. The majority of these scenario aren't well explored, analyzed and/or standardized yet.

Some solutions above cited present a first approach to virtual sensor concept, but in order to properly address the issues above mentioned, this concept might not be enough. The introduction of an IoT middleware that supports and fills the lack of using only the virtual sensor concept. In [83], a study of the role and the importance of a middleware for IoT. In that research work the authors introduce the middleware for IoT as an entity that acts as a bond joining the heterogeneous domains of applications communicating over heterogeneous interfaces. Furthermore, it is roughly introduced the concept of using an ontology to describe the abstract virtual sensor entities as a virtual node. Another remarkable research work on IoT middleware is SOCAM. It is the acronym of Service-Oriented Context-Aware Middleware and it is presented in [84]. SOCAM is defined as a middleware architecture for IoT for the building and rapid prototyping of context-aware services. This middleware aims to provide efficient support for acquiring, discovering, interpreting and accessing various IoT contexts to build context-aware services. This middleware also sketches the concept of virtual sensor to gather data from physical sensor not directly connected to the middleware, e.g. web service, or third-party device. A first outline of virtual sensor architecture along with a middleware in presented in [85]. This architecture sketch, although it is only an outline, is similar to that one adopted in this research work. It presents a three layered architecture in which the second layer has the purpose of virtualize the physical sensors and resources in the cloud. According to the authors, this virtualization is then used for enabling the provisioning of cloud-based sensor services and other IT resources remotely to the end-user without being worried about the sensors exact locations. The authors, in this architecture, have proposed the creation of virtual sensor based on the utilization of a service template. In their architecture, the service templates are prepared by the service providers as service catalog, and this catalog enables the creation of service instances automatically that are accessed by multiple users.

2.2 Fog for IoT and Discovery Protocol

As it is already stated, IoT scenarios produce huge amounts of raw data and that introduces vulnerabilities in many aspects, among the others the most relevant are objects mobility, device availability, nodes sustainability/energy efficiency, security, adn privacy [27]. The integration IoT-cloud via fog middleware enables fully-supported on demand platforms that are able to promptly overcome to the various needs and configuration changes the many IoT scenarios can have. Furthermore, the integration of IoT with the Cloud cuts the capital and operational expenditures for IoT applications [28]. Due to the frequent sensing and actuation, high volume of data in-transit, and lightweight analysis on the data, sustainability problem in IoT devices has appeared as a grand challenge [86]. Fog computing and its distributed middleware architecture has been proposed as solution to overcome the above mentioned IoT issues. By its definition fog computing brings computational resources, and storage resources from the cloud to the IoT devices. Fog middleware also provides location awareness, mobility support, low latency, geo-distribution, to IoT scenarios. In this doctoral research projects fog middleware has given a crucial support for IoT by primary overcoming device mobility, object heterogeneity, location awareness, node availability, and, maybe one of the most important, device sustainability and power efficiency. In this research project, I leveraged the possibility of easily integrating short and medium range wireless connectivity in the same IoT devices to propose novel cross-network management operations to overcome those typical limitations of IoT node discovery. The power efficiency issues is crucial in a scenario where the totality of the nodes run on battery. The fog middleware can also address this type of issues by providing support for IoT by making the traditional node discovery protocol, smart [40, 41]. This research project has targeted the presentation of an architecture and two protocols aiming to improve the traditional IoT node discovery and achieve a better power management. The architecture exploits Internet connectivity and fog paradigm to enhance the conventional discovery process. The proposed protocols is technology agnostic and can by applied on each scanning/advertise paradigm. The protocols are an enhanced node discovery solution specifically tailored for IoT-Fog environments to ensure sustainability/energy efficiency and discoverability, as well as reliability. The fog layer entity (fog node) bridges the IoT devices and the cloud and provides needed context awareness about nodes in the locality. They only require that the IoT nodes would be capable of internet connection. The protocols are not technology dependent because one of the main goal of the proposed fog middleware is to overcome device heterogeneity. The protocols developed and adopted in this project are application-layer protocol, in this way, working at high level, they are not prone to the typical technology related problem. Moreover, with the fog middleware support, they are not onerous for the IoT nodes themselves.

In the research the protocols have been implemented Bluetooth Low Energy technology. IoT devices at the edge are considered to be a Bluetooth Low Energy Scanner (BLE-S) and a Bluetooth Low Energy Advertiser (BLE-A) in the subscriber and publisher roles, respectively and the fog node keeps track of the trajectory of the BLE-A, and implements a signalling scheme to control the Bluetooth interface of the BLE-S depending on the geo-location of the BLE-A which is communicated through WiFi interface. The signaling scheme allows to synchronize the advertisement and scanning frames leading to 100% discoverability of the devices, and remarkable savings in the BLE, CPU, and per-application battery consumption. BLE has been chosen as reference technologies because, it has become the main technologies to connect smart objects to the IoT [87]. For this reasons, more and more researcher are focusing their effort in studying the BLE discovery phase, and its power efficiency. The power efficiency of the BLE discovery process is still an open challenge, and it can be faced in several ways. For example, in [88], the authors proposed a method for improving the BLE device discovery by tuning the BLE advertisement's interval. Hence in that article the authors focused their effort on the advertiser's settings instead scanner's one as the approach adopted in this project. Another type of approach, used in literature, is to work at lower level [89, 90]. Several researchers are focusing their attention in tuning the settings parameters of the discovery process. The authors, in [91], tune the discovery parameters according the mathematical theory they proposed in order to improve the device discovery both in terms of power efficiency and discoverability. One of the closest approach to that one used in this project, is presented in [92]. In the article the authors use an adaptive approach for significantly improving the device discoverability and the power efficiency of the process. This approach differs from that one we present in terms of discovery protocol. A similar idea for context-aware discovery has been adopted by [93]. In that article the authors presents an technology agnostic context-aware approach for discovery. CANDi, this is the name of the presented protocol, works on clusters of nodes with a more complex and structured architecture. CANDi works at MAC level. On the other hand, the adopted approach proposed in this research, works with the granularity of single node, and presents a much more lightweight and simpler architecture. The adopted approaches in the protocols work at application level. In [92] is used an adaptive advertising strategy based on user's behavior, while our approach exploits the advantages provided by Fog. The fog has a role of main importance, indeed it helps to exploit the fully potential of IoT nodes and it paved the way to the development of our model and a power efficient node discovery protocol [94]. Finally, in December 2017, the last version of BLE has been introduced, the Bluetooth 5.0. This version has introduced several discovery features and improvements compared to the previous one. In [95], an evaluation of the BLE discovery performance of the new features of Bluetooth 5.0 is presented.

2.2.1 Energy Management

The IoT node discovery protocols adopted in this project also target the device sustainability and power efficiency. The protocols exploits the internet connectivity for communicating with fog middleware and to be alerted when a new IoT node is approaching. This type of scheme allows the IoT node to keep the BLE interface always OFF until it will be strictly needed, saving energy. At this point the last source of power consumption remain the internet connectivity. The power efficiency among the Internet connectivity interfaces is still under investigation and it drawing the attention of a relevant part of the community. Several studies have been done about the power efficiency of WiFi, LTE and 3G technologies, among the others. For example, in [96], the authors provide a in-depth comprehensive study on per-frame energy consumption of the WiFi interface. They address all the aspect of power consumption due to the behavior of the IEEE 802.11 interface for different device types (such as tablets, smartphones, wireless router, and embedded devices), and for both UDP and TCP traffic. On the other hand, in [97], the authors have analyzed the power efficiency aspect of the LTE technology. They highlight how LTE is power consuming, and propose a different timer setting to achieve a good trade-off between power saving and traffic performance. Finally, speaking of energy analysis of Internet connectivity, it is worthy to be highlighted [98]. In this article, an exhaustively study on power consumption among WiFi, 4G LTE and 3G technologies is addressed. The authors also provide a comprehensive comparison of the power consumption under the technologies aforementioned. From this analysis is emerged that in terms of power consumption WiFi is more efficient than LTE and 3G, while these mobile technologies, especially LTE, are more suitable for applications that require an high throughput.

This research also proposes an optimization model, based on the sky rental problem formulation, to save energy by introducing an adaptive BLE interface switching ON/OFF strategy based on the advertisers' arrival frequency. Starting from more theoretical approaches for the smart management of energy. In this paper, we have adopted an energy-saving strategy based on the Ski Rental problem [99]. This type of theoretical problem belongs to the class of problems to help in choosing between a periodic cost, paid repeatedly (rent a pair of skis), and a determined price paid once (buying price) also known as the "ski rental problem". It can be demonstrated that the optimal off-line deterministic strategy to minimize losses for this class of problems is to pay the repeated rent cost until it is equal to the buying price, after then it is better paying the buying cost. We used an off-line deterministic strategy because this kind of approaches are the simplest and achieve easily reproducible results [100]. The authors in [101], demonstrate that this problem is 2-competitive. This kind of strategy perfectly suits that kind of problem in which there is the dilemma between paying a small amount repeatedly or paying a bigger cost just once; another application of this strategy is in [102]. In addition to a power efficient strategy the choice of communication technology plays an important role on the power consumption balance of the mobile nodes.

2.3 Cloud-based Support for IoT

The success of the IoT world requires service provision attributed with ubiquity, reliability, high-performance, efficiency, and scalability [103]. This research project aims to exploit the Semantic Web and its standard de-facto query language SPARQL for exposing the heterogeneous abstraction of services at cloud level. Many researches have been addressed the IoT-cloud data delivery model to try of achieving efficiency and scalability. For example, in [104], the authors have proposed a PaaS framework that provides essential platform services for IoT solution providers to efficiently deliver and continuously extend their services. The basis for the convergence of cloud and IoT was established in the work of Web of Things [105], which has proposed a set of methods to access devices through web-based technologies such as web services and RESTful interfaces. It has solved the problem of managing and using IoT resources in a service-oriented framework. The early work on the convergence of cloud and IoT are mostly direct applications of WoT architecture on cloud. Hassan et al. integrated cloud and wireless sensor networks (WSN) by developing several key functional components of WSN on cloud, namely pub/sub broker and resource registry [106]. In [107], the authors virtualize physical sensors as software entities on cloud, which provides users with sensory service provisioning, resource management, and monitoring. Semantic technologies have long been adopted in modeling sensory information. Alam et al. [108, 109], enhanced sensor virtualization through semantic abstraction for sensor capabilities. Generally, the focus of the early work is on IoT resource management rather than service delivery. Cloud is viewed as computing infrastructure to facilitate the management of large amounts of IoT resources. Following these early results, domain-specific systems have been proposed on ambient living [110], healthcare [111], agriculture [112], and so on. Since recently, new research initiatives have started to emerge on exploiting the service delivery models of cloud to accommodate the growing scale and diversity of IoT services. Soldatos et al. in [113], has presented the idea of converging IoT and utility computing on cloud as the core concept of the OpenIoT project12. The proposed architecture is based on CoAP and linked data [114]. The work uses the cloud concept at infrastructure level, in the way that the utility of services provided by interconnected objects is measured. However, the concept does not address the problem of efficient service delivery and multi-tenancy in a PaaS model.

As it is already stated, among all the approaches listed above to make IoT data available to the cloud, in this research has been chosen the Semantic Web.

2.3.1 Semantic Web

The Semantic Web, as envisioned by Tim Berners-Lee and described by the W3C Semantic Web Activity, is an evolving extension of the World Wide Web in which the semantics, or meaning, of information on the Web is formally defined. Formal definitions are captured in ontologies, making it possible for machines to interpret and relate data content more effectively. The principal technologies of the Semantic Web include the Resource Description Framework (RDF) data representation model and the ontology representation languages RDF Schema and Web Ontology Language (OWL). Semantic Web has long ago emerged as the transformation of the World Wide Web in an environment where contents (HTML pages, binary files, images/media content, and so on) are enriched with metadata that specify the semantic context of any content itself. Metadata languages and formats (such as RDF and OWL) are primarily conceived to easily express information about content, and automatically perform semantic data query (e.g., via search engines), interpretation, and, more generally, to automatic aggregation and reasoning. Semantic Web advances interaction between computers and humans one step further, and allows humans to leverage a more autonomous and intelligent machine support in the execution of the generic tasks. A key element of the Semantic Web is W3C SPARQL, which is a query language for Resource Description Framework (RDF), and has long set itself as the de facto standard to perform semantic queries on content exposed on the Web. RDF describes the concepts and relationships about them through the introduction of triples (subject-predicate-object); triples that have some elements in common become parts of a knowledge graph. SPARQL helps navigating such knowledge graphs and searching for sub-graphs corresponding to user requests. Semantic Web and SPARQL query language form a promising platform to support Data Governance and federation needs, and allow building a connected network of information [115]. However, federation of semantic data and navigation via SPARQL queries is still at an early stage and requires users to explicitly express the distributed nodes upon which to perform semantic queries and subsequent result aggregations, therefore negating the intrinsic benefits of adopting a semantic approach to distributed data aggregation and reasoning [116, 117]. To exploit the Semantic Web as a resource for the resource discovery and providing at cloud level, let me introduce the concept of Semantics of Sensors. Sensors encoding of observed phenomena are by nature opaque (often in binary or proprietary formats); therefore, metadata play an essential role in managing sensor data. A semantically rich sensor network would provide spatial, temporal, and thematic information essential for discovering and analyzing sensor data. Spatial metadata provide information regarding the sensor location and data, in terms of either a geographical reference system, local reference, or named location

Local Metadata Local reference is especially useful when a sensor is attached to a moving object such as a car or airplane. Although the sensor's location is constantly changing, its location can be statically determined relative to the moving object. In addition, data from remote sensors, such as video and images from cameras and satellites, require complex spatial models to represent the field of view being monitored, which is distinct from the sensor's location.

Temporal Metadata Temporal metadata provides information regarding the time instant or interval when the sensor data is captured. Thematic metadata describe a real-world state from sensor observations, such as objects or events. Every discipline contains unique domain-specific information, such as concepts describing weather phenomena, structural integrity values of buildings, and biomedical events representing a patient's health status.

Thematic metadata can be created or derived by several means, such as sensor data analysis, extraction of textual descriptions, or social tagging.

The location type within the query could be a single coordinate location, a spatial region within a bounding-box, or a named location such as a park or school. The semantics of the time interval specified by the query could be about weather conditions that fall within the time interval, contain the time interval, or overlap with the time interval. The type of metadata necessary to answer the queries listed requires knowledge of the situation the sensors observe. Such knowledge can be represented in ontologies and used to annotate and reason over sensor data to answer complex queries. Initially, IoT is considered as a RFID based technology. However, now the vision is much broader and anticipated as combination of different technology enablers such as wireless sensor networks, SOA, semantic web, and virtualization. These technology enablers act as building blocks for the realization of the Web of Thing (WoT) vision. In this regard, open geospatial consortium (OGC) provides a suite of standards (i.e., sensor web enablement (SWE)) [118] that provides base line technologies for modeling sensors, observations and transducers. It also provides specifications for sensor observation, planning, alert, and notification services at cloud level. Jongwoo Sung and colleagues proposed metadata framework that stores sensor nodes capabilities in distributed servers on the internet instead of sensor nodes [119]. Sensor metadata is described in XML format and accessed via web services or HTTP. The framework has the advantage of minimizing the metadata overhead and discovering sensor services efficiently. In [120], the authors highlighted the importance of resource discovery in the wireless sensor networks and provided the solution in the situation of heterogeneity and mobility. The simulation results show that the usage of customized encoding formats consume less energy but lack interoperability. On the other hand, XML encoding formats provide interoperability at the cost of more energy consumption. Thus, there is a tradeoff between efficiency and interoperability. The aforementioned research works lack in providing the intended meaning to the data due to the use of XML. In relation to make sense of sensor data and services, Amit Sheth and his colleagues implied semantic web technologies to annotate the sensor data with semantic metadata, exactly as this research proposed, which results in increasing interoperability [121]. Furthermore, different vocabularies are proposed to model sensor, and smart objects services [122, 123, 124]. While, in Haung and Javed, in [125], proposed a Semantic Web Architecture for Sensor Networks (SWASN), focusing on sensor data and inference over sensor data for wide range of WSNs. In [119], is presented a general architecture, attempting to meet the IoT requirements. However, the work only considered the near field communication (NFC) enabled mobile phones to sense physical objects embedded with NFC tags. The interaction between user and services is handled by interaction proxy, which utilized the semantic reasoning capabilities for composing services and entertaining the user request. Most recent works on smart objects are focusing on linking and sharing smart object services and data so that the novel IoT application will be able to enhance their autonomous decision making capabilities. In this regard, in [126], the authors introduced the inspiring concept of Linked Sensor Data. They developed a framework to make sensor data and metadata publicly accessible by publishing it on the Linked Open Data (LOD) cloud. The enablement of smart object sensor data in the form of Linked Sensor data brings different challenges such as discovery, access control and privacy. In [127], authors proposed a middleware that includes a sensor registry and a sensor discovery service to access and discover sensor using named- locations. Most of the aforementioned research works either focus only on vocabulary or sensor network framework. Whereas, we imply a holistic approach, considering the complete IoT environment and provide a semantic overlay of underneath infrastructure that enables the framework to offer IoT capabilities in the form of web services.

Chapter 3

Middleware Architecture and Discovery Protocol

This chapter is the core part of this dissertation, it will present the model of virtual sensor and its architecture, then the fog middleware architecture and two smart fog middleware driven device discovery protocols will be presented and illustrated. Finally, this chapter concludes with the experimental results of the proposed protocols.

3.1 DaaS and Virtual Sensor

The main goal of this research is to realize a fog-based middleware capable of overcoming the hottest still open issues of IoT, and enabling an easy, secure, technology-independent interaction with the cloud. The two elements that have been identified as key enabling components in order to fulfill the final goal, are: a fog-based architecture, and a smart fog-driven device discovery protocol. The fog-based architecture bases its core on the appliance of the fog paradigm to the IoT world and on the exploiting its high standing qualities to overcome the IoT limitations. The architecture exploits a smart fog-driven discovery protocol to discover any IoT device in a completely technology independent manner. This protocol has a crucial relevance for overcoming the heterogeneity of the IoT devices and prolonging the battery life time of IoT nodes. Indeed, the proposed protocol exploit the location awareness, geo-distribution, the mobility support, local management and overview provided by fog paradigm to solve the technology difference related problem of IoT nodes, improve the discoverability capacity of traditional IoT discovery schemes, and prolong the battery life-time of the devices. The higher level model, this project is presenting, is DaaS. As it is already stated, DaaS's goal is to abstract the physical IoT objects in order to surpass their limitation in terms of hardware and technology heterogeneity, aggregate them to provide better solutions to complex scenarios, solve the problems due to physical node high dynamicity, fault tolerance and battery life. In a nutshell, DaaS is a virtual sensor. The virtual sensor model exploits the above mentioned fog-based architecture and smart fog-driven device discovery protocol to provide an abstraction of the underlying world at cloud level. The next section describes a comprehensive model of virtual sensor and its architecture, then the next two, heavily focus of the core part of this project, the fog-based architecture and the smart fog-driven device discovery protocol, along with in-depth detail discussion and the experimental results.

3.1.1 Virtual Sensor Architecture: Theoretical Model

This project aims to design, implement and evaluate a new paradigm for the IoT panorama, more precisely, this paradigm consist of a middleware capable to make available on the Cloud the physical heterogeneous sensors as virtual homogeneous devices. This will significantly contribute to the industrialization and the massive wide spreading of the IoT-Cloud applications and developments. The proposed virtual sensor architecture model is laid on three layers and bases its core on the fog intermediate tier. The architecture is inspired by the three layered fog computing architecture described in the subsection 1.2.2. The fig 3.1 illustrates the proposed virtual sensor architecture. The fog-based middleware has most crucial role in the architecture. It enables the interactions between IoT and Cloud layers. It fetches data from IoT tier, manages and aggregates them, and it finally makes them available at cloud level as virtual data source, decoupling the created abstraction from its real physical source.

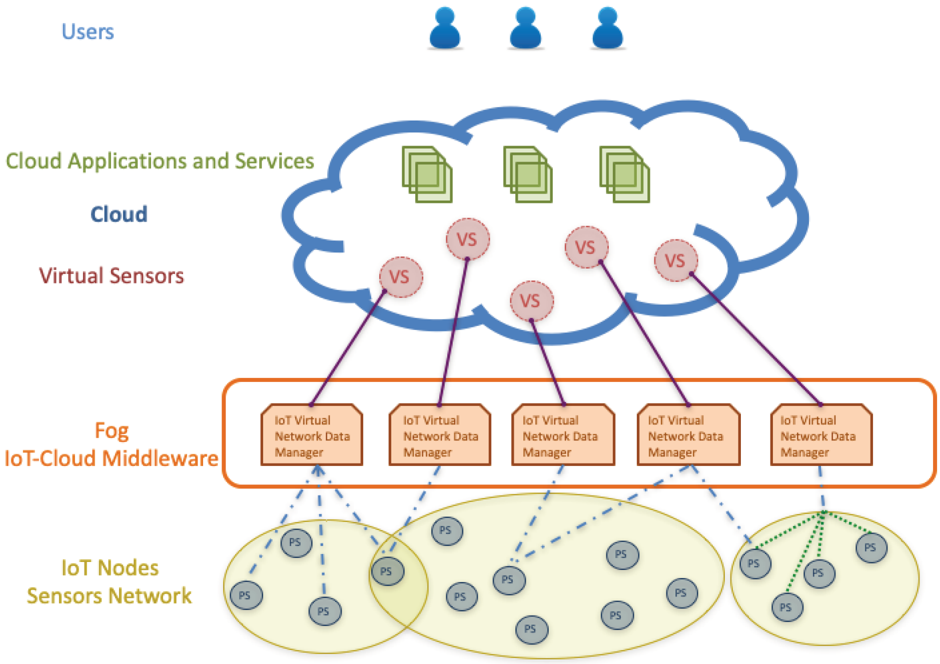


Figure 3.1: Virtual Sensor Architecture

.

The three layer of the architecture in fig 3.1 are detailed, from the bottom to the top. as follow:

IoT Nodes Tier This is the bottom layer of the architecture and it consists of all IoT devices. This layer accounts all the sensing and acting activities of IoT nodes, it also handle a chunk of the device discovery process, and it enables the nodes D2D communication. The IoT Nodes layer strongly interacts with the upper Fog Middleware, exploiting its features to improve the activities and the interactions among the nodes. The nodes of the IoT layer provides the environment information, and data sensed or acted to the Fog Middleware

Fog Middleware Tier The Fog Middleware layer is the mid tier of the architecture and it has most relevant role. It accounts the creation of the physical IoT node abstraction and its providing to upper Cloud Layer. At this scope, Fog middleware has to manage the both side of interaction, IoT Nodes - Fog Middleware, and Fog Middleware - Cloud. This tier receives and harvests the big amount of data from IoT nodes, processes, aggregates, and manipulates it in order to provide an on-demand abstraction to Cloud layer services and applications. Fog Middleware exploits the environment information and data provided by IoT Nodes tier to create a proper virtual sensor configuration, mapped on real physical objects, able to hit the cloud services and applications requirements. The type of virtual sensors' configuration is addressed more in detail in the subsection 3.1.2. On the Cloud side, the Fog Middleware layer, provides virtual sensors as a abstract logical entities and the methods to access and query them.

Cloud Virtual Sensor Tier The Cloud Virtual Sensor Tier is the highest layer of the architecture. It accounts the exploiting of virtual sensors, provided by underlying tier, to meet the requirements and request of the final user applications and/or other services. This layer is totally unaware of how the virtual sensor is really realized, located, obtained and queried, it sees the virtual sensor as a service that can be queried or demanded. The Cloud Virtual Sensor layer requests to Fog Middleware for a virtual sensor, then restricts itself to provide a way to access and interact with it. In a nutshell, the main role of this layer is to expose the virtual sensor to the requester. For requester is meant any type entity allow to request a virtual sensor, it can be a final user, an application, a service, or any combination of the previous.

The proposed middleware creates an abstraction layer that absolutely uncouples the physical and the business/applications layer. With this abstraction level the real devices will be totally hidden, making the applications entirely independent from device's identity, location, connection technology, etc. Second, this extremely simplifies the interaction and the integration of widespread and industry-relevant cloud/sensor already existent solutions. Third, the virtual sensors into the Cloud could acts as a data manager/manipulator/handler, in this way, it optimizes the IoT - Cloud communications in terms of traffic, reduction of latency, response times and device's energy consumption. Fourth, the improvement of device recovery and availability is another main aspect the proposed architectural model can theoretically achieve. With the virtual sensor a faulted physical real device might be transparently replaced with a backup device without any service interruption. Fifth, virtual sensor enables and facilitates the facing of incoming system load peak conditions. With an appropriated resizing/reshaping of the virtual sensor network, the incoming load peak can be balanced. An efficient and effortlessly resizing/reshaping of the virtual sensor network can be achieved, through a smart fog-driven device discovery protocol and a dynamic device provisioning.

3.1.2 Virtual Sensor Configuration

As it already stated in the previous subsection, the interaction Fog Middelware - IoT Nodes layer has a crucial importance in the virtual sensor creation process. This interaction is the key for the decoupling between physical device and virtual sensor. The virtual sensor abstraction can be created by combining an arbitrary number or types of physical device. A physical real device might provide data and information to multiple distinct virtual sensors or, a single virtual sensor might fetch information and data from multiple physical devices or, multiple virtual sensors might expose data and information provided by multiple real devices, or even more, a virtual sensor might be obtained by a combination of multiple devices of different types. There are many configuration possible between real physical nodes and virtual sensors, some of them have been presented in [128]. This idea has been extended by introducing network/configuration's reshaping and scalability. Following, the fig 3.2 depicting the configuration of virtual, along their description.

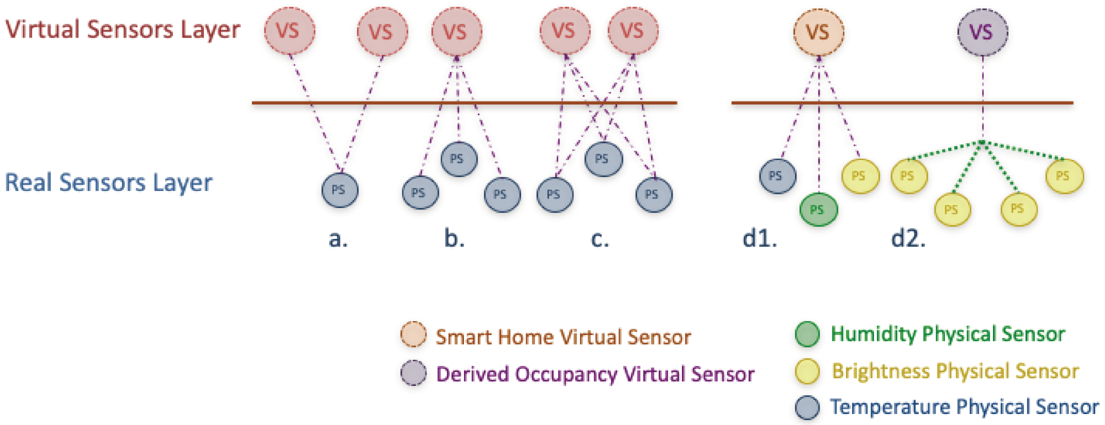


Figure 3.2: Virtual Sensor Configurations

.

We envision four initial Virtual Sensor-WSN configurations are:

One-To-Many : One physical device (inside a WSN or stand-alone) is shared among many Virtual Sensors (or a single virtual sensor) and its data and its control is accessible by every virtual sensors sharing it. Fig 3.2.a.

Many-To-One This configuration provides an aggregated data from a wide Wireless Sensor Network. Often an application can be interested to a particular type of data, regarding a specific phenomena; this configuration makes the application gather that specific type of data enabling all, and only, sensors in the WSN those relate to that particular interesting phenomena. By aggregating data, this configuration provides fault tolerance and data quality. Fig 3.2.b.

Many-To-Many This configuration trivially represents the combination between the two above. A physical object can correspond to many virtual sensors and being part of aggregated data sensors network. Fig 3.2.c.

Derived The derived configuration is the generalization of all the configurations above. It exposes a virtual sensor as an aggregation of several different types of physical real devices (other configurations work with devices of the same type instead). The virtual sensor with a a Derived configuration allows to sense complex event type, for example it is possible to monitor the state of an entire room with a single Virtual Sensor configured as Derived (fig 3.2.d.1). Moreover, Derived configuration, has another impressive application, it allows the simulation of data gathered by a sensor not physically deployed in a local area. To clarify this remarkable aspect, it is enough to think to an occupancy virtual sensor obtained by a derived configuration of a multitude of brightness sensors deployed in a room. The occupancy information is derived by the calculation of the interpolation of the brightness devices' light intensity variation (fig 3.2.d.2).

The crucially main role for the creation of virtual sensor configuration is played by the discovery protocol. The discovery protocol has to be able to overcome the heterogeneity of the different devices. The efficiency of the protocol has a main importance. Indeed, in environments extremely dynamic like IoT, the discoverability of a new device is decisive; moreover in case of configuration reshaping need, elasticity need, fault tolerance handling, and quality of service or data.
The next subsection addresses the proposed smart fog-driven device discovery protocol.

3.2 Smart Fog-Driven Device Discovery Protocol

The smart fog-driven discovery protocol, as it has already described, is one of the two core part of this project. IoT nodes can have different nature, resource capabilities, mobility pattern, battery life etc. The proposed discovery protocol has to be technology independent in order to overcome such heterogeneity. it has to be applicable to any IoT environment, scenario, and object type. In a remarkably dynamic environment like IoT, the efficiency of a discovery protocol is a characteristic with a certainly high importance, moreover for the middleware and the goals this project aims to achieve. An high efficiency protocol, in terms of device discoverability, is capable to assure the new device discoverability regardless its type, its speed, its mobility pattern, etc. This feature is crucial for the proposed middleware. Indeed, the creation of virtual sensor configuration, the continuous availability, the fault tolerance, the WSN elasticity, the virtual sensor homogeneity, are all features achievable only with an extremely high efficiency of the device discovery protocol. In addition, the battery life of the IoT device has a remarkable role. As it is already stated, IoT scenarios have an high dinamicity and the continuously running of the discovery protocol drives to a quite quick battery drain. Hence, a protocol capable to save power and reduce the energy waste surely brings huge benefits. The proposed device discovery protocol aims to overcome the device heterogeneity, be discoverability efficient, and be energy sustainable, by exploiting the main quality provided by fog paradigm. The proposed protocol has an hybrid nature and it is run part in D2D mode among IoT objects, and part with a IoT device - fog interaction. This fog-driven hybrid protocol falls in both the discovery protocol family introduced in subsection 1.3.2, Request-Based Discovery and Direct Discovery. The protocol is hardware agnostic as long as the IoT object has internet connectivity. According to its nature, the internet connectivity is necessary in order to let the IoT node interact with fog. The necessity of internet connectivity is not a such strong constraint, because all nowadays smart object has at least one interface capable of internet connection. Moreover, by definition, an IoT object should have internet connectivity. The proposed smart fog-driven device discovery protocol follows the following generic steps:

1. At the beginning fog loads to IoT objects the areas of interest
2. When the IoT nodes enters in any area of interest, it starts to periodically ping the fog with its position
3. The fog node alerts the in-range IoT device closest to the approaching node
4. The alerted IoT object initiates the D2D communication
5. The alerted IoT object establishes the connection with the approaching node
6. The IoT node became part of the network

Nowadays Bluetooth is the standard de-facto technology for the wireless connecting any IoT devices. The Bluetooth SIG, with the release of Bluetooth Low Energy (BLE), has paved the way to the massive adoption of Bluetooth as node communication enabling technology in Iot field. While the IoT fog interaction for the alerting/wake-up mechanism is based on Message Queue Telemetry Transport (MQTT), a lightweight M2M communication protocol suited for resource constrained devices and largely used in IoT. MQTT is one of the most widespread, adopted, supported and used in IoT environments. For this reason BLE and MQTT have been selected as study case technologies for the proposed smart fog-driven discovery protocol.
In the next subsection is provided an overview of the MQTT and Bluetooth Low Energy discovery process.

3.2.1 IoT Application-Layer Protocols

There are two main categories of application-layer protocols in IoT: Publish/Subscribe and Request/Response protocols. The two most representative protocols of these two categories are Message Queue Telemetry Transport (MQTT) and COnstraint Application Protocol (COAP), respectively. The architecture of COAP is based on REpresentational State Transfer (REST). COAP can work on UDP over 6LowPAN, and its implementations include reliability techniques in order to overcome the unreliability of UDP [129]. The security services are considered as a drawback in COAP. Therefore, its implementations use Datagram Transport Layer Security (DTLS) for protecting the UDP messages, although DTLS does not fit well in the IoT applications [130]. As regards MQTT, it belongs to the Publish/Subscribe category, and it is a lightweight M2M communication protocol. As opposed to COAP, MQTT works on simplified TCP. Pub/sub messaging requires less communication bandwidth and computational power in IoT applications when compared to request/response messaging protocol like COAP [130]. While there are reference studies for designing publish/subscribe patterns for COAP [131], the common-sense is that MQTT is perfected tailored for resource-constrained devices with limited bandwidth. MQTT architecture can be broken down into three main components as follows:

Subscriber , is an entity that is subscribed to one or more topics, which automatically receives updates from the broker about its subscribed topics;

Publisher , is an entity that publishes messages on one or more topics while these messages are delivered to the subscribers by the Broker.

Broker , is a server that waits for the incoming messages on the topics, and dispatches them upon receipt. It is worth mentioning that publisher-subscriber peers communicate through a topic without knowing each other. that being said, the broker is also responsible for ensuring security in the application layer [129].

MQTT is highly capable of suiting the needs of resource-limited devices. In particular, MQTT has the potential to prolong the battery lifetime of IoT devices. However, the impact of MQTT protocol on the lifetime of IoT devices is still under investigation. For example, in [132] the authors state that the system can be put in sleep state; however, the analysis of individual components in an MQTT-driven node discovery system remains an open issue. Besides these, a comparative study on the impact of the application-layer protocols on IoT networks [133] reports that COAP is preferable if energy efficiency/battery lifetime is a concern whereas whereas MQTT is a beneficial choice when reliability is a concern. More recently, the authors in [134] presented a comparison of COAP, MQTT, MQTT-SN, WebSocket, and TCP, and provided insights for the performance and energy aspects of these protocols. To wrap up the debates on the effectiveness and efficiency of application layer protocols, it can be stated that there is almost a consensus on the suitability of the MQTT protocol for IoT applications that seek persistence of messages, secure multicast communications and high reliability.

3.2.2 Bluetooth Low Energy Discovery

Bluetooth Low Energy (BLE) is a fairly recent wireless technology with widespread usage. It represents a remarkable improvement over the regular Bluetooth, and it has been introduced since Bluetooth version 4.1. All the improvements (both hardware and software) that BLE has brought up paved the way to a widespread diffusion and use of this technology in the IoT domain because BLE meets most of IoT needs, especially in device discovery. The device discovery process in BLE is led through two main entities, a scanner, since now on named BLE-S, and an advertiser, since now on named BLE-A. The BLE-A is the entity that typically broadcasts packets, namely advertising packets, to allow

the BLE-A to be discovered by other BLE-S devices. The presence of an advertising packet in the scanning window of a BLE-S implies the discovery of the BLE-S.
There are two types of advertising processes:

- advertisement to establish a bidirectional connection between BLE-A and BLE-S
- 2) transfer of information without requesting any connection

Indeed, each advertising process has its own advertisement packet type. When a BLE-A is advertising, it sends the same packet on three channels (namely, 37, 38, 39, at 2402 MHz, 2426 MHz, 2480 MHz respectively) at the physical layer. The frequencies of the channels have been selected in order to minimize the interference with the WiFi channels. Then, the advertising cycle consists of sending the same advertising packet on the channel 37, 38, 39 consecutively as depicted in Fig. 3.3. It is pretty intuitive that transmission of advertising packets on fewer channels would save power, however power savings will be at the expense of lower discoverability of the BLE-A device.

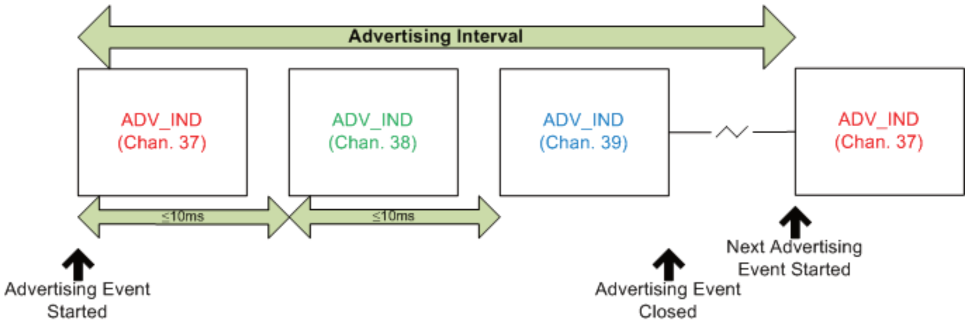


Figure 3.3: BLE Advertising process duty-cycle

.

BLE-S plays the other main role in the BLE discovery (scanning) process. Indeed, BLE-S can activate and scan for other advertising devices at any time. Two types of BLE device discovery procedures can be performed, namely active and passive scanning. In the former, BLE-S, after having detected a new BLE-A, may send a further scan request as an inquiry about the new device whereas in the latter, the BLE-S can only receive the advertising data from the BLE-A. Analogously to advertising process, the scanner listens for advertising packets on the same three channels (i.e., 37, 38, and 39). A BLE-S remains in the listening mode for new devices for a fixed time window. The time window in which the BLE-S remains in the listening mode is referred to as the scan window or scan frame. Thus, BLE-S initiates a scanning window on each channel cyclically for a fixed period, which stands for the *scanning frame length*. In case an advertising packet is detected in a scanning window, the BLA-A is marked as discovered. Fig. 3.4 illustrates the scanning process.
Prior to wrapping up the brief review of the BLE discovery process, it is worth noting that parameters in both processes are tunable. For the advertisement procedure, these parameters include: advertising interval, combination of advertising channels, and advertising type in advertising scheme. For the scanning procedure, the parameters are: scan interval, window length, duration, and type. Although all of these parameters are configurable, it might happen that BLE-S and BLE-A do not discover each other with a high

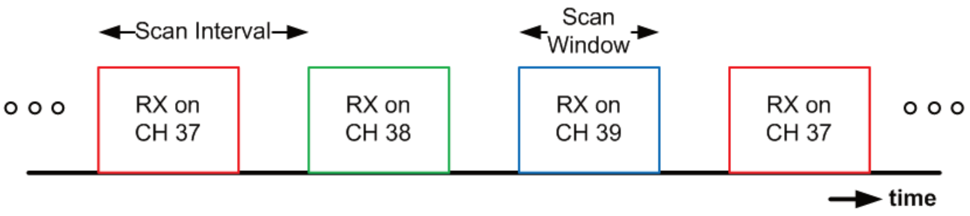


Figure 3.4: BLE scanning process duty-cycle

.

success rate. A possible scenario, which may cause low success rate in the discovery can be as follows: an advertising packet is not launched on a different channel or at a different time with respect to the scanning window open time. Moreover, as better discussed in the following, properly configuring these parameters would make it possible to significantly improve energy efficiency of the standard BLE discovery protocol. The proposed smart fog-driven device discovery protocol provide insights for the executing context at the service discovery layer, by the improving of discovery protocol performances in terms of success of the discovery process, and enabling power-efficient discovery.

3.2.3 Proposed Sustainable Discovery Protocol

3.2.4 Model

The wide spread of wearable, Body Area Network (BAN), and Personal Area Network (PAN) calls for new solutions able to efficiently support the continuous connectivity of these smart objects. As it is already stated, Bluetooth has become the standard de-facto technology for the wireless connecting any smart devices. This trend has led the whole research community to focus its attention on the crucial aspects of this technology. In the mobile communication field, one of the hottest topic is the node discovery process. Moreover, speaking of battery-operated mobile nodes, another crucial aspect to address is the energy management. This is the main cause developing a smart fog-driven device discovery protocol is crucial for the proposed fog middleware. The device discovery has a central role in IoT, it enables the nodes connection and the IoT service providing. BLE has paved the way to the massive adoption of Bluetooth as node communication enabling technology in Iot field. BLE is able to provide a quite simple discovery process keeping the energy consumption relatively low. The applications of Bluetooth Low Energy discovery-based systems fall several fields such as marketing advertisement, city's point of interest discovery, sports performance monitoring, indoor positioning systems, smart health systems, just to name few. The traditional or conventional BLE discovery process has a quite simple and trivial architecture, it involves two type of entities, a scanner and an advertiser. The BLE scanner (BLE-S) looks for other devices in the proximity, while the BLE advertiser (BLE-A) announces its presence to the nearby BLE-S. Let me introduce an example of a typical conventional BLE discovery scenario, a BLE beacon is located in proximity of a painting in a museum and it is continuously in the active state scanning the vicinity for other devices. In this scenario, when a new device approaches, it gets discovered and the scanner sends to it all the information regarding the painting. In this traditional BLE discovery scenario, one of the two entities has a static location and continuously advertises its presence or scans for other devices while, on the other hand, the other entity, it supposed to be moving. I consider this as my reference, a scenario in which the BLE-Ss have a static and known position, while the BLE-As are mobile. In the default BLE discovery protocol, the BLE-S constantly and periodically scans for devices in the proximity, being the scan process an endless loop. The other entity of the model, the BLE-A, continuously advertises its presence. As I mentioned before, the BLE-A does not have a static and fixed location, it is in continual movement, carried by pedestrians. The BLE discovery process succeeds when a advertising packet sent by BLE-A, hits a BLE-S active scanning window. In other words, when an advertising BLE-A is close enough to a scanning BLE-S to allow an advertising packet to hit the BLE-S's scanning window, the BLE-A is discovered. This traditional discovery approach might result to be inefficient. For example, the BLE-Ss keep scanning for devices even if there are no BLE-As nearby, on the other hand, the BLE-As keep advertising even in the areas where there are no BLE-Ss. Even if a BLE-S and a BLE-A are in proximity, they might not discover each other cause the scanning and the advertising process are not synchronized (for a detailed description of how the discovery process works at low level, refer 3.2.2). If a BLE-A sends an advertising packet on a channel and a BLE-s is scanning on a different one, they does not discover each other. Hence, there might be the possibility that two devices do not discover each other despite they are in proximity. BLE is worldwide used as main short ranged node discovery technology, but it does not mean that its discovery process cannot be improved both in terms of device discoverability and power consumption. BLE is well known to be power efficient, but energy wastage could still occur and its power efficiency could be improved even more. I have already introduced the cases in which the conventional BLE discovery scenario results to be not efficient in terms of device discovererability, now I highlight its inefficiency in terms of power consumption. In the traditional BLE discovery scenario, the scanners have the BLE interface always active and scan for devices periodically and constantly, i.e. a BLE beacon. On the other hand, the advertisers have also the BLE interface always active and they are in advertising mode, continuously. This approach is clearly inefficient in terms of power consumption. A continuously scanning BLE-S, with no BLE-As around, is wasting energy. In the same way, a BLE-A would significantly save power if it does not advertise its presence in an area where there are no BLE-Ss. Under this point of view, the most power efficient approach would result in the BLE-S and BLE-A active their BLE interface, and start scanning/advertising only when they are in range of each other. The main idea driving this protocol is to exploit the fog paradigm for overcoming the aforementioned issues of conventional BLE discovery scenario.

By aiming at sustainable discovery of BLE nodes in IoT application domains, a smart fog-driven discovery protocol, is proposed. This protocol is fog-driven because it exploits the support provided by fog in order to improve the traditional discovery protocols and achieve the above mentioned goals. The interaction with fog is based on MQTT protocol, hence, since now on it will be referred as MQTT-driven protocol. It is named Power Efficient Node Discovery (PEND). PEND has been further extended in order to better overcome the energy wasting challenges it faced, which will be further presented. PEND adopts an MQTT-based BLE discovery which consists of three entities, BLE-S, BLE-A, and Fog

Node (FN). BLE-S is located at a previously determined fixed position to periodically scan the nearby BLE devices in the vicinity. As opposed to BLE-S, the BLE-A is not fixed to a static position; thus, BLE-A may or may not happen to be present in the scanning range of BLE-S at the time of scanning. The FNs are located in fixed position spread all over the territory as part of the infrastructure.

Firstly, let me recap what I am going to take as a base reference, namely Benchmark, traditional, conventional, or Regular scenario. It is a usual BLE discovery scenario wide used and known. It involves only two entities, the BLE-S and BLE-A without any type of support. In this scenario, BLE-S are in a fixed, static and known location and it scans for devices periodically and constantly. The BLE-As are capable to move and they advertises constantly and periodically. In this scenario there is not any intelligence and or support to the discovery process. In case of no devices nearby, the scanner keeps scanning the region as it is set to be active, scanning continuously and periodically. Under conventional discovery scenario, there is no guarantee of device discovery within BLE-S scanning frame. Indeed, there might be scanning frames with no devices discovered, or it is also possible to have a new device entering in BLE discovery range between two consecutive scanning frames resulting in the advertiser and the scanner being mutually non-discoverable. According to that, the traditional BLE node discovery results to waste significant power if BLE-S is battery-operated. The smart MQTT-driven discovery protocols aim to overcome these issues by exploiting the geo-location of the devices provided by FNs via Internet connectivity. The goal is to reduce the waste of energy of the conventional or benchmark approach and to improve significantly the DMR evaluated for each device. The Device Matching Ratio, DMR, is defined as the percentage of the scanning windows in which at least an BLE-A has been discovered out of the total number of scanning windows within a limited period of time. For example, if a BLE-S runs twenty scanning processes (scanning windows) in ten minutes and only in ten of them it discovers at least an BLE-A, the DMR results to be 50%. Consequently, with a 100% DMR, I mean the case in which a BLE-S discovers at least a BLE-A in each scanning window in a finite period of time. DMR is formulated as shown in Eq. 3.1 where F_s is the total number of scanning frames, and δ_f denotes a binary variable that is one if a BLE-A was discovered within the scanning frame f.

$$DMR = \frac{\sum_{f=1}^{F_s} \delta_f}{F_s} \tag{3.1}$$

Following, the novel proposed discovery protocol and its extension, along with technical discussion on the of battery power consumption and the improvement of DMR, will be addressed.

PEND: Power Efficient Node Discovery

In order to overcome to the deficiencies of the conventional BLE device discovery, PEND activates the BLE interface and the scanning procedure only if a new advertiser gets in the proximity of the BLE-S. A BLE-S is able to detect a new device inside its own BLE discovery range exploiting the geo-location of the devices provided and the location awareness by fog. Indeed, an integral part of PEND protocol is the interaction with the FN. This communication is based on MQTT protocol. In PEND scenario, the FN is supposed to carry an MQTT broker, installed and running. The location of every BLE-S is static and known by the server running on the FN. It is worth noting that each BLE-S is equipped with a WiFi interface, continuously active and subscribed on an application-specific MQTT topic on the broker. In this way, each BLE-S gets alerted via MQTT when a BLE-S is in its BLE discovery range. The scanning procedure in PEND scenario is MQTT-triggered. PEND also takes advantage of the low-cost WiFi connectivity as opposed to its counterparts. The figure 3.5 provides an illustration of PEND protocol and its message flow.

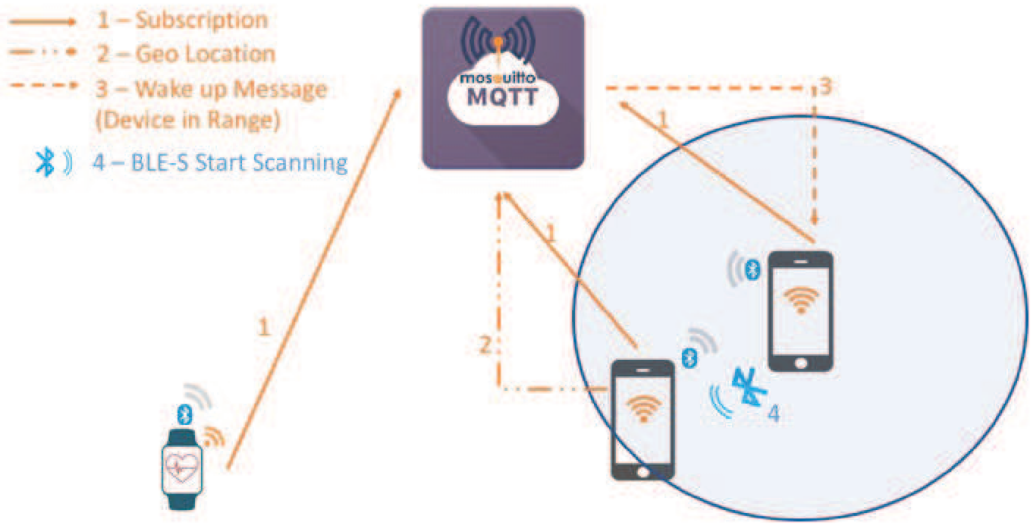


Figure 3.5: An illustration of PEND with message streams extending MQTT

.

In my scenario, Bluetooth interface of the BLE-S is initially off. Every BLE-A has preloaded all the geofences of the BLE-Ss' proximity. The BLE-As start to send their location to the MQTT broker on FN as soon as they get in a BLE-S's proximity. The geofences on the BLE-As allow the advertisers to send their location only when they enter in a scanner proximity region. Once that the BLE-As enter in the area, they sends their location periodically to fog. The FN calculates the distance between the BLE-A and the nearest BLE-S upon the MQTT broker has received the advertiser's location message. If the distance between scanner and advertiser falls within the BLE discovery range, the MQTT broker wakes the scanner up with a WAKEUP message. When a BLE-S receives a WAKEUP message, it enables its BLE interface, and it starts scanning. The scanning frame is kept active for a limited amount of time. The BLE interface is turned off upon the scanning period is over. Thus, according to PEND, a BLE-S activates the BLE interface and scans for devices only if an advertiser is in the vicinity, within a certain distance, and only for limited period of time. According with this behavior, I expect PEND improves the battery lifetime of a BLE-S which is not always plugged in, compared to conventional scenario. It is worthy to highlight that the main feature of the proposed protocol is to guarantee the discoverability of new devices. Indeed, the activation the BLE interface and scanning only when a new node is within the range ensures a 100% DMR. Despite the efficiency of PEND in terms of DMR and battery power consumption, it still experiences challenges.

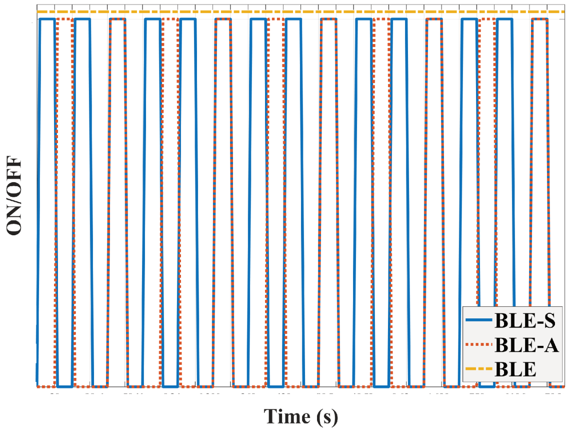
Firstly, selecting the optimal length of a scanning frame (L_s) requires thorough investigation. Secondly, the spawning frequency of a new BLE-A (f_a) remains an open issue whereas the proximity threshold for the MQTT broker to wake up a BLE-S (P_{th}) also requires further study. Moreover, a question might be risen: Is it possible to consume less power by having BLE always ON instead of switching on and off? In this paper, I aim at addressing these issues through a thorough performance study.

SPEND: Smart and Power Efficient Node Discovery

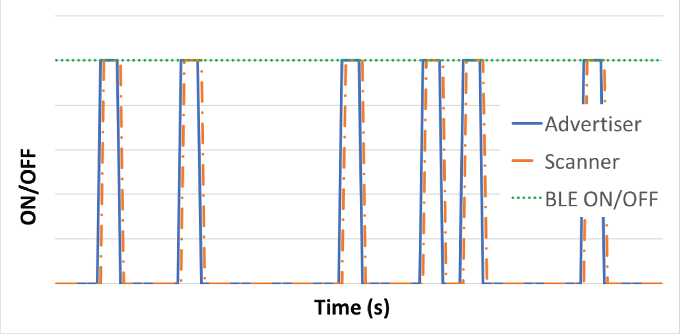
In this subsection, SPEND is introduced, as an improvement of PEND protocol. SPEND stands for Smart and Power Efficient Node Discovery (SPEND), and it is a smarter version of PEND protocol. Being PEND already ables to perform 100% DMR, with SPEND, I mainly aim to improve the power efficiency. In SPEND, the power consumption can be significantly reduced under some circumstances, however there is no guarantee for minimization. In PEND, the BLE-S is waken up by FN through a MQTT message, when a new node is detected as approaching. The BLE-S starts scanning for the new nearby device for the duration of the scanning frame. PEND keeps the BLE interface active and scanning for a deterministic duration, and this might turn into energy waste if BLE-S immediately discover the BLE-A and the scanning frame length is long. The deterministic active state duration of the BLE interface in PEND might result in increased power consumption under long frame lengths. In order to save more energy as possible, SPEND forces the scanner to terminate the scanning process as soon as the nearby approaching BLE-A has been discovered, regardless of whether the scanning frame should end. In SPEND the BLE-A sends to FN either the location and the UUID of the device. This UUID is forwarded to BLE-S as content of WAKEUP message by FN. In this way the BLE-S is able to perform a selective scanning process, guaranteeing the discovering only that specific device. With a selective scanning, I are sure of closing the scanning window only when the new nearby approaching BLE-A has been discovered. In SPEND, FN plays a important role even more, indeed it enables the BLE-Ss to perform a filtered scanning. The filter is based on a device's UUID, and it makes the discovery process ables to discover only those specific devices effectively interested in the service or information provided by the scanner. Moreover, the filter allows the BLE-Ss to shut the scanning window down as soon as those devices have been discovered, saving power. If the new nearby approaching device has not been discovered after a certain amount of time, the scanning frame terminates anyway. In the worst case, SPEND results to be PEND. With this kind of approach, SPEND saves power in the most of cases, in the worst case the performances in terms of power consumption are similar to PEND.

In order to highlight even more the different behaviors of the three proposed schemes, conventional or benchmark, PEND, and SPEND, the fig 3.6 illustrates the behavioral wave forms of the three approaches.

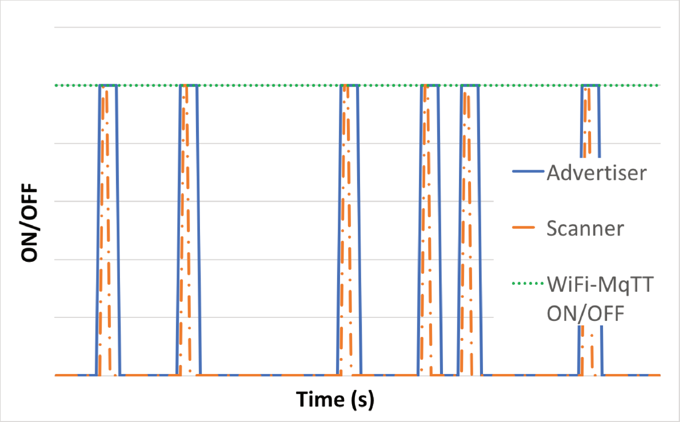
The sub figures 3.6a, 3.6b, 3.6c depict the behavioural waveform of benchmark, PEND, and SPEND schemes respectively. In fig. 3.6a, the continuous blue line represents the scanning frames of BLE-S, the dotted orange line represents the advertising frames of BLE-A, and the dashed yellow line is the behaviour of BLE-S's BLE interface. As it is easy to spot, the BLE-S scans the proximity periodically and constantly, while the BLE-A does



(a) The Benchmark Waveform



(b) PEND Waveform



(c) SPEND Waveform

Figure 3.6: An illustrative scenario for energy savings and DMR (a) Waveforms of BLE-S and BLE-A frames, and the Bluetooth (BLE) interface of the BLE-S under conventional discovery (i.e. benchmark), (b) Waveforms of BLE-S, BLE-A frames as well as the WiFi interface under the MQTT-driven PEND, (c) Waveforms of BLE-S, BLE-A frames as well as the WiFi interface under the MQTT-driven SPEND. In (a), the Bluetooth interface is always ON while the BLE-A and BLE-S continuously keep sending advertisement and scanning frames, respectively. Discovery of the BLE-A becomes possible when the advertisement and scanning frames overlap. In (b),(c) the wave forms of BLE-S and BLE-A are overlapped, this because, under PEND and SPEND, the BLE-S scanning is triggered by the MQTT Broker when the BLE-A is nearby (detected via WiFi) and advertising. These signaling protocols ensure synchronization of the scanning and advertising frames resulting in 100% DMR.

not have a static and fixed mobility pattern, it could not enter at all in BLE-S range or it can enter between two consecutive BLE-S scanning frames. In the first case the BLE-S keeps scanning in vain the proximity, wasting energy. Even worse, in the second case, the BLE-A enters in the BLE-S discovery range without has been discovered. In both of the cases it results in energy waste and bad DMR. In fig 3.6b, and 3.6c, the continuous blue lines represent the advertising frames of BLE-A, and the dot-dashed orange lines are the representation of BLE-S scanning frames. As it possible to see in both sub figures the BLE-S scans the area for new approaching devices only when a new BLE-A is effectively in-range. This particular behaviour is possible because the activation of BLE-S scanning process is triggered by the FN via MQTT. Here, it is even more highlighted, how the exploiting of fog is essential to provide an high DMR and to reduce the wasting of energy. Figures 3.6b, and 3.6c, hence PEND and SPEND, differs to each other for the BLE-S scanning policy. In fig 3.6b, is possible to notice how the BLE-S scanning windows is kept active for a fixed amount of time, namely scanning frame length, and for its whole duration. While, on the other hand, from fig. 3.6c, emerges how SPEND immediately shuts the scanning frame down upon the approaching BLE-A has been discovered.

Before presenting the high number of experiments run under several conditions, let me introduce the other core part of this project, the hybrid IoT-fog architecture that, through the above presented protocols, will be able to provide the physical IoT devices to an higher level.

3.3 Fog-Enabled Architecture for BLE-Based Discovery

This section, after a brief introduction aimed to contextualize the presented fog-middleware architecture, first introduce the distributed architecture, then presents and explains an adaptive strategy for the proposed environment. For the sake of clarity, for proposed environment, is meant the Fog-Enabled architecture for IoT and the smart MQTT-driven device discovery protocols introduced in section 3.2.3. The introduction of fog paradigm in the architecture has the main crucial role of leveraging all the IoT needs. Indeed, fog middleware is able to support IoT and IoT devices with computing resources, communications protocol, location awareness, mobility support, low latency, and geo-distribution. In addition, the exploiting of fog paradigm enhances the IoT user experience. The fog middleware architecture has a key importance also in supporting the node discovery process. Indeed, through the fog middleware is possible to drastically improve the IoT device discoverability and IoT nodes sustainability. By exploiting the location awareness provided by fog nodes, it is possible to make the IoT nodes aware of the presence of each other in the proximity. In this way, the discovery process can start only when there effectively are two nodes in the proximity. With this kind of fog support, the discoverability and sustainability of IoT nodes will be increased. Next sub section presents an architecture and a new interaction model aimed to heavily improve the BLE device discoverability and to apply a better energy management. The model is based on introducing fog computing paradigm into the traditional BLE discovery scenario. The idea is to provide an IoT-Fog architecture ables of exploiting the main features provided by fog computing such as geo-distribution, location awareness, mobility support and so on, and applying them to IoT level in order to optimize the devices' discoverability and reduce the power consumption at the same time.

3.3.1 Distributed Architecture

The architecture I am proposing is composed by three entities: BLE-S, BLE-A and Fog Node (FN). Fog nodes have been introduced in the model, unlike the convention scenario, to exploit advantages and quality provided by the fog paradigm to improve the power efficiency of the BLE discovery process and optimize the devices' discoverability. Via FNs, I target the discovery synchronization, in other words, I want to make the BLE-Ss and the BLE-As aware of when they are in proximity of each other, synchronizing the advertising and the scanning process. In this way, the BLE-A and the BLE-S start advertising and scanning only when they are in their proximity, saving power and optimizing the device discoverability. The architectural model I am going to refer is depicted in fig. 3.7. I assume the BLE-Ss are located in a fixed and static locations, their goal is to discover BLE-As in the most efficient way in terms of power consumption. The BLE-Ss are not connected to any power supply. The BLE-As do not have a static and known position, they are free to move. I suppose the BLE-As to be carried by pedestrians. Their goal is to be discovered and to receive information by the BLE-Ss. The BLE-Ss and BLE-As are smart devices like smartphones. These devices have internet connectivity and a BLE interface. The FNs are smart entities connected to the infrastructure. At the beginning, the FNs load on BLE-As all the areas in where there are BLE-Ss providing services of interest. For example, an advertiser might be interested in some particular commercial advertising, food or drink spots, some touristic attractions, therefore any type of location-based service. In this way, only a restricted set of BLE-Ss are loaded to the BLE-A and it will not be involved in the discovery process by each nearby BLE-S. When a BLE-A enters in one of the pre-fetched areas of interest (green areas in fig.3.7), it starts sending its location to the FNs. In this way, the FNs become aware about BLE-As position and, when they enters in the BLE discovery range of a BLE-S (blue areas in fig.3.7), the FN alerts that specific BLE-S and makes the discovery process synchronized and starting. When a BLE-S is alerted by a FN, it switches the BLE interface ON and begins the scanning process, at the same time, the interested BLE-A begins the advertising process, in a fully synchronized manner. In this way, with this model and architecture, the BLE device discoverability is optimized, and, activating BLE interface and discovery process only when it is strictly needed, the power consumption is also reduced, saving power.

The model exploits the awareness given by fog paradigm in order to optimize the devices' discoverability and minimize the scanners power consumption at the same time. The model depicted in fig. 3.7 describes a three entities architecture composed by BLE-Ss, BLE-As, and FNs. The communications between the FNs and the BLE-Ss and that one between the BLE-As and FNs are via internet connectivity using the MQTT protocol. Further, WiFi is used and referred as internet connectivity . In the model there two factors of power consumption, the BLE discovery process and the internet connectivity. Firstly, I focus on the biggest of these two factor in terms of power consumption, the BLE discovery process. At the first stage, I keep the internet connectivity of the scanner active, focusing on saving the power consumption due to BLE usage. The BLE interface of BLE-Ss is kept

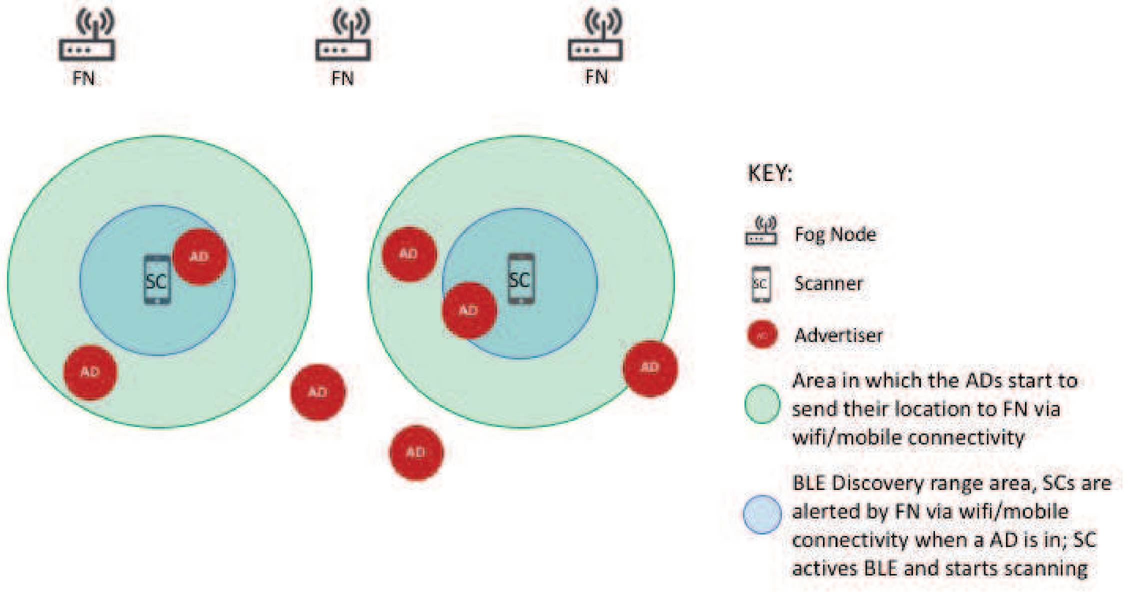


Figure 3.7: A model IoT-Fog Distributed Architecture

.

off at the beginning. In the proposed model, the BLE-As are in permanent movement, as soon as an BLE-A enters in the BLE-S's proximity area (green area), it starts sending its location and its UUID to the FN via MQTT, making the FN aware about the position of the advertiser and its ID. When the BLE-A gets closer to BLE-S, in the BLE discovery range (the blue area), the FN alerts the relative BLE-S that switches on the BLE interface and starts scanning for that specific BLE-A. The FNs, being aware of the location of the BLE-Ss and BLE-As, alert and wake up the BLE-S only when the BLE-A is in the BLE range. This synchronization makes the device discoverability guaranteed, minimizing the active time of the BLE interface. Once the BLE-A is discovered, the BLE interface will be switched OFF. It easy to understand that, according with the approach aforementioned, the switching ON/OFF of the BLE interface is strictly dependent by the arrival of BLE-As. This kind of interaction protocol and an its improvement have already been described as Power Efficient Node Discovery (PEND) and Smart PEND (SPEND), in the previous section, 3.2.3. The aforementioned protocols bring a remarkable improvement, in terms of power efficiency, of the conventional scenario in which the fog paradigm is not involved. The protocols face the interaction D2D, or rather between BLE-A and BLE-S. This section focuses the attention on a new strategy for the cost optimization, aimed to reduce even more the power consumption. Furthermore, the model is made even more power efficient by addressing the second main factor of the power consumption, the BLE-Ss' internet connectivity always active. In the model already presented the BLE interface usage is reduced but the internet connectivity is always active. In order to improve the power saving, this aspect is faced by reducing also the internet connectivity usage. The problem is addressed by keeping the internet interface OFF at beginning, and setting a timer on BLE-Ss. When the timer triggers the BLE-S connects to the FNs, and if there are BLE-As in the proximity, it activates the BLE interface a starts scanning for devices. The FN, being aware of the BLE-As' location and their speed, sets up the next timer value.

The disadvantage of this kind of approach is that the model might not guarantee the discoverability of a BLE-A. Indeed, with a bad setting of the wake-up timer, or with a sudden direction or speed change, a BLE-A might pass through the BLE-S's area without being discovered. This kind of approach paves the way to several challenges, for instance, what is the minimum value of the timer that guarantee the discoverability of the BLE-As? And then, What is the best trade-off between losing some BLE-As and saving more power?
In the next section a deep analysis regarding a switching on/off strategy to minimize the power consumption for the first stage of the model will be presented. Furthermore, the calculation of the minimum wakeup timer of the BLE-Ss for avoiding the advertisers losses, is faced.

3.3.2 Adaptive BLE switching on/off strategy for energy saving

The first factor of power consumption of the proposed architectural model is the energy consumed by BLE activities. This aspect is faced by switching on/off of the BLE-Ss' BLE interface accordingly with the arrival of the BLE-As. It is easy to understand that this approach is strongly dependent by the BLE-As' arrival frequency. If this frequency significantly grows, a BLE-S might enter in a continuous BLE switching on/off status. This kind of status results to be more power consuming than keeping the BLE always active (baseline scenario). With a high arrival frequency of the BLE-As might be more power efficient to keep the BLE interface of the BLE-Ss always active. This falls in a balance dilemma between switching continuously on/off the BLE interface and adopting the baseline strategy, keeping the interface always active. The balance dilemma just described belongs to class of problems of choosing between paying a periodic cost (rent a pair of skis) or paying a bigger cost just once (buying price). This class of problems is known as 'Ski Rental problem' [135]. The best off-line deterministic strategy is paying the rental cost until the accumulated expense does not reach the buying cost, then, it is better paying the buying cost. In the presented model, it means that the optimal strategy in term of power saving is to adopt the strategy of switching on/off the BLE interface of the scanners at every BLE-A's arrival until the power consumption due to continuous switching becomes equal to keep the interface active. The problem is addressed only in terms of power consumption, assuming the scanning frequency of the baseline scenario high enough to avoid the device discoverability losses. This strategy belongs to the class of two-competitive algorithms in which the optimum is given by the ratio between the buying and the rental price. More formally, the proposed analysis is based on two metrics; Energy of SWitching (ESW) is the energy consumed for switching on the BLE interface, the energy required for the scanning process, and the energy required to switch off the interface. The sum of these three elements has to be repeated each time a BLE-A enters in the BLE discoverability range, plus the energy consumed by keeping the internet connectivity active and the MQTT activities. ESW is namely the repeating cost of renting. The Energy UP (EUP) is the energy consumed by keeping the BLE interface always active with a periodic scan process. It is given by the sum of the energy consumed by switching on and off the interface just once, plus the energy consumed by the scanning process multiplied for the scanning frequency. The scanning frequency is supposed to be static and calculated over a finite period of time, i.e. one hour. EUP is namely the buying cost of the ski. Formally the two metrics are expressed in the equation 3.2a and 3.2b

$$ESW = e_{wifi} + \sum_{i=1}^{N} (e_{ONi} + e_{OFFi} + e_{SCANi}) * fr_{ARRi} * \delta_i \tag{3.2a}$$

$$EUP = e_{ON} + e_{OFF} + e_{MAN} + (e_{SCAN} * fr_{SCAN}) \tag{3.2b}$$

Where N is total number of BLE-As in the system, e_{ON} and e_{OFF} are the energies required to activate and deactivate the BLE interface, respectively. The e_{MAN} is the energy required to keep the interface active, while e_{SCAN} represents the energy consumed by the scanning process. Furthermore, let e_{wifi} be the energy consumed by keeping the internet connectivity always active and the MQTT activities. Then, fr_{SCAN} and fr_{ARR} are respectively the scanning frequency of a BLE-S and the arrival frequency of the BLE-As. While δ_i is a boolean variable, it is 1, if the i_{th} BLE-A is in the BLE discovery range of a BLE-S, 0 otherwise. Once we have defined the aforementioned metrics, we can obtain the the optimum of the balance dilemma. The optimum, according with the deterministic two competitive strategy for this class of problem is given by the division of EUP over ESW. The optimum of the dilemma means the maximum number switching on/off of the BLE interface before to keep the interface always active becomes more power efficient. The optimum OPT is calculated with the formula 3.3

$$\begin{aligned} OPT &= \frac{EUP}{ESW} \\ &= \frac{e_{ON} + e_{OFF} + e_{MAN} + (e_{SCAN} * fr_{SCAN})}{e_{wifi} + \sum_{i=1}^{N} (e_{ONi} + e_{OFFi} + e_{SCANi}) * fr_{ARRi} * \delta_i} \end{aligned} \tag{3.3a}$$

considering the worst case in which every BLE-A enters in the BLE discoverability range and defining e_{SW} as (e_{ON} + e_{OFF}), equation 3.3 becomes:

$$= \frac{e_{SW} + e_{MAN} + (e_{SCAN} * fr_{SCAN})}{e_{wifi} + [(e_{SW} + e_{SCAN}) * N * fr_{ARR}]} \tag{3.3b}$$

As already stated, considering a finite period of time T, fr_{SCAN} is static and fixed. It makes the calculation of the optimum totally depended by fr_{ARR}.

The second power consumption factor of the system is given by the internet connectivity always active. This aspect is faced by turning the internet connectivity of the BLE-Ss to dormant state for a certain period of time. The timer that wakes up the connectivity of the BLE-Ss is driven by FNs. The FNs are aware about BLE-As movements, speed, and frequency. Given all those information, the FN is capable to efficiently set the timer on the BLE-Ss. The problem regarding the calculation of the minimum timeout that guarantees the discoverability of all BLE-As, is faced. I assume that, the BLE-Ss are placed on points of interest, and they have a circular area of radius r as BLE discoverability range. Moreover, as already stated, the BLE-As are devices carried by pedestrians with a certain speed($V_{BLE\text{-}A}$) and an arrival frequency(fr_{ARR}). It is reasonable to state that the minimum wakeup period of the BLE-Ss must be less than the BLE-A's arrival period plus the time the BLE-A takes to entirely across the discoverability area of the BLE-S(T_{AREA}). Formally this relation is expressed in the formula 3.4 :

$$
\begin{aligned}
T_{WAKEUP} &< T_{ARR} + T_{AREA} \\
T_{AREA} &= \frac{2 * r}{V_{BLE-A}}
\end{aligned}
\tag{3.4a}
$$

$$
=> T_{WAKEUP} < T_{ARR} + \frac{2 * r}{V_{BLE-A}} \tag{3.4b}
$$

$$
=> fr_{WAKEUP} > \frac{1}{T_{ARR} + \frac{2*r}{V_{BLE-A}}} \tag{3.4c}
$$

Assuming the speed of the BLE-As known and constant, the fr_{WAKEUP} becomes dependent by fr_{ARR}. The FNs are aware of BLE-A's speed and the radius of the area, and according with the historical of the arrival frequency of the BLE-As, the FNs are capable to estimate and set the wakeup timeout on the BLE-Ss.
In the next section the experimental results will be presented. I run tests and experiments targeting the proposed protocols and architecture under several and different conditions. Each sub section will be formed by two parts, the first will address the general settings of the experiments, while the second one will present the experiment results along their discussion. Furthermore, the next section will provide the application of the adaptive BLE switching on/off strategy and will present a new version of PEND protocol.

3.4 Experimental Results

In this section, the performance study of the proposed MQTT-driven node discovery solutions are presented. The protocols have been tested under many conditions, cases, and Key Performance Indicators (KPIs). These condition can be broken down in three components:

Scanner Sliding Window The first broken down component of the aforementioned conditions is the BLE-S's sliding window. The protocols have been tested under different BLE-S's scanning frame lengths

Advertiser Dinamicity In the second component, PEND and SPEND have been tested considering the arrival of BLE-As dynamic

Dinamicity on Sliding Window The third component is the combination of the previous two. The two proposed protocols have been tested measuring their performances considering the impact of BLE-A's dinamicity on the varying of the BLE-S's scanning frame length

Under all the conditions presented above the performance of the two proposed protocols is compared with that one of benchmark, traditional scenario. Each of these three components are presented in a different sub section, and each sub section is composed by two parts. The first part address the experiment setting along a brief description, while the second part effectively presents the experimental results along their discussion discussion. Finally, this section presents the application of the adaptive switching on/off strategy described in 3.3.2, and it also introduces a new version of PEND protocol along its performance and discussion.

3.4.1 Scanner Sliding Window

This sub section presents the experiments run on PEND and SPEND by varying the length of the BLE-S scanning window. Table 3.1 summarizes the experimental settings in detail. Two smartphones have been used for emulating the BLE-A and BLE-S in the environment, while a personal computer running the Mosquitto server to serve as the MQTT broker at the fog level. Implementing the MQTT broker as a fog layer entity has already been shown as a feasible solution in relaying the communication between IoT nodes and the cloud platforms, where data processing and storage take place [136]. The smartphones run Android 6.0.1 Marshmallow version and Android API level 23. The MQTT client has been implemented using two libraries, Eclipse Paho Client version 3.1.1, and Eclipse Paho service 1.1. These provide the API implementing the MQTT protocol and allow the Android applications to interface with the MQTT brokers. The computer that implements the fog level, runs an MQTT broker based on Mosquitto server and creates a private WiFi network for the smartphones [137]. Android Debug Bridge (ADB) has been used to fetch the battery statistics [138].

The experiments have been run under the following three schemes. 1) Conventional node discovery which only involves Bluetooth-based discovery of the BLE-A by the BLE-S. In order to investigate the impact of MQTT-driven sustainable node discovery, traditional discovery scenario has been used as benchmark, 2) PEND, which is MQTT-triggered and involves WiFi connection, 3) SPEND, as the faster and more sustainable version of PEND. The experiments aim at studying the different behavior and the variation in terms of Power Consumption of the three discovery mechanisms and the two new protocols PEND and SPEND. In order to investigate the impact of the BLE-S frame length, the scanning frame length has been varied from 10 seconds to 1 minute. SPEND stops scanning as soon as a device match has been accomplished; hence the frame length does not affect SPEND since the entire duration of the frame length is never reached. Therefore, any experiment under identical settings but different frame lengths will result in the same device matching and consequently the same level of battery drain by all components.

For each data shown in the following charts the same experiment has been run three times, and the value in the charts is the result of the average of the three runs. Each run is one hour long. The results and the trends might be more clear and stable with a larger number of runs.

Experimental Results

In this subsection, the experimental results of PEND and SPEND with different BLE-S scanning frame lengths, are presented. As mentioned in the above section, the evaluation of the BLE-S performances is based on two key performance indicators (KPI), measured under the three approaches. The KPIs are : DMR, and power consumption. The goals of the two novel proposed algorithms are to maximize the DMR and to reduce the power consumption. The power consumption KPI, can be broken down in three different performances of battery drain in order to have a fully and complete overview about the battery consumption. The study and the analysis of different aspects of the battery

Table 3.1: Experimental settings. First part of the table presents general experimental settings. Second part presents discovery scheme-specific settings

Parameter	**Value**
Node Discovery Schemes	Conventional, PEND, SPEND
BLE-A and BLE-S operating systems	Android 6.0.1 Marshmallow
Number of experiments per scenario	6
Total number of runs per experiment	3
Duration of a single run	1 hour
BLE-S Scanning Frame Length	$\{10, 20, 30, 40, 50, 60\}$ sec.
BLE-A Advertising Frame Length (ℓ)	30 s
BLE-A Inter-arrival duration	60 sec.
MQTT Broker Type	Mosquitto Server
BLE Activity in Conventional Discovery	Always Active
BLE Activity in PEND	MQTT Broker-Triggered
BLE Activity in SPEND	MQTT Broker-Triggered
WiFi Activity in Conventional Discovery	OFF
WiFi Activity in PEND	Always ON
WiFi Activity in SPEND	Always ON
WiFi Activity of MQTT Broker	Always ON
BLE-S Scanning Frame Rate in Conventional Discovery	1 every 30 sec.
BLE-S Scanning Frame Rate in PEND/SPEND	On demand

drained is performed on the scanner during the discovery phase. The power drain has been evaluated on the following components for the BLE-S: 1) Battery drained by the BLE interface of the BLE-S, 2) Battery drained by the CPU due to BLE operations, 3) Battery drained by the node discovery application.

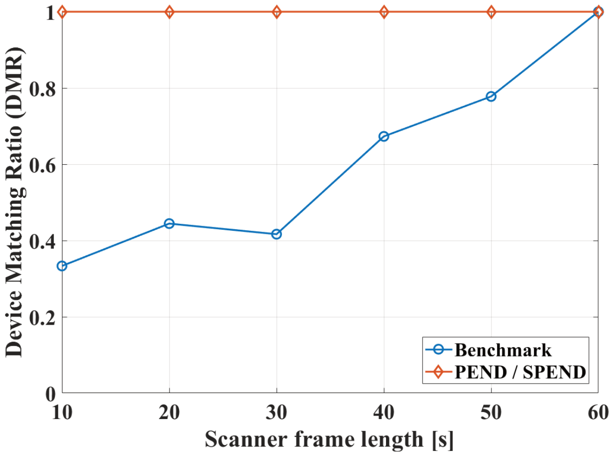


Figure 3.8: Device Matching Ratio (DMR) under the benchmark scheme and the proposed schemes, PEND and SPEND. The larger the frame length of the BLE-S, the higher the probability of finding the BLE-S in the active scanning mode when a new BLE-A approaches. On the other hand, the DMR of PEND and SPEND is unaffected by frame length cause the two proposed protocol are MQTT-triggered, hence the DMR is constantly 100 per cent; The BLE-A discovery is always guaranteed regardless of the scanning frame length.

Fig. 3.8 presents the DMR behavior under varying BLE-S frame length. As shown in the figure, the benchmark improves the DMR as the length of the BLE-S scanning frame increases. The experiment is run by keeping constant the BLE-A arrival rate. The experimental results show how the DMR performance of conventional locality-driven BLE-based node discovery increases with the enlarging of the scanning frame length, indeed it is approximately 30% under a frame length of 10 seconds, and it grows up to 100% as the length of the BLE-S frame reaches 60 seconds. It is worthy to notice that these results are strictly dependent by the defined settings depicted in table 3.1. Indeed with a different BLE-A arrival rate the DMR performance under conventional discovery protocol would change. Under the conventional discovery process, unless with a scanning frame always active, the 100% of DMR might not be ever reached. This behavior is expected because the probability that a new BLE-A approaches during the BLE-S is in active scanning mode is higher if larger is the scanning frame length. On the other hand, the discovery under PEND and SPEND schemes is always guaranteed, with a DMR constantly at 100%; this because PEND and SPEND are not affected by scanning frame length. The reason is as follows: In PEND and SPEND, the activation of the scanning mode of BLE-S is triggered by the MQTT broker running on the fog node. It occurs only in case a new BLE-A is nearby and effectively discoverable. Hence, the scanning mode of BLE-S is activated by the MQTT message sent by FN, and it makes the scanner independent by the BLE-A arrival rate and scanning frame length.

Fig. 3.9 depicts the first broken down component of the battery drain. This chart shows the battery consumption of the BLE-S due to BLE utilization. Before analyzing the behav-

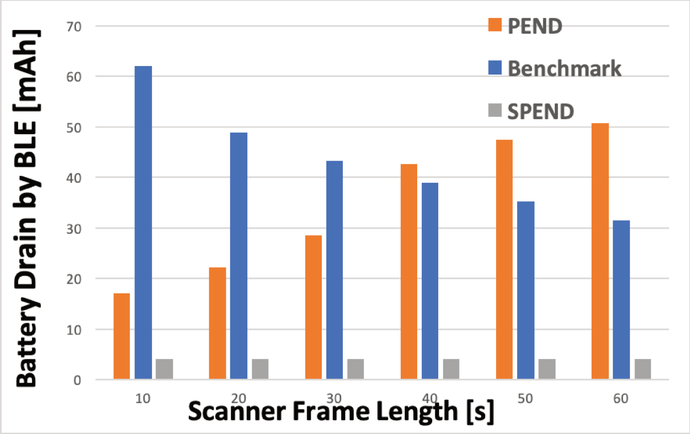


Figure 3.9: Battery drained by the BLE interface of the BLE-S under the benchmark scheme and the proposed PEND and SPEND schemes. The experiment is run for a finite period of time, hence under larger BLE-S scanning frames, the total number of frames decreases. Under benchmark the battery drain decreases as the frame length increases. While, on the other hand, the battery is drained less by the BLE when the scanning frame length of the BLE-S is no larger than 30 seconds, under the proposed PEND scheme. The battery consumption due BLE utilization under the proposed SPEND protocol is not affected by the length of the frames or by the total number of the frames.

iors under the three approaches, it is worthy to be emphasized that the SPEND scheme is not affected by the varying length of the BLE-S scanning frame. The experiments have been run in a fixed one hour duration. Hence, the enlarging of the scanning windows results in the reduction of the total number of the frames under conventional discovery scheme. While, under PEND and SPEND, the number of the scanning frame is strictly dependent by the BLE-A's arrivals. Benchmark approach results to have an higher power consumption under shorter scanning frame length, i.e., 10-30 seconds, while the battery drain of the scanner due to BLE utilization decreases as the frame length gets larger. According with what I stated above, with larger BLE-S scanning frames, the total number of frames decreases. Hence, the Benchmark approach results to drain more energy due to BLE interface with a shorter scanning frame length and an higher number of frames. Under the proposed PEND scheme, the power consumption due to BLE is less when the scanning frame length of the BLE-S is shorter than 40 seconds. Therefore, it is reasonable to state that the MQTT-driven device discovery helps the BLE-S to save energy in the case of short scanning frames. This is reasonable because in PEND, the activation of scanning process is tightly related to the arrival of the BLE-A devices. When the frame length is large and the BLE-A arrival rate is high, PEND tends to keep the scanning process frequently active. However, under PEND, the scanner is aware of the approaching of a new nearby device, and it is activate only when the BLE-A is within the scanner discovery range. Under these considerations, having a large scanning frame length might result to be unnecessary. The battery consumption due to BLE utilization under the proposed SPEND scheme is not affected by the length of the frames or by the total number of the frames. The reason of this behavior is that SPEND shuts the scanning windows and the BLE interface down as soon as the BLE-S has discovered the specific advertiser, therefore, under SPEND protocol, the scanner results to have a very short discovery process. This allows to keep the battery consumption significantly low, i.e., approximately

60%-80% of the battery drained by BLE under the conventional.

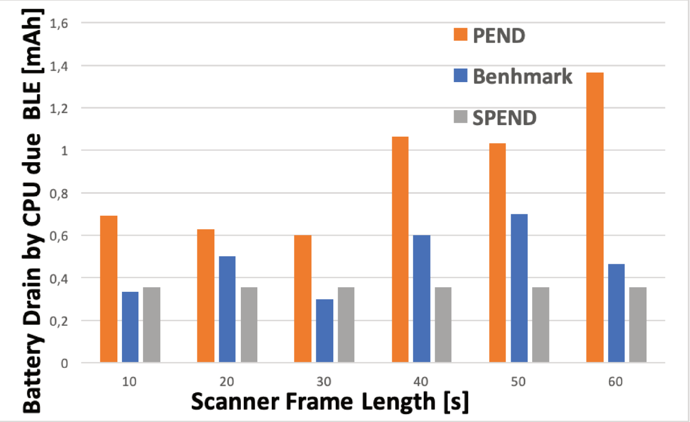


Figure 3.10: Battery drained by the CPU due to BLE activities under the benchmark scheme and the proposed PEND / SPEND schemes. The ON/OFF switching operation adds an overhead in CPU usage and the power drained by the CPU due to BLE utilization increases. SPEND smooths this aspect taking advantage of its very short scanning frame, being able to cope with the CPU usage overhead.

The Fig. 3.10 depicts the second second broken down battery drain component, the battery drained by the CPU due to BLE operations. It is obvious that the battery consumed by the CPU due to BLE utilization is lower under the conventional approach when compared to PEND. This phenomenon is reasonable because PEND and SPEND turn the BLE interface ON/OFF upon each advertiser's arrival. These switching ON/OFF operations of the BLE interface introduce a CPU overhead that impacts to the power consumption. Indeed, controlling switching ON/OFF the BLE interface upon each BLE-A arrival increases CPU usage, and consequently the power drained by the CPU due to BLE utilization increases. In the conventional scenario, the BLE interface of BLE-S is always active and the scanning process is run periodically; hence there is no CPU usage overhead to manage and control the switching operations of the BLE interface. As I stated above, under PEND and SPEND, the switching ON/OFF operations are dependent by the BLE-A arrivals, hence the CPU overhead due to these operations is strictly dependent by the advertiser's arrival rate. Consequently, the varying of the BLE-A arrival rate leads to an higher or lower battery drained by the CPU due to BLE operations. SPEND smooths the CPU usage overhead by taking advantage of the extremely short scanning frame so that the ON/OFF switching does not significantly impact the battery drained by CPU due to BLE usage. It is worth noting that the behavior of PEND in Fig. 3.10 and Fig. 3.11 is similar since the battery drained by the CPU due to BLE implicitly reflects the usage of the BLE interface and the frequency of the ON/OFF switching. The two main factors that impact on the power consumption are the effective scanning operations and, mainly, the operation of switching ON/OFF the BLE interface. Being the BLE interface always ON under benchmark, the CPU is not involved in controlling the switching ON/OFF operation as PEND and SPEND do, hence it leads to have a lower power drained by the CPU due to BLE operation in comparison to PEND.

The CPU battery usage of the BLE-S application is depicted in Fig. 3.11 as last broken down component of the battery consumption study. As the chart shows, the trends of battery drain in Benchmark and SPEND reflect their one in Fig. 3.10. The discovery

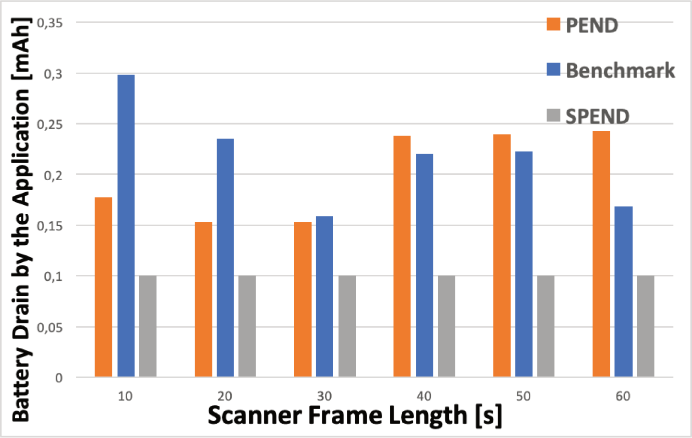


Figure 3.11: CPU battery drained by the application for device discovery under the conventional approach and the novel proposed PEND and SPEND protocols. Battery drain under SPEND is nearly 55%-62% of the battery drained by discovery application under the benchmark.

application drains nearly 55%-62% of battery less under SPEND scheme than the conventional discovery approach. However, the battery drain in PEND is lower than that one in Benchmark approach when the scanning window is not larger than 30 seconds. It is worthy to be underlined that the power consumption of the discovery application under PEND and SPEND also includes the battery drained by the application due to WiFi interface usage.

BLE Active PEND: A New Version

In this subsection we introduce a new version a PEND scheme along with a study of its performance in terms of battery consumption. I have introduced this new version in order to better investigate the impact of the BLE switching ON/OFF mechanism and trying to smooth its effect. As it is depicted Fig. 3.10, the controlling of switching ON/OFF of the BLE interface introduce a CPU overhead under PEND approach. This effects is surely dependent by the BLE-A arrival rate, indeed the higher is the arrival rate, higher is the number of switching ON/OFF of the BLE interface. With this new version of PEND, we study how these operations impact on PEND performance, while how this effect change on varying the BLE-A's arrival rate is presented in [139]. In order to achieve the set goal, in this new version of PEND, the BLE interface is kept always active. In this way, only the scanning process is triggered by FN via MQTT message, while the BLE interface is always ON. In this new version, the scanning process remains triggered by the FN, therefore the device discovery is still guaranteed. The experimental results of this new version of PEND are presented, since the DMR is still 100%, the results presented address only the power consumption aspects. Also here, the battery drain performance is presented through two broken down components, the power drained by the BLE activities and the CPU power drain due to BLE activities.

In Fig. 3.12 the first broken down metric of power consumption is depicted. In order to have the fairest comparison as possible, the same parameters setting of Fig 3.9 are the same, with just the introduction of the new version of PEND. Even in this new version, PEND is not affected by the total number of frames, but the activation of the scanning

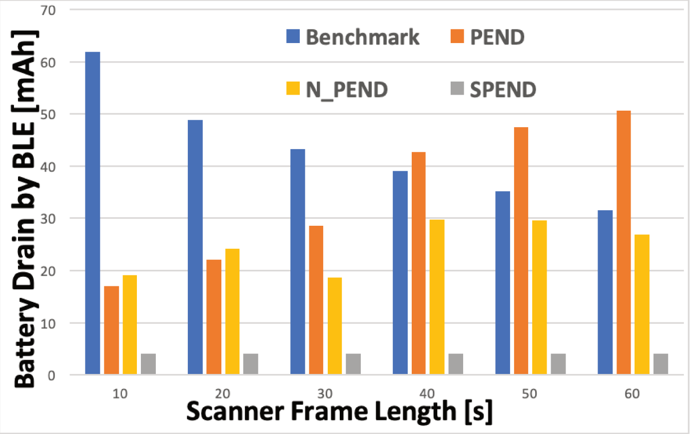


Figure 3.12: Battery drained by the BLE interface of the BLE-S under the three schemes already presented and the new version of PEND in addition. The general settings of the experiments are still the same of the previous one.

process is strictly related to BLE-A arrival rate. The power drain of the BLE interface is generally and significantly improved under new PEND. The chart highlights how the the switching ON/OFF mechanism of BLE interface has a remarkable impact on the power drain. The power consumed by the BLE activities under new PEND result to be better than the performance of conventional discovery under any frame length. This improvement is more remarkable under short frame length, i.e. 10-30 seconds. On the other hand, with a the larger frame lengths, 40-60 seconds, the new PEND version has a significant improvement compared to the original version of PEND. The new proposed version of PEND results to have a really better performance in battery consumption due to BLE activities exactly where the original PEND has its worst.

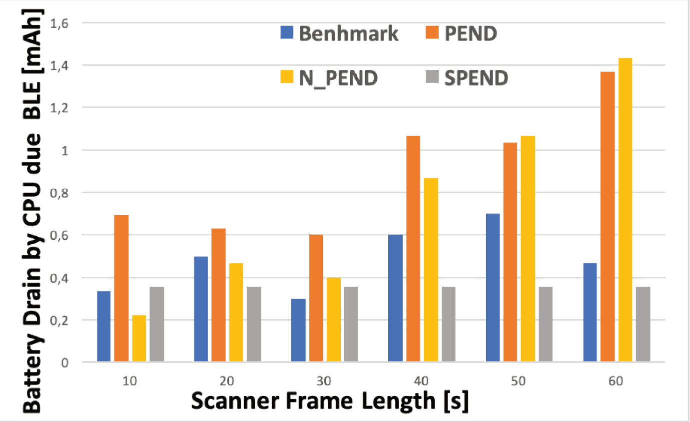


Figure 3.13: Battery drained by the BLE interface of the BLE-S under the three schemes already presented and the new version of PEND in addition. The general settings of the experiments are still the same of the previous one.

Finally, the last broken down component of power consumption depicts the battery drained by the CPU due to BLE activities, Fig. 3.13. In this figure is compared the performance of the above mentioned metric of the new version of PEND with the all others three schemes already depicted in fig. 3.10. With the short frame lengths, the new version of PEND has very good performance in terms of CPU power drain. The level of battery drained by the CPU in relation of BLE activities of new PEND is greatly lower than its original version, slightly lower than the conventional discovery approach, and absolutely comparable with SPEND protocol. According to that, it is reasonable to state that the CPU overhead due to the switching ON/OFF mechanism is significantly reduced. The performance of new PEND are totally comparable with SPEND, the best algorithm in terms of power saving with 100% DMR. It is worthy to remember that PEND does not take advantage from using a extremely short scanning window, contrary to what SPEND does. Under the larger frame lengths, the performance of new PEND degenerates, becoming comparable to its original version. This is reasonable and expectable because with the larger scanning windows, i.e. 40-60 seconds, the component of CPU power consumption due to the BLE scanning cycles prevails on the component due to the interface switching ON/OFF. This because, within a larger scanning window, there are more scanning cycles compared to a shorter one. The power drained by the CPU due to BLE activities is mainly affected by two components, the scanning cycles and the switching ON/OFF of the BLE interface. With the shorter scanning windows the switching component impacts for the most part to CPU power consumption, while on the other hand, with larger scanning windows the predominating component of the CPU power consumption is the scanning cycles.

3.4.2 The Impact of Advertiser Dynamicity

This section presents a thorough investigation of the two MQTT-driven node discovery schemes already presented under the impact of advertisers' dynamic arrival. As it has done in the previous section, it follows a brief presentation of the experimental settings, and then the effectively presentation of the experimental results. In this case, the study of the impact of the dynamic BLE-A arrival patterns on the DMR and power consumption of PEND and SPEND has been addressed.

As the first set of experiments, to emulate the BLE-A and BLE-S have been used two smartphones (Android 6.0.1 Marshmallow version and Android API level 23). The MQTT broker in the fog layer has been run in laptop, with an operating Mosquitto server. Over the two applications, Eclipse Paho Client 3.1.1 and Eclipse Paho service 1.1 provide the API for the MQTT protocol implementation and facilitate an interface between Android applications and MQTT broker.

The objective of the experiments is to investigate the impact of dynamic arrival patterns of BLE-A under fixed scanning frame length for the BLE-S. Primarily, the focus has gone on DMR, then on Power Consumption. As mentioned earlier, in order to study the impact of the dynamic arrival of BLE-A nodes, I fixed the BLE-S frame length to 30 seconds and I scheduled the arrival rates of the BLE-A according to the Poisson Distribution with average arrival rate of λ. Thus, the inter-arrival times would follow negative exponential distribution with the mean $\beta = 1/\lambda$. In the experiments, we vary the inter-arrival time (β) within the following set {30s, 60s, 90s, 120s}. As the previous section, I present the performance of the approaches Benchmark, since now on Baseline, PEND, and SPEND. Regardless of the scanning frame length, SPEND deactivates the scanning frame immediately upon the detection of a device match. Therefore, SPEND is not affected by the frame length in the experiments. That being said, any two subsequent experiments with identical settings except the scanning frame length will output the same DMR results, as well as the same battery drain under the SPEND discovery regime. Most of the settings are still the same of the previous set of experiments, however, in table 3.2 are summarized the discovery scheme-specific settings.

Table 3.2: General and discovery strategy-specific settings in the experiments.

Parameter	**Value**
Node Discovery Schemes	Baseline, PEND, SPEND
BLE-A and BLE-S operating systems	Android 6.0.1 Marshmallow
Number of experiments per scenario	4
Total number of runs per experiment	3
Duration of a single run	1 hour
BLE-S Scanning Frame Length	30 sec.
BLE-A Advertising Frame Length (ℓ)	30 sec.
BLE-A Inter-arrival duration	scheduled according to Poisson Distribution
Inter-arrival time β	$1/\lambda$
(λ) values for Poisson Distribution	$\{30, 60, 90, 120\}$ sec.
MQTT Broker Type	Mosquitto Server
BLE Activity in Conventional Discovery	Always Active
BLE Activity in PEND	MQTT Broker-Triggered
BLE Activity in SPEND	MQTT Broker-Triggered
WiFi Activity in Conventional Discovery	OFF
WiFi Activity in PEND	Always ON
WiFi Activity in SPEND	Always ON
WiFi Activity of MQTT Broker	Always ON
BLE-S Scanning Frame Rate in Conventional Discovery	1 every 30 sec.
BLE-S Scanning Frame Rate in PEND/SPEND	On demand

Experimental Results

All performance results in this subsection are presented in terms of two performance metrics, namely the DMR, and power consumption. Also in this experiment, the battery drained by the BLE-S during the discovery phase can be broken down into the following components: 1) Battery energy consumption of the BLE interface of the BLE-S, 2) Battery energy consumption of the CPU as a result of BLE-initiated processes, and 3) Battery energy consumption of the discovery application.

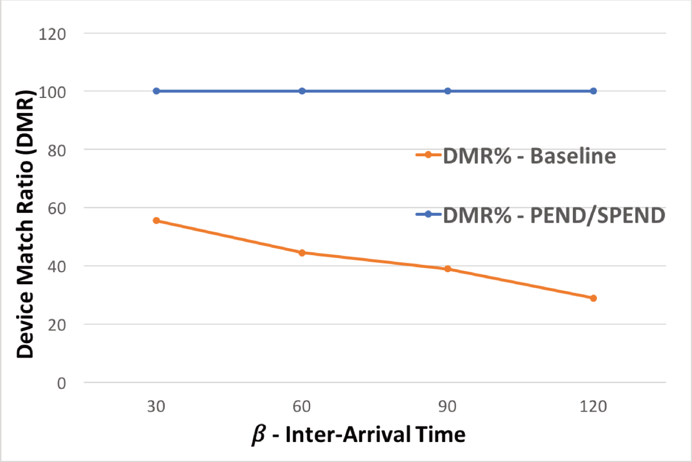


Figure 3.14: Device Matching Ratio (DMR) under the baseline scheme and the PEND and SPEND schemes. The lower the inter-arrival time (β), the higher the probability of finding the BLE-S in the active scanning mode when a new BLE-A approaches. The results also confirm that, the PEND and SPEND can ensure 100 per cent DMR; which is an "always guaranteed" solution for the BLE-A discovery. Further, the DMR under PEND and SPEND is not affected by the arrival pattern of the BLE-A devices.

In fig. 3.14, the DMR behavior is presented under varying arrival rates (i.e. inter-arrival times, β) of BLE-A when the scanning frame length of the BLE-S is fixed to 30 seconds. As seen in the figure, the benchmark decreases the DMR as the inter-arrival time between BLE-As gets longer. The experimental results show that the *baseline* node discovery results in a DMR of approximately 55% under an average inter-arrival time of 30 seconds, and the DMR under the baseline almost constantly decreases down to approximately 30% as the inter-arrival time between BLE-As is extended towards 120s. Indeed, this behavior is expected for the following reasons. First, the larger the average BLE-A inter-arrival time, the higher the probability of missing the BLE-S in the active scanning mode when a new BLE-A approaches. Second, conforming with our previous study, the DMR under PEND and SPEND is constantly 100%; in other words, neither PEND nor SPEND is impacted by the arrival characteristics of the BLE-As, and they always ensure 100% DMR. Third, the transparency of the DMR performance of PEND and SPEND can be explained as follows: BLE-S enters the scanning mode upon receiving a WAKEUP message from the MQTT broker while the WAKEUP message is issued as soon as a BLE-A is discovered in the vicinity and is likely to be discovered effectively. Indeed, both SPEND and PEND might be affected under extremely short (i.e. unfeasible) advertising frame length of the BLE-A.

Fig. 3.15 presents the battery energy consumption of the BLE interface of the BLE-S.

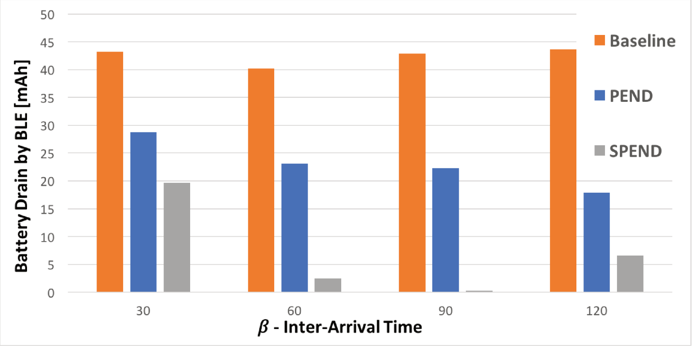


Figure 3.15: Battery drained by the BLE interface of BLE-S under the baseline solution, and the PEND - SPEND schemes. The baseline solution results in high battery consumption under all BLE-A inter-arrival times. On the flip side, as the duration between two BLE-A advertising frames gets longer, the battery consumption under PEND decreases. The trend of the battery drained under SPEND is similar to that under PEND.

The battery drain is tested under a varying inter arrival times from β=30 s up to β=120 s. It should be emphasized that the baseline scheme is not affected by the varying the inter-arrival time of the BLE-A advertising frame. Furthermore, the PEND-based discovery results in higher battery drain in the case of short inter-arrival of BLE-A advertising frames, i.e., β=30,60 s. However, the battery drained by the BLE interface of the BLE-S decreases as the BLE-As arrive less frequently, or under longer duration of BLE-A frames. Under shorter inter arrival times, BLE-A is spawned more frequently, which results in more frequent WAKEUP messages sent to the BLE-S. Hence, the battery drained by the BLE interface under PEND/SPEND regimes is inversely related to the dynamicity of the BLE-A arrivals. Under the baseline solution, battery energy consumption is significantly higher than PEND and SPEND. First of all, it is worth mentioning that the BLE-S BLE interface activation and scanning process are totally BLE-A arrival independent. Indeed, under the benchmark scheme, the BLE interface of BLE-S is always active and in the beaconing mode as opposed to being activated by MQTT Broker. Hence, appointment of the MQTT Broker as the initiating source of device discovery results in significant energy savings in the BLE-Ss under different arrival behavior of the BLE-A advertising frames.

In fig. 3.16, battery consumption of the CPU due to BLE utilization is presented. The CPU of the BLE-S consumes consumes more battery energy due to BLE utilization under PEND in comparison to the baseline solution. As I stated earlier, having the BLE interface always ON might be a wiser decision rather than turning it on/off upon the arrival of BLE-A advertising frames. The performance of PEND in this figure supports our initial hypothesis. Thus, the baseline solution does not need to control the activation/de-activation of the BLE interface by the CPU; therefore the CPU power consumption due to BLE utilization is lower under the baseline solution when compared to PEND. SPEND, due to introducing significantly short scanning frames, can cope with this phenomenon. Furthermore, as the average inter arrival duration between the BLE-A advertising frames gets longer, the BLE interface is activated/deactivated less frequently by the MQTT Broker (and consequently the CPU of the BLE-S), and the battery drained by the CPU gets significantly lower under longer inter-arrival times between the BLE-A advertising frames. In the last two figures,

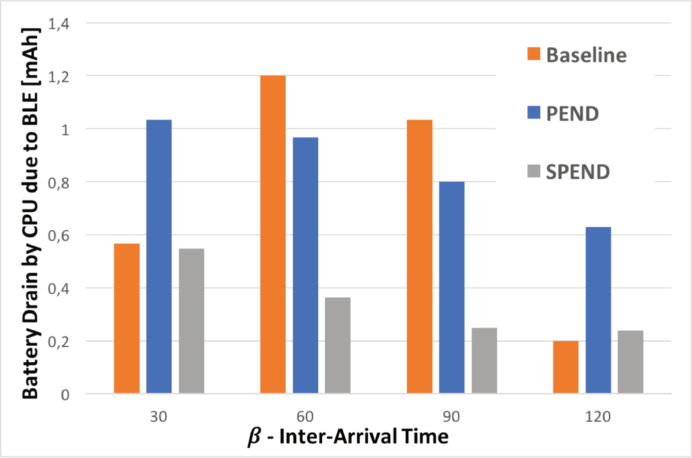


Figure 3.16: Battery energy consumption by the CPU due to the utilization of the BLE interface under the baseline solution and the proposed PEND / SPEND schemes. The MQTT-driven ON/OFF switching at the BLE-S results in increased CPU usage and consequently higher power consumption by the CPU due to BLE utilization. To cope with this, SPEND takes benefit of its "immediate shutdown of BLE upon discovery" principle so that the scanning frame length is also significantly reduced.

PEND and SPEND demonstrate similar trends with respect to the inter arrival times between the BLE-A advertising frames. In fact, this is an intuitive phenomenon since the CPU power drain due to controlling the BLE interface gets more frequent under shorter BLE-A inter-frame duration.

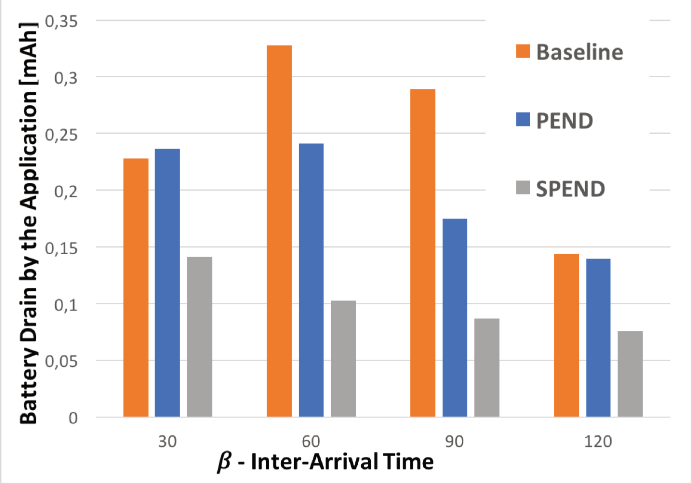


Figure 3.17: Battery energy consumption by the application for BLE-A discovery under the baseline solution and the PEND / SPEND schemes. Battery energy consumption under SPEND is roughly 50%-60% of the battery energy consumption under the baseline solution.

Finally, the CPU battery energy consumption of the BLE-S application is presented. As seen in fig. 3.17. Battery energy consumption characteristics of the baseline and SPEND schemes follow the same behavior in fig. 3.16. Battery energy consumption under SPEND is roughly 50%-60% of that under the baseline solution. However, energy consumption of the BLE-S application under PEND is close to that of the baseline when inter-arrival time between consecutive BLE-A frames is equal to 30 and 120 seconds. The reason of this behavior can be explained as follows: PEND consumes more energy due to activating/deactivating the BLE interface rather than the number of scanning frames as the scanning process is triggered/initiated by the MQTT Broker. On the other hand, under baseline solution, the energy consumed for activation/deactivation of the BLE interface is significantly low whereas the battery is drained due to the higher number of scanning frames. It is worth mentioning that the energy consumption of the BLE-A discovery application under PEND and SPEND also includes the energy consumed due to the utilization of the WiFi interface.

3.4.3 The impact of Advertiser Dynamicity on Scanner Sliding Window

The crucial importance of BLE-As dynamic arrival on the protocols and model has been already highlighted along the whole set of experiments addressed so far. In this subsection I present a set of new experiment runs born by the combination of the impact of dynamic arrival of advertisers and the different lengths of the scanning frame. Following the same pattern, I firstly lay the focus of the dynamic arrival on the Device Matching Ratio (DMR) and, then, on the power consumption. As the above sets of experiments, I study the performance of Baseline, PEND and SPEND schemes by varying the advertiser arrival rate according the Poisson distribution, over three different scanning frame length. I scheduled the BLE-A's arrival with an average rate of λ. Hence, the inter-arrival times between two consecutive advertisers would follow the negative exponential distribution with the mean $\beta = 1/\lambda$, as I already have explained in the above sub section. In the experiments, the inter-arrival time (β) has been varied within the following set {30s, 60s, 90s, 120s}, while the scanning frame length within this other one {10s, 30s, 60s}. It is worthy to remind that, SPEND deactivates the scanning frame immediately upon the detection of a device match. Therefore, SPEND is not affected by the frame length in the experiments. All the settings of the experiments is listed into the table 3.3. According with the approach used in the previous sets of experiments, the performance study regarding the energy consumption is broken down into the same components: 1) Battery energy consumption of the BLE interface of the BLE-S, 2) Battery energy consumption of the CPU as a result of BLE-initiated processes, and 3) Battery energy consumption of the discovery application. Most of the settings are still the same of the previous sets of experiments, however, in table 3.3 are summarized all the used settings.

Experimental Results

The following charts show the trend of DMR under Baseline, PEND and SPEND by varying the frame length of the BLE-S. Fig. 3.18 depicts four different series, one for each β according the parameter listed in table 3.3.

In Fig. 3.18 is depicted the DMR behaviors of the three schemes under varying arrival rates (i.e. inter-arrival times, β) of BLE-A and the BLE-S scanning frame length. As seen in Fig. 3.18, the PEND/SPEND schemes are represented by only one line, this because the scanning process under these two approaches is triggered by the fog node any time a new BLE-A approaches. This policy guarantees the discoverability of the BLE-A, hence the DMR is 100% under any BLE-A arrival rate. On the other hand, in the Baseline scenario, the scanning process is constant and periodic, hence the DMR is

Table 3.3: Experiment Settings of this set

Parameter	**Value**
Node Discovery Schemes	Baseline, PEND, SPEND
BLE-A and BLE-S operating systems	Android 6.0.1 Marshmallow
Number of experiments per scenario	12
Total number of runs per experiment	3
Duration of a single run	1 hour
BLE-S Scanning Frame Length	$\{10, 30, 60\}$ sec.
BLE-A Advertising Frame Length (ℓ)	30 sec.
BLE-A Inter-arrival duration	scheduled according to Poisson Distribution
Inter-arrival time β	$1/\lambda$
(β) values for Poisson Distribution	$\{30, 60, 90, 120\}$ sec.
MQTT Broker Type	Mosquitto Server
BLE Activity in Conventional Discovery	Always Active
BLE Activity in PEND	MQTT Broker-Triggered
BLE Activity in SPEND	MQTT Broker-Triggered
WiFi Activity in Conventional Discovery	OFF
WiFi Activity in PEND	Always ON
WiFi Activity in SPEND	Always ON
WiFi Activity of MQTT Broker	Always ON
BLE-S Scanning Frame Rate in Conventional Discovery	1 every 30 sec.
BLE-S Scanning Frame Rate in PEND/SPEND	On demand

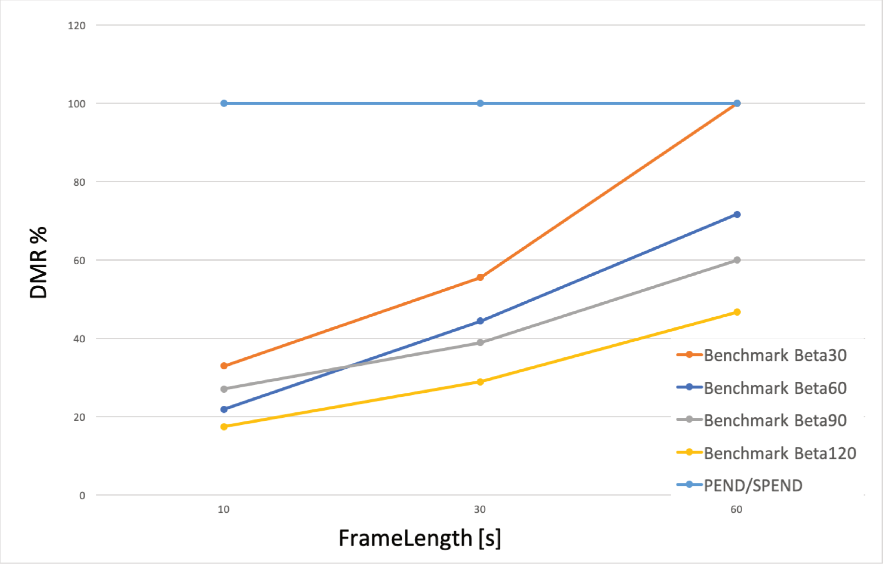
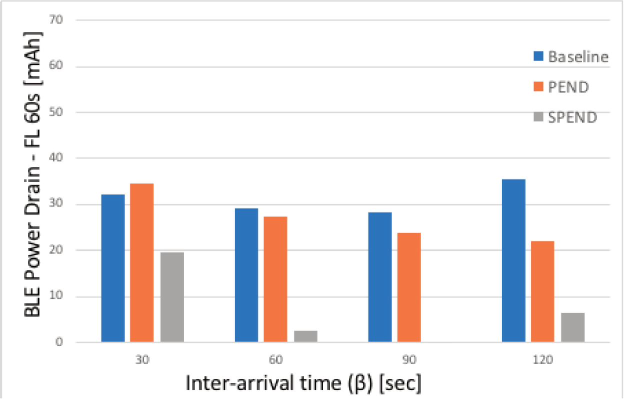


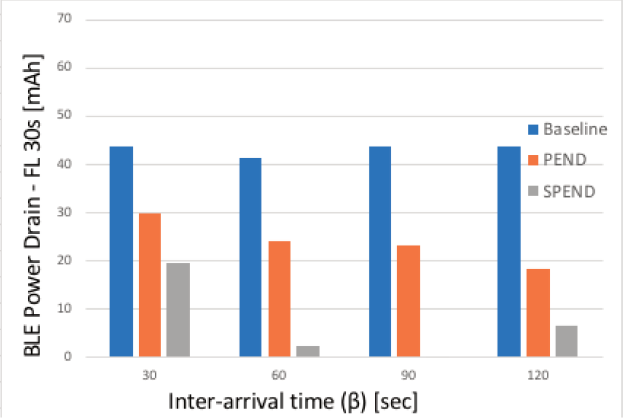
Figure 3.18: Device Match Ratio of Baseline, PEND and SPEND over the different BLE-S's frame length. The series depicted in the chart represent the affection of different β on the DMR of Baseline, PEND and SPEND. The DMR of PEND and SPEND is not influenced by the dynamical arrival of BLE-As since they activate the BLE interface and the scanning process if and only if there is a discoverable BLE-A. Thus there is just one trend attributed to PEND/SPEND.

strongly dependent by the BLE-A's arrival. From Fig. 3.18 is easy to see how the DMR of the Baseline increases with the enlarging of the frame length. On the BLE-A arrival rate side, the DMR decreases as the inter-arrival(β) time grows. From the Fig. 3.18 emerges that in case of advertisers' arrival rate reasonable high, i.e. β = 30 sec., and a scanner's frame length large enough, i.e. 60 sec., the DMR performance of the Baseline scheme might reach a performance close to the PEND/SPEND one; As a side effect, it can drive to massive battery consumption. On the other hand, when the advertisers' inter-arrival time grows, the DMR performance of Baseline scheme significantly decrease down to less then 20%, in case of λ = 120 sec, and BLE-S' scanning frame length 10 seconds.

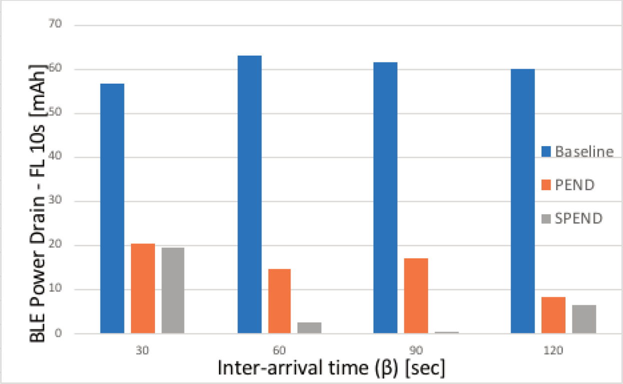
In Fig. 3.19 are depicted the broken down charts of BLE interface power consumption on varying of BLE-S scanning frame length and the BLE-A inter-arrival time. It is studied the power consumption of the BLE under a scanning frame length of 10, 30 and 60 seconds and by setting β, the advertiser inter-arrival time, to 30, 60, 90, and 120 seconds. From Fig 3.19 emerges how, under baseline scheme, the power consumption decreases as the scanning frame length gets longer. While the trend of the power consumption grows with the enlarging of the scanning windows under PEND. On the other hand, PEND scheme takes advantage form fog node awareness and the BLE-S's scanning process is activated only when a device is in discovery range, hence it also guarantees the discovery of new device. Under this assumption, it is reasonable to state that PEND does not need to have a long frame length. With an high BLE-A arrival rate (β = 30 s), the power drained under PEND increases accordingly with the increasing of frame length. It becomes greater than the energy drained under Baseline. This because, under PEND scheme, with this experiments settings, an high arrival rate combined with a large frame length leads the BLE-S to be always ON and the scanning process always active. On the other hand, the Baseline scheme is not advertiser's arrival dependent, hence it is not affected by the arrival rate.



(a) BLE power drain with a scanning frame length of 10 seconds, under different inter-arrival time



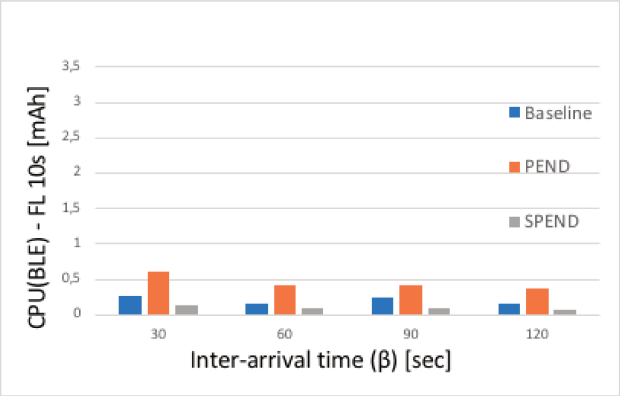
(b) BLE power drain with a scanning frame length of 30 seconds, under different inter-arrival time



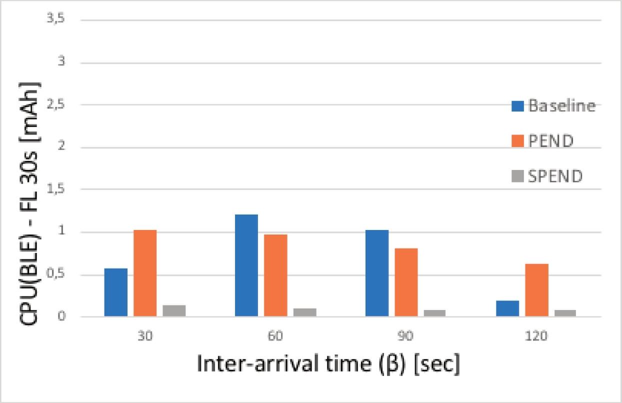
(c) BLE power drain with a scanning frame length of 60 seconds, under different inter-arrival time

Figure 3.19: BLE power consumption under different scanning frame length and advertiser's inter-arrival time

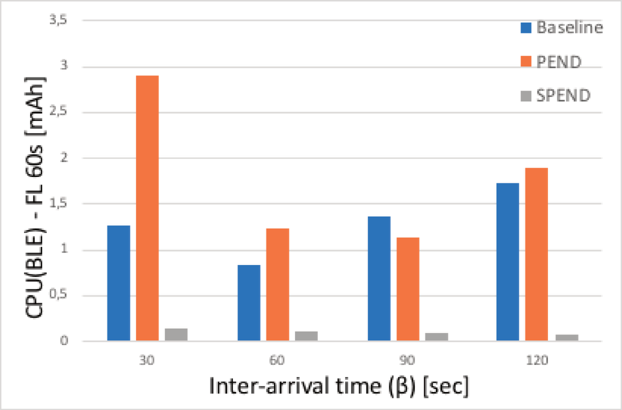
This means that, if the advertisers' arrival rate is very high, it might be better to adopt a Baseline-like strategy instead of PEND-like one in terms of power efficiency. The dependency of PEND/SPEND from the advertisers' arrival leads the proposed schemes to have better performance in term of power consumption as the time between two consecutive arrivals (β) increases. On the other hand, the Baseline scheme is arrival independent, hence the power consumption is about constant under different β. SPEND shuts immediately down the scanning process and the BLE interface after the discovery of the new device, it does not wait for the entire frame length. This behavior makes the scheme independent by the different scanning frame lengths. SPEND results to have the best performance in terms of power consumption under every condition, it also guarantees the device discoverability. In the next figure (Fig. 3.20) is depicted the power drained by the CPU due to the BLE usage.



(a) CPU power drain due to Bluetooth activity with a scanning frame length of 10 seconds, under different inter-arrival time



(b) CPU power drain due to Bluetooth activity with a scanning frame length of 30 seconds, under different inter-arrival time

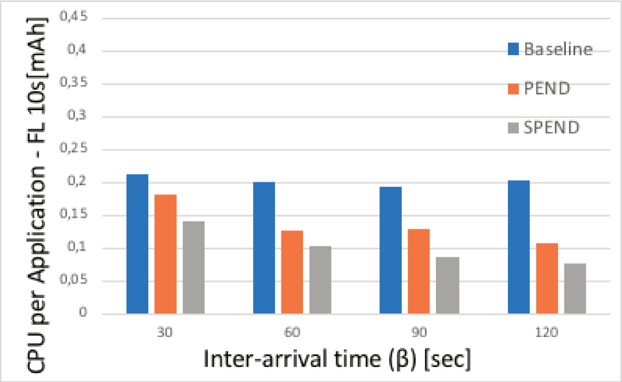


(c) CPU power drain due to Bluetooth activity with a scanning frame length of 60 seconds, under different inter-arrival time

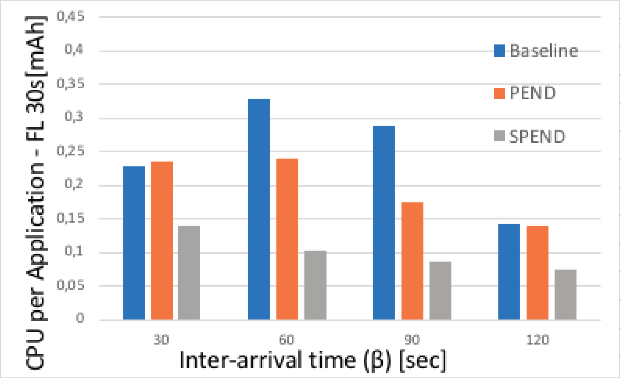
Figure 3.20: CPU power drain due to Bluetooth activity under different scanning frame length and advertiser's inter-arrival time

Fig. 3.20 shows the broken down charts of the power drained by the CPU due to the BLE interface usage. The charts represents the power consumed by varying the arrival time between two consecutive BLE-As (β) from 30 up to 120 seconds in average. The three charts relates the power consumption just described with different values of β on changing of the BLE-S's scanning frame length, 10, 30, 60 seconds in, 3.20a, 3.20b, and 3.20c respectively. Fig. 3.20 highlights how the power consumption generally grows as the BLE-S's scanning window enlarges. One of the impact factor of this consumption is the number of scanning cycles on the three different frequency channels during the scanning process. In fact, when the scanning process is active, the BLE interface constantly scans three channels on three different frequencies, namely 37. 38. 39 (2402, 2426, and 2480 MHz). Hence, it is understandable that larger is the scanning frame length, higher is the number of cycles per window. The worst case occurs under PEND scheme with frame length 60 seconds and inter-arrival time 30 seconds. It is worthy to remind that the scanning process under PEND is BLE-A's arrival dependent and MQTT-triggered, with a large scanning window and an arrival rate high enough, the BLE-S's scanning process would be always active. On the other hand, Baseline scenario is not arrival dependent and the consumption under this scheme is driven by the number of times BLE-S discovers a new device. Finally, SPEND is proved to be the best scheme even under this aspect. As the other experiments, it is not affected by the window's length, and thanks to its policy of shutting down the scanning process as soon as a new device is discovered, SPEND has very low power consumption. The energy drained is low also because SPEND discovers the new device very probably at the first scanning cycle, then it stops the scanning process, hence, the consumption of energy is kept as low as possible. In the next figure, the energy consumption due to CPU usage by the application is depicted.

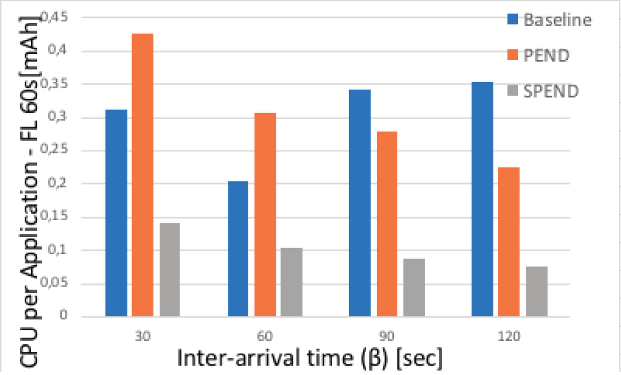
The charts in Fig. 3.21 depicts the power consumed by the CPU of whole application under different settings of BLE-S's scanning frame length and BLE-A's inter-arrival time. It is worthy to be underlined that the CPU power consumption showed in this figure, represents the CPU power consumption of the whole application, and it differs from that one in fig.3.20 because it refers to the CPU power drained due to BLE utilization, and it is accounted into BLE power consumption. Generally, the trend of CPU power consumption of the whole application follows the BLE's one. It increases as the scanning frame gets larger under each scheme. From Fig. 3.21a emerges that the Baseline scenario has an higher power consumption than PEND and SPEND, and its level of CPU power drain is more or less constant on varying of BLE-A's arrival rate. This chart clearly shows how the Baseline scenario is device arrival independent. On the other hand, under PEND and SPEND, the scanning process is device's arrivals driven hence, lower is the arrival frequency, lower is the power consumption. SPEND consumes less energy than PEND thanks to its immediately stopping scan process policy upon the device is discovered. The CPU power consumption of the application under PEND and SPEND scheme reasonably increases as the arrival frequency grows, in other words, as the β increases, the frequency accordingly decreases and the power consumed lowers. The worst scenario for PEND scheme, as in Fig. 3.20, occurs with a scanning window of 60 seconds and an advertiser's inter-arrival time of 30 seconds. As explained above, with an high arrival rate and a long scanning frame length, the scanner would be always active and it would scan continuously. The consumes under SPEND are the same on all three charts, for all



(a) CPU power consumption of the whole application. It run with a scanning frame length of 10 seconds, under different inter-arrival time



(b) CPU power consumption of the whole application. It run with a scanning frame length of 30 seconds, under different inter-arrival time



(c) CPU power consumption of the whole application. It run with a scanning frame length of 60 seconds, under different inter-arrival time

Figure 3.21: CPU power consumption of the whole application. with different BLE-S's scanning frame length and varying the BLE-A's inter-arrival time

values of frame length. SPEND is not affected by the scanning frame length. Also in this case, SPEND is proved to be the best scheme in terms of power consumption.

3.4.4 Energy saving strategy implementation

In this subsection, it is presented the study of the proposed energy saving strategy due to the implementation of the formula 3.3. All the parameters has been calculated and estimated from the experiment run and presented in the above sections. Traditional conventional scenario or Baseline, PEND, and SPEND, have been considered for the application of formulas. Those data has been used to calculate the theoretical maximum number of BLE interface ON/OFF switching before a smart interface switching approach turns to be more power inefficient than traditional scenario. Then, it is also analyzed and plot how this number change by varying the frequency of BLE-A's arrival. As it is already mentioned, the data used for implementing the formula3.3b has been calculated by the experiments run on PEND, and conventional scenario, and the data are summarized in table3.4.

Table 3.4: Parameters values of the optimal number of switching equation.

Parameter	**Value**
EUP	43,5 [mAh]
e_{wifi}	3,25 [mAh]
e_{sw} + e_{SCAN}	0,48 [mAh]
N	$\{20, 30, 40, 50\}$

EUP is the energy of keeping the BLE interface constantly ON and in periodic scanning. It is the energy consumption that reflects the scanner behaviour under the conventional scenario. Briefly, it is statically tuned the BLE-S's scanning frequency and scanning frame length by setting the inter-scanning period to 60 seconds, and the frame length to 30 seconds. By doing so, I focus the attention on BLE-A's arrival, and I make the equation strictly dependent by advertiser's arrival frequency. Once the scanning frequency is defined as static, EUP becomes a constant and its value can be easily fetched by conventional scenario experiment. The other values of the parameters has been fetched by the energy consumption report made by the Android Device Bridge during the experiments [138]. The values are calibrated on experiment with a finite time duration of 60 minutes. The energy consumption due to wifi interface utilization is expressed with e_{wifi}, while e_{sw} + e_{SCAN} is the energy consumed by a single switching ON/OFF of the BLE interface plus the energy due to the scanning process. N is the number of the total advertisers in the system. Fig. 3.22 depicts the ration between EUP and ESW on varying of the BLE-A's arrival period. Four trends are plotted, each of them represents a different total number of BLE-A in the system, N.

The chart in Fig. 3.22 shows how the number of optimal switching varies according with the arrival of the BLE-As. The proposed strategy is implemented with four different number of total advertiser devices within the whole system. The values of N used are, 20, 30, 40, 50 devices. On the X axis there are the values of the advertiser arrival

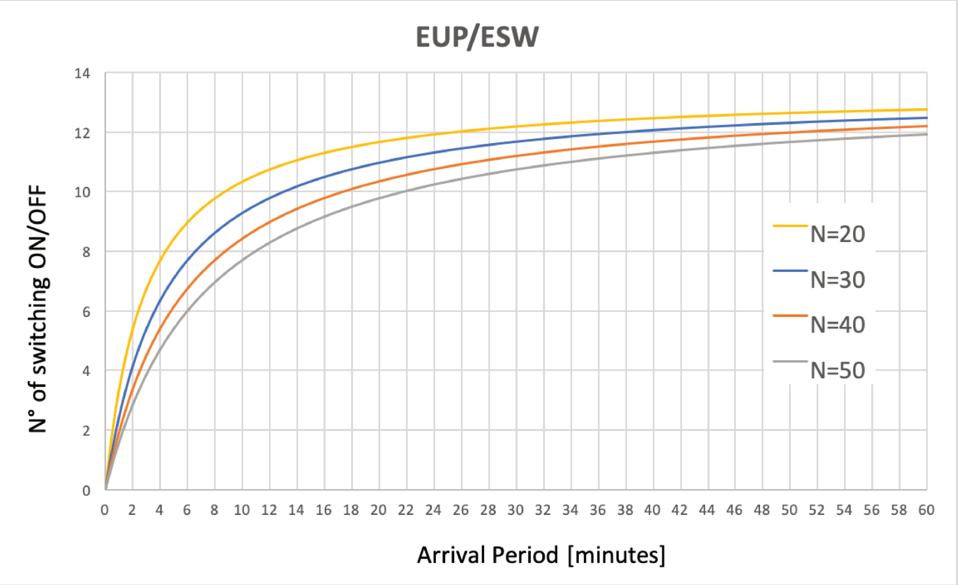


Figure 3.22: Ratio between EUP and ESW. It results in the optimal number of ON/OFF switching of the scanner's BLE interface according with 2-competitive strategy applied to the model

period, expressed in minutes, while the Y axis represent the optimal number of ON/OFF switching. From the chart in Fig. 3.22 emerges the ratio between EUP and ESW has an asymptotic trend. This behaviour is reasonable because with the enlarging of the arrival period, the BLE-A arrives ever more sporadically. With a that very occasional arrival, the BLE-S basically, does not ever switch ON its BLE interface, and the e_{wifi} becomes the predominant component of the ESW. If the arrival period is pushed to the infinite, the energy consumption of the BLE interface becomes 0, because the BLE-S never switches the BLE interface ON, and the ratio between EUP and ESW turns to be a fraction of two constant value. The value of the asymptote is the result of outcoming division. On the other hand, with a BLE-A's arrival period close to zero, hence with an arrival frequency very high, the number of time of BLE interface ON/OFF switching drastically drops to zero. This because, under an high BLE-A arrival rate, it would be much more power efficient to keep the BLE interface always ON and scanning. In the chart depicted in Fig. 3.23, it is put in relation the energy values of the two component, EUP, and ESW, with the BLE-As' arrival period.

Also in Fig. 3.23, the trends of ESW are plotted under the same four different number of total devices as before, 20, 30, 40, 50. In addition, in this chart, it is also ploted the behaviour of EUP. This trend has a constant value because, by definition, it is not affected by BLE-A's arrival. On the other hand, the trends referring to ESW are strongly dependent by advertisers' dinamicity. Indeed, from the Fig. 3.23, clearly emerges how the energy consumption varies according with the arrival period. The ESW component has a asymptotic behavior. With the enlarging of arrival period, the BLE-S using the BLE interface switching strategy consumes ever less. If the BLE-As' arrival period is pushed to infinite the trends of ESW result to be 3,25, the value of energy consumed by the wifi usage. This is reasonable because with an infinite arrival period, hence an arrival frequency equal to zero, the BLE-As result to be never approaching, hence, the component of energy consumption due to the BLE interface usage is 0, and the ESW

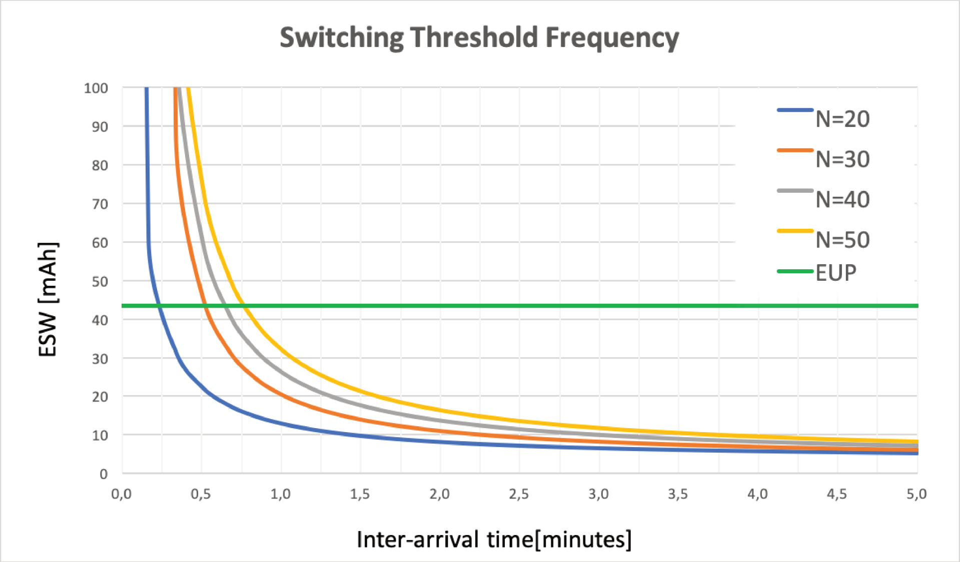


Figure 3.23: Energy values of EUP and ESW in relation with the BLE-A arrival period

results to be e_{wifi}. In the opposite case, with a very little arrival period, hence with an arrival frequency high, the energy consumption due to the switching strategy results to be extremely high. Under this condition applying the strategy of keeping the BLE interface always ON and in periodic scanning would result much more power efficient. According with what it has already stated so far, a question might be risen, what is the advertiser's threshold arrival frequency after which is better to switch from a BLE interface switching strategy to a BLE interface static one? It is worthy to be highlighted that the EUP straight line intersects the curves of the ESW trends in a point. The value of projection of that point on the X axis is the threshold arrival period, hence the threshold arrival frequency can be easily obtained.

Chapter 4

Cloud Infrastructure for IoT

In the previous chapter it has been addressed the interaction Fog-IoT aimed to resources provisioning. The smart fog-driven device discovery protocol and fog middleware architecture have been introduced, presented and properly addressed. The IoT-Fog interaction accounts the resource provisioning, in other terms, the physical real device provisioning in order to fetch data and environment information to expose at higher layer. This chapter addresses the upper layer of the architecture depicted in fig 3.1. It presents the Fog-Cloud interaction, the agents/actors working at high level, and the applications of the entire architectural stack from cloud top to the bottom IoT tier. The interaction fog middleware-cloud has the crucial role of making the data and the environment information, gathered from IoT layer, available at higher layer. In this way the data sensed, acted, and other information, can be used by cloud services and application. With this level of abstraction the virtual sensor, more generally, the data provided by the middleware, can be directly used by end user application or can be combined with other services or application in order to create a complex system. This chapter, will address a methodology for resource provisioning, a common way to access, query, and perform actions on the data provided by the fog middleware. The chapter will also present an fog middleware application in a real and mature telecommunication scenario. This application highlights the capacity of fog middleware to provide elasticity in a complex multi services system. Finally, a complete use case of the entire vertical architecture implemented in the European Arrowhead project. With an higher level of detail, the main content this chapter is going to address are:

Resource Provisioning and Data Availability The first point addressed by this chapter is how a third party entity, that might be an end user application or another service, can access the data, and can use them for its purpose. For achieving data accessibility and availability it has been chosen Semantic Web, and its standard de-facto query language, SPARQL. This choice has been taken because Semantic Web and SPARQL have proven to be crucial in modeling e managing heterogeneous type of data/entities and supporting interoperability.

Virtual Sensor Orchestration At cloud level, once the data and information are made available and accessible, the providing of those resources has to be managed and orchestrated among the several requests. As it is already stated in section 1.2.3, NFV in fog computing is paving the way to new solutions in many fields. The field that benefits the most is telecommunication. This chapter presents an application of NFV on fog nodes and how this function is orchestrated cloud-side to respond to elasticity needs.

Real Use Case Finally, this chapter presents a real use case based on a real and concrete application of the vertical cloud-fog-IoT architecture applied to an electromobility (EM) scenario. This scenario is part of an European project named Arrowhead. In this EM scenario the presented architecture has been used to facilitate the usage of electric cars by providing an entire framework aimed to offer to the user many services.

4.1 Virtual Sensor Data Accessibility and Availability

Upon data, and environmental information have been collected from IoT physical nodes, the next step the fog middleware is to expose such data to cloud side. At cloud layer such data can be accessed, managed, manipulated, and used by end user application or other services. These data are heterogeneous and have heterogeneous sources, but it is needed to provide a common method to let the other entities in the cloud access such information. Among the several methodologies to access data in the web, in this research work, it has been chosen Semantic Web.

Semantic web principles, methodologies, and techniques have long ago proven to be crucial in modeling and managing complex relationships between entities/data and to support interoperability, reasoning and inference/automation in coarse-grained, semistructured, and heterogeneous data environments: in this field, the SPARQL protocol (based on the Resource Description Framework – RDF – specification [140]) has become the de-facto standard to extract and manipulate information from distributed data sources on the web [141, 142]. However, traditional Semantic Web methodologies and techniques typically exhibit severe limitations in large heterogeneous scenarios where multiple distributed data sources have to be integrated. In those cases, Semantic approaches would require complete a priori knowledge of all data sources and their network distribution and topology. That limitation becomes important if applied to IoT-Fog distributed and heterogeneous scenarios where a complete, durable, and a priori data source knowledge is often unrealistic and unfeasible. Another typical example in which the above limitation are even more particularly crucial in Enterprise IT/Data Governance scenarios. In this case, a possible example would be one where multiple public and private academic institutions are willing to integrate to optimize their geographic and demographic coverage (e.g., to avoid geographic course overlap from different academias). Such synergic education offering would require a continuous data integration flow from both education (e.g., currently available courses and teaching areas) and administrative departments (e.g., students distribution, cost and revenue distribution per area and department), with demographic data from municipalities (e.g. Open Data about citizen distribution per geographic area and age). Both municipalities and academic institutions will likely feature their own, very different and constrained data sources, information systems and data gathering and management processes, so making them converge to single, uniform and standardized data sources, systems, and processes would be simply unrealistic. Similarly, both academic institutions and municipalities should be left free to change/evolve their own IT infrastructures/systems and processes, and still guarantee data interoperability and integration. Despite the aforementioned limitations in terms of distributed data source knowledge, this research project proves that flexibility and extensibility of Semantic approaches are particularly suitable for IoT-Fog and IT/Data Governance scenarios. This research project proposes a novel, semantic-based approach to overcome data provisioning complexity and to mitigate/hide heterogeneity and distribution of data sources. The next sub sections will present and describe the reference architecture model, adopted in this research, that relies on a federation of SPARQL endpoints, as well as a full implementation on top of an existing, largely adopted Semantic framework, and a real use-case scenarios as example of the viability of the used approach. This approach goes further than traditional IoT nodes managing models (where data harvesting and integration are expensive, difficult, and often limited to specific domains/areas), and fosters a much broader-scoped, horizontal, and continuous integration of data: the adopted federated approach promotes a higher level of data normalization and integration which proposes a more proactive managing/provisioning model where data analysis may be used not only to lead retrospective analysis, but also to proactively support upcoming strategic decisions. The above integration example will be used as a fil rouge throughout the next sections.

4.1.1 The SPARQL Federation Model

Data Federation is crucial in letting the fog middleware easily and effectively shaping and supporting information flow across the various services and application branches. The research done aimed at defining a viable and effective model and implementation to adopt Semantic Web methodologies and the SPARQL implementation to overcome typical issues such as heterogeneity and distribution of data sources. Key principles in defining the used approach are:

Opennes: data federation and integration in large IoT scenarios typically means integrating data sources from heterogeneous (both custom and third party) nodes/devices; avoiding vendor lock-in and preserving openness and portability is key in defining a sustainable, long-term data federation strategy for any IoT-Fog scenario;

Lightweightness: the adopted solution poses from a limited up to no overhead on running systems, so as to minimize the impact of Data Federation on large IoT scenarios with complex, highly distributed data sources;

Autonomy and ease of use: the proposed solution should be able to cope with uncertainty, and to autonomously discover relationships among federated data even in case of partial a priori data model and network knowledge.

Semantic Web standards and the SPARQL query language have long proven to be the key in enabling open, autonomous, machine-driven content matching and reasoning knowledge management infrastructures. However, SPARQL support for data federation – via the SERVICE construct (e.g. distributed data source integration and query) – is still at an early stage and it is still poorly suitable for large, real-world scenarios. Current SPARQL data federation limitations mainly relate to the fact that designing federated queries requires complete, a priori knowledge of data models and actual data across all data sources involved; this clearly becomes a relevant constraint in large complex scenarios where data sources are heterogeneous and can change frequently and at different paces from each other, likes IoT. The only way to overcome that limit is to find new mechanisms capable of dynamic behavior for a suitable adaptation to any scenario change, very likely to occur in large environments with many actors.

In the appendix A, the main architectural approaches to realize a scalable and extensible SPARQL endpoint federation has been described. In addition, that appendix introduces the KPIs used to compare the different approaches and highlighting the motivation of choosing the Federation Web Services solution. Then, the appendix A, addresses the study several evaluated middleware platform. This study compares all the evaluated middleware platforms according the listed KPIs and motivates the choice of OpenLink Virtuoso [143].

4.1.2 Adopted Architecture and Platform

This sub section highlights some relevant implementation details related to the selection of the Semantic middleware adopted (and related evaluation KPIs) and Federation Web Service implementation.

4.1.3 Federation Web Service implementation

The implementation of federation consisted of two main aspects: the Federation Ontology, and the Federation Web Service that together have granted the full support needed in the project.

Federation Ontology

IoT-Fog contexts stress the need to manage large amounts of data that show properties such as heterogeneity and distribution of the data model. However, aggregation of heterogeneous data is crucial for business processes such as Decision Making, Device Quality, and Data Management. This aggregation supports the concept of federation, which provides a federative pact on which to build the federation itself. The implementation follows the example introduced above (section 4.1). The implementation has gone in this direction, defining a uniform data model and to be shared between all members of the federation, in order to eliminate the problems described above. The ontology is composed by three main elements:

- The concept of federation;
- The concept of the federation member
- The member_of relationship that represents the relationship between the member of the federation and the federation itself.

The ontology described above enables the management of the federation independently of semantic platform and can be deployed and queried to any SPARQL endpoint. From an implementation standpoint, we realized our ontology according to the RDF schema modeling.

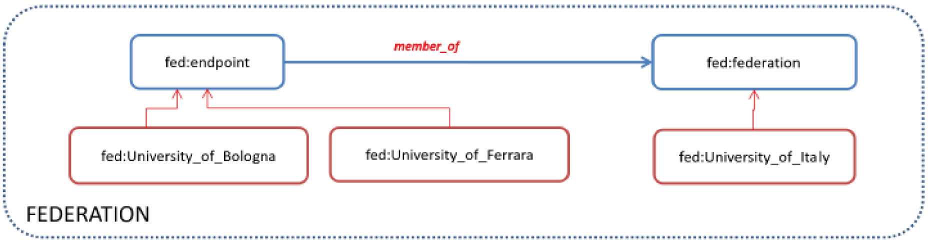


Figure 4.1: Federation Ontology

Figure 4 describes an implementation of our ontology that maps Italian academic institutions: *fed:University_of_Italy* represents a federation of universities, and *fed:University_of_Bologna* and *fed:University_of_Ferrara* are endpoint members of the *fed:University_of_Italy federation*.

Federation Web Service

The Federation Web Service manages semantic data query and aggregation via the steps described in the following.

1. Clients can query the Web Service with a standard SPARQL query, and express a specific federation (in our examples, we modeled disparate data sets from Italian universities)
2. The Web Service inspects the federation to determine which endpoints should be queried
3. The Web Service performs the semantic query on each such node
4. The Web Service aggregates results from the above queries and returns them back to the caller

Result aggregation is a particularly crucial task, the adopted Federated Web Services support three main strategies:

APPEND Strategy: results coming from different endpoints are simply stacked on top of each other, and sent back to the client once all replies are received; this strategy is implemented via the APPEND operator and does not handle result triple duplication;

INTERSECTION Strategy: this approach returns all the triples that are common to each involved endpoint;

UNION Strategy: results from endpoints are combined into a set of unique triples, hence performing duplicate detection and removal.

The Federation Web Service clearly decouples the query definition, execution, and aggregation from the definition and management of the data set distribution. Even though this solution may conceptually handle highly distributed data sets, some management concerns may arise:

Endpoint Timeout: how to handle alive nodes that fail to reply to queries in a given amount of time (e.g., due to temporary network issues); ignore/retry strategies should be carefully planned to reach a viable balance between result completeness (e.g. retrying queries in order to overcome the temporary network outage), and total response time;

endpoint unavailability: endpoints may become unavailable for larger amounts of time, e.g. due to severe server/system failures.

4.1.4 Evaluation

In this sub section we describe some Semantic platform benchmarks, as well as a specific use case of our SPARQL federation model

Semantic platforms benchmarking

In order to prove the feasibility of our approach, we provide some useful insights about semantic platform performance under heavy load. These quantitative analyses justify both Virtuoso adoption as the Semantic middleware/platform of our choice with respect to other contenders, and shed some light on absolute performance of Virtuoso under load.

Dataset Load Time Loading large amounts of data becomes crucial in complex distributed scenarios where each node may contribute with a set of millions of records, thus limiting the overall performance of the whole system. The table below shows a load time comparison for datasets ranging from 1M to 100M entries, the times in the table are expressed according the dd-hh-mm-ss format.

Table 4.1: Semantic middleware performance – dataset load time

	1M	25M	100M
Sesame	00:02:59	12:17:05	03:06:27:25
Virtuoso	**00:00:34**	**00:17:15**	**1:03:53**
Allegrograph	00:00:52	00:36:49	00:48:30
Jena TDB	00:49	00:16:53	01:34:14

Query execution Querying large amounts of data becomes crucial in complex distributed scenarios where each node may contribute with a set of millions of records, thus limiting the overall performance of the whole system. The table below shows a load time comparison for datasets ranging from 1M to 100M entries (Query per hour).

Table 4.2: Semantic middleware performance – query execution/single client

Single Client	1M	25M	100M
Sesame	18094	1343	254
Virtuoso	**17424**	**12972**	**4407**
Allegrograph	4075	493	656
Jena TDB	4450	353	81

Table 4.3: Semantic middleware performance – query execution/multiple clients

Multi Client	1M	25M	100M
Sesame	19057	18295	16517
Virtuoso	**28985**	**32668**	**33339**
Allegrograph	5861	7453	7888
Jena TDB	6752	8453	8664

The Semantic middleware taken as reference implementation – Virtuoso – clearly outperforms its contenders both in terms of initial dataset load time, and in terms of query efficiency. These numbers also clearly evidence how Virtuoso may scale linearly and handle extremely high workloads both in terms of initial dataset load, and in terms of single/parallel query executions, thus proving itself as a viable, production-grade option for large real data federation scenarios.
The following sub section presents a realization of a complex integration scenario adopting the federated web services. The scanario addressed is that one described at the beginning of section 4.1.

Federated Web Service: a Use Case

The SPARQL Federation model has realized the complex integration scenario described in section 4.1. It has been realized a proof of concept Federated Education Portal that provides a synergic academic offering that federates education and administrative data sources from Italian academic institutions together with citizenship distribution data sources from Italian municipalities in order to realize a more integrated education offering. A traditional Semantic approach allows to decouple actual data sources and their implementations, hence allowing to reuse the same SPARQL query over any academic and municipality data source, no matter the real data source implementation (would it be an ERP system backed by a traditional RDBMS, or a legacy mainframe system). However, a SPARQL-only approach for the Federated Education Portal would require to:

- have a priori knowledge of all municipalities and universities involved in the overall integrated offering;
- explicitly/manually perform the same SPARQL query on academic data sources to retrieve students distribution;

- explicitly/manually perform the same SPARQL query on municipality data sources to retrieve citizen distribution;
- explicitly map and combine both semantic result sets into a cohesive data set that highlights gaps in actual course offering with respect to real citizen distribution.

This process is obviously largely inefficient and poorly extensible: our Federated Education Portal should be extended any time new data sources get added, in order to query new endpoints and combine results with old ones. The approach adopted leverages the following elements:

- A Federation Ontology implementation that maps Italian municipalities and relevant information about citizen geographic distribution;
- A Federation Ontology implementation that maps Italian academic institutions and relevant information about courses and student distribution;
- Students and Citizens are linked via the semantic notion of person (via the *foaf:Person ontology*);
- a set of Virtuoso instances as the default Semantic middleware;
- a set of SPARQL endpoints, for both municipalities and universities, on top of Virtuoso Semantic middleware;
- an instance of our Federation Web Service.

In this scenario

- Federated Education Portal performs a single query to retrieve geographic distribution of both persons involved in academic courses (students), and persons (citizens) from municipalities;
- the Federation Web Service identifies all involved academic and municipality data sources and transparently query each one of them;
- the Federation Web Service takes care of combining results (e.g., via an INTERSECTION strategy).

The adopted approach dramatically facilitated the realization of the Federated Education Portal:

- no a priori data source knowledge should be hard-coded into the Federated Education Portal, hence resulting in a more open and flexible solution;
- a single query can be executed both on academia and on municipality endpoints, hence facilitating the development efforts of the overall solution.

4.2 Fog Nodes for NFV, a Service Composition Application

The vastly spread of IoT application usage and development led to consider the networking-relative issues as a primary concern. With the huge number of devices and such massive IoT application all the network-oriented aspects become crucial for the IoT Quality of Service, hence for satisfying the end user. Among the other, to keep latency low, has a main role in achieving a very positive user experience. Under these presuppositions, NFV is even more an emerging solution for fog environments. As it is already explained, NFV replaces the network functions with VM instances. Since the key enabler of fog computing is virtualization and those VMs can be dynamically created, destroyed and offloaded, NFV will benefit fog computing in many networking aspects by virtualizing gateways, switches, load balancers, and placing those instances on fog nodes. This section presents an application that exploits NFV techniques in telecommunication field. This application is cloud-based but through the adoption of NFV, it proves that NFV applied to fog and fog nodes, not only it is feasible, but it will play a very important role in supporting the future telecommunication's infrastructures starting from 5G network. The following example is part of vast and very important European project, named EU FP7 Mobile Cloud Networking, I have joined [144].

MCN Brief Introduction

The Mobile Cloud Networking (MCN) is a EU FP7 Large-scale Integrating Project (IP) funded by the European Commission. MCN project was launched in November 2012 for the period of 36 month. The project is coordinated by SAP, and ZHAW is the technical leader. In total top-tier 19 partners from industry and academia commit to jointly establish the vision of Mobile Cloud Networking. The project is primarily motivated by an ongoing transformation that drives the convergence between the Mobile Communications and Cloud Computing industry enabled by the Internet. These observations led to a number of objectives to be investigated, implemented, and evaluated over the course of the project [144]. In this project, I joined for the six months extension inside a project that touches involved in multiple Work Packages (WP). More precisely, my work has primary fallen in WP3.

This WP addressed the providing of the necessary foundational infrastructural resources and services required to create, enable and deliver fully virtualised end-to-end Mobile-Cloud services. They will be offered in an on-demand, self-service manner, charged on a per usage/time basis. They will be implemented with resource pooling and multi-tenancy capabilities. Those underlying resources will be easily scaled up and down, scaled out and in, both through self-service facilities and also internal QoE/QoS adjustment mechanisms. The main objectives targeted by WP3 are:

- To design and implement a system enabling and providing virtualised foundational infrastructural resources from the various service domains within MobileCloud, namely Radio Access Network, Mobile Core Network and Data Centre. Here providing virtualisation technologies upon commodity hardware will be a focus;

- To design a coordination and orchestration function from the infrastructure perspective for the composition, provisioning, life cycle management of services and maintenance of agreed Service Level Agreement (SLA) guarantees spread across MobileCloud's service domains;
- To provide monitoring facilities for extraction, pre-processing, distribution, storage, analysis and notification of metrics such that higher-level services can utilise these facilities to enable adaptive platform services (such as SLAs; the Follow-Me concept), where certain guarantees and service quality (QoS) and/or experience (QoE) must be maintained;
- To design the mechanisms to deliver those foundational virtual resources as service for the execution of non-traditional virtualised instances such as baseband unit, IMS and EPC functionality (WP4, WP5);
- To define unified and consistent interfaces on top of and between disparate infrastructure management frameworks. Through the definition of these interfaces this work package will receive and supply architectural input from and to WP2.

Following a brief context overview about where the application is placed, and then, how the NFV techniques have been exploited and adopted in order to provide infrastructure elasticity.

4.2.1 Scenario Overview

Elasticity represents one of the most relevant properties in cloud computing, providing capabilities for increasing or decreasing on demand the infrastructure resources utilized for building up web services. More recently, also telco service providers are looking for ways to flexibly and cost-effectively deploy their services on top of private cloud infrastructures, as well as to offer cloud service enablement of virtualized network infrastructures on top of their next generation networking infrastructures, all bundled under the Network Functions Virtualization (NFV) initiative within ETSI [145]. ETSI NFV provides a set of guidelines for defining a framework architecture aimed to offer Network Functions as a Service (NFaaS). There is the need for relevant integration, deployment, and experimentation work in order to provide mature framework prototypes where telco service providers have the possibility to monitor, control, audit and accordingly perform dynamic management operations on cloud-provided resources. Of course, one primary technical challenge is the ability to offer guaranteed quality for service provisioning in virtualized environments, with only partially predictable traffic trends, that is comparable to the quality traditionally offered by telco providers via dedicated hardware resources [146]. Considering the 5G evolution, applications will play a central role in defining the new set of requirements of the new core network architecture. Among these, the IP Multimedia Subsystem (IMS) has played and is still playing a relevant role in 3G/4G telco support infrastructures, with many general concepts that are going to persist in 5G networking [147]. The IMS defines an overlay architecture for session control in all-IP next generation networking to obtain openness and interoperability by using an application-layer approach, mainly by using the Session Initiation Protocol (SIP) as the central core mechanism. Although IMS and

its Cloud integration are no longer an innovative novelty, there are some still open well-recognized challenges, for example the resource provisioning for multimedia-oriented applications or scalability of IMS components. This work overcomes the scalability limitation of IMS-based cloud scenarios. In literature some solutions to overcome this issues are already present, the majority of them being focused on scalability and/or reliability of the signaling plane. Our work, instead, is fully integrated with the signaling plane, but its main focus is on scalability of data plane so to provide horizontal elasticity for increasing user demands.
Within this general context, the following research addresses the work of development aimed to achieve cost-effective and industry-mature elasticity of IMS-based multimedia applications. This work proposes a novel solution to overcome above mentioned limitations that presents the following main features. First, it has been proposed and realized a novel MultiMedia Application Server (MMAS), which plays the role of an Application Server in the IMS infrastructure (at the signaling-plane) handling the scale in/out operations of a media gateway (at the data-plane). Second, the integration of MMAS with the novel, open-source, and fully NFV-compliant Open Baton NFV framework has been leveraged, and, it has been recently integrated in the MCN architecture and exploited by providing elasticity mechanisms that can be automatically triggered and invoked at run-time for optimizing resource usage. Third, a practical example of implementation work in full compliance with industrial standards and fully based on open-source reference implementations has been done. This practical example is widely recognized as crucial in this field to enable rapid adoption in real industrial environments. The work is based also on standard and industry-mature SIP User Agent Clients (UACs) and User Agent Servers (UASs), as well as with the widespread Asterisk Media Gateway (AMGW) for multimedia dataflow management. Fourth, it has provided to the IMS community the implemented MMAS that interworks with the MCN Monitoring as a Service (MaaS) in order to trigger proper scale in/out management operations and to deploy/manage the resource instances via the intermediation of the Open Baton orchestration1. The implementation and deployment experience proved the feasibility of the adopted approach, even under the challenging technical constraints due to multimedia provisioning and the associated quality requirements, and it paves the way to the massive adoption of NFV techniques on fog nodes for the next generation network.

In the appendix B, it is provided a detailed and deepened overview of the orchestrator used in this project, namely Open Baton.

4.2.2 Elasticity in IMSaaS: Concept and Architecure

Our preliminary work in the field has shown that the most critical element of the IMS infrastructure is the Media Gateway (MGW) component [148], especially when offering multimedia provisioning services with quality requirements and for high numbers of concurrent multimedia data flows. For this reason, our proposal is concentrated on simple extensible mechanisms and integration work for supporting scaling-out/-in operations of the MGW function over the multimedia data plane. Our scale-out/-in solution is based on three main components, namely, the MCN MaaS, the OpenBaton Orchestrator (indicated as NFVO in the following), and our original MMAS, working in a pipeline to operate the needed scalability management actions. First of all, we exploit the MaaS service for triggering scaling alarms as needed. In particular, MaaS monitors the execution of the most critical MGW function and can raise monitoring alarms to the NFVO that, for the sake of simplicity, here we consider both as the scale-out/-in decision entity and orchestrator of the whole scale process. Hence, once alerted, and in case of overloading of a MGW instance (for the sake of simplicity, let us assume to have initially one only overloaded MGW instance, namely, MGW1), NFVO can horizontally scale-out the MGW, by adding another instance. Finally, the MMAS behaves like a dynamic load-sharing for the incoming session establishment requests at runtime, able to exploit all available MGW instances. After introducing the general concept, we fully and comprehensively describe the architecture of our proposed solution for Elastic IMSaaS (EIMSaaS), also for the full understanding of the proposal and of its implementation insights, as well as the primary motivations behind our design choices. The EIMSaaS includes a fully-compliant IMS Client, which controls session setup and media transport by implementing all SIP extensions specified by IETF and 3GPP IMS-related standards. Any session is setup between two IMS clients playing the roles of User Agent Client (UAC) and User Agent Server (UAS). Then, as already anticipated, the second crucial component of our architecture is MMAS, which allows the introduction of new IMS services and extensions, including the MGW. MMAS has full control over traversing SIP dialogs and session descriptions. The third EIMSaaS architecture component is the standard IMS Serving-Call Session Control Function (S-CSCF), which receives register requests from IMS clients and authenticates them; then, depending on filters/triggers specified by client profiles and dynamically downloaded from the Home Subscriber Server [147], S-CSCF may either route incoming SIP messages from UAC directly to UAS or forward them to our MMAS. The last component of central interest here is the MGW, an external component controlled by IMS that takes part to the media delivery plane between UAC and UAS, and that we have integrated in our solution by employing the widespread Asterisk media server, as detailed in the following. As well known, given its ability to re-route traffic in the IMS infrastructure, the S-CSCF can extend UAC-to-UAS session signaling paths through the interposition of convenient application servers. We exploit exactly this characteristic in our solution where our MMAS can participate to multimedia content transport/buffering/adaptation as well as enabling elastic scalability by interacting with media gateways suitably deployed at the IMS media plane. By delving into finer details fully complying with the IMS standard specification, we have designed and implemented the EIMSaaS architecture as depicted in fig. 4.2.

The architecture consists of two main components that interwork together to glue the session control plane and the multimedia data plane, namely MMAS and MGW. MMAS, deployed at the UAC home network, participates to IMS session signaling and controls the MGW as required by ongoing session dialogs by acting as its stub at the session control plane (steps 1 and 2 in fig.4.2). MGW, at the data plane, works as a middlebox component that splits the (otherwise direct end-to-end) multimedia data path between UAC and UAS (step 3) by enabling several possible advanced multimedia facilities such as adaptation, mixing, and handoff of traversing multimedia flows. For MMAS-to-MGW interaction, we used SIP, given its wide diffusion, also beyond IMS; in addition, this choice

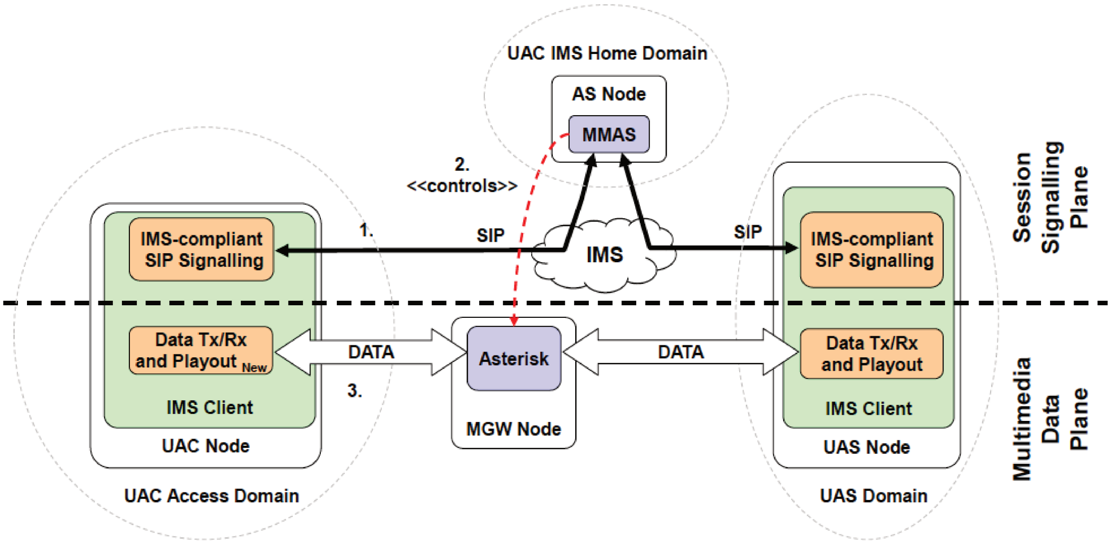


Figure 4.2: EIMSaaS Architecture

simplifies MGW integration with MMAS because MMAS uses SIP to interact with existing IMS frameworks anyway.

4.2.3 The Elastic IMSaaS: Procolol and Main Component

After sketching the general concept and architecture in the previous section, here we fully and comprehensively describe the main components of the architecture and the whole interaction protocol among them. Let us start presenting Fig. 5 illustrating the sequence diagram of the interaction steps operating our MGW elasticity function to improve overall scalability, as detailed in the following. The sequence diagram shows the usage of the MCN.

MaaS to monitor and keep track of the MGW instance load (MaaS exploits the widespread Zabbix monitoring mechanisms). If the monitored indicators exceed the associated target thresholds, the related trigger will activate, and MaaS raises an alarm sending a request towards the NFVO in order to create a new Asterisk instance (steps 20-21). Once the new Asterisk instance is created and ready, the NFVO informs our MMAS about the availability of the new Asterisk instance (step 22); this step is very important and allows MMAS to update its internal state, thus making available the new MGW instance for upcoming new session requests. As new INVITE requests arrive, the data plane integration protocol procedure takes place (steps 23-42) and shares the load between the currently available MGW instances; the load-sharing algorithm may be either static, such as a hash calculated over the incoming UAC user id, or more dynamic, such as taking a decision based on the current load of MGWs. In the example shown in fig.4.2, when new client requests arrive to MMAS, it shares the incoming traffic dynamically and, thus, forwards the requests towards the most newly activated MGW2. Let us note that that the elasticity support through load sharing is typically sufficient, rather than using more dynamic load balancing mechanisms (i.e., able also to move ongoing sessions between old and new MGWs). In fact, load balancing support would require appropriate APIs to extract the current RTP flows state from the old MGW and to install it at the new MGW. Such APIs are

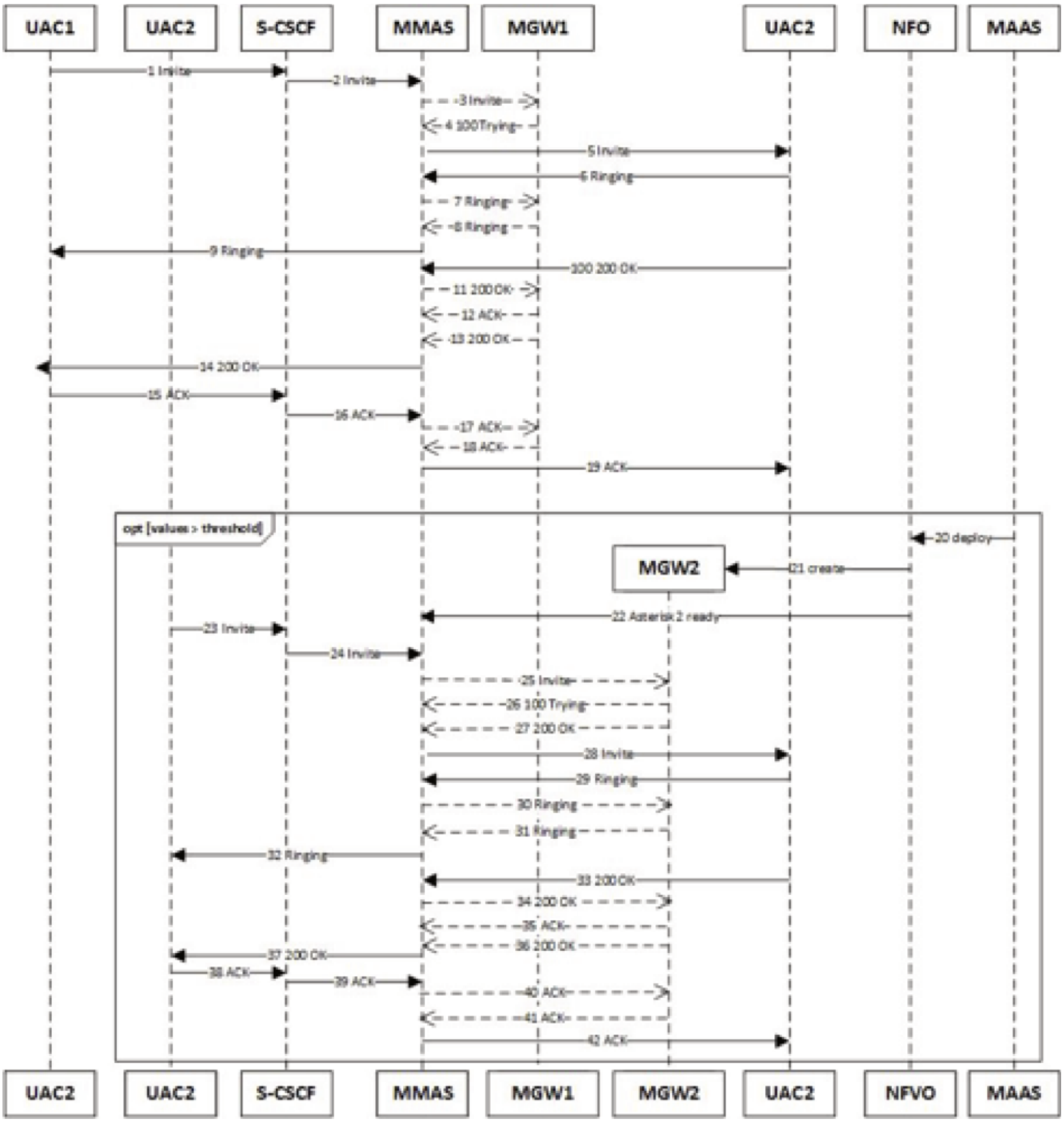


Figure 4.3: The sequence diagram with the primary interactions among the components of our elasticity IMSaaS solution.

currently not supported by most commercial off-theshelf MGW servers, also because it is usually assumed that the session duration is short enough, such as for phone services, to assume that load sharing is sufficient to distribute fast the load in a short time from the old to new MGW, as ongoing sessions at the old MGW terminate. In the following, the focus is led on main involved distributed components at the session signaling and multimedia data planes.

User Agent Client Implementation As depicted in fig.4.3, SIP UAC communicates with our MMAS in order to establish a valid session and then, once this operation is successfully completed, it dialogs with UAS through the Asterisk MGW. In fact, to successfully establish a call, two steps are required. The REGISTRATION is the first operation required. This operation has to be done either by UAC and UAS, respectively to register Callers and Callees. The second step is to establish a valid SIP session; this operation is UAC-initiated (client sending the INVITE message to UAS); the message passes through two intermediaries, firstly the AS, and then the Asterisk MGW. The media streaming is instead a UAS-initiated operation and UAC just “echoes” back the data. SIP UAS has a completely dual role. It receives INVITE messages and, once the session is established, it starts to send the requested data flow. As UAC does, also UAS has to run the registration procedure before starting to establish its SIP sessions. In addition, UAS registers the callees to the Asterisk MGW. After terminating the media flow transmission, UAS has to hang up the call by explicitly sending the SIP BYE message.

MultiMedia Application Server As already stated, our MMAS is providing the main functionalities for achieving elastic media plane. It manages a pool of MGW composed at least by one instance at any time. In fact, at initiation time, MMAS has only one instance of MGW and forwards every SIP message towards it. MMAS receives the SIP messages from the UAC and UAS, it modifies the headers of the SIP messages in order to place itself in the middle of the conversation, and then forwards them to the Asterisk MGW. Concordantly to the IMS standard specification, MMAS does not manage the data streaming: the stream flows from UAS to MGW and from MGW to UAC directly (and vice versa). In case of congestion of the MGW instance, NFVO can request another instance to be deployed and then to be added to the MMAS pool. In the same way, an instance can be removed from a pool in case of low workload. In case of multiple MGW instances, MMAS does not redirect the on-going calls; only the new incoming calls are forwarded to the newly added instances; analogously, in the dual case of instance removal, the “old” instance continues to handle its calls until they terminate

Asterisk Media Gateway In the prototype, for the MGW we opted for the integration of the widely diffused and supported Asterisk server. Asterisk can interact with various session signaling control protocols, e.g., H.323, Media Gateway Control Protocol (MGCP), SIP, and the proprietary Inter-Asterisk eXchange (IAX). AMGW has the task of creating a proper and valid channel for the calls and managing their data flows. The EIMSaaS architecture always includes at least one instance of AMGW (main instance), and it can have more extra instances according to the supported workload. In order to establish a proper media channel, AMGW has to handle also the session/related messages: in fact, in our architecture, AMGW receives SIP messages from our MMAS and creates two different sessions, one between UAC and the gateway itself, the other between the gateway and UAS. The two distinguished sessions are completely independent one from the other, for example it is possible to set two different audio/video codecs for them. The registration phase is very important; in this phase the two agents register all the users capable of calling. The registration of each user (caller or callee) communicates to the AMGW all the necessary information for the communication. For instance, these data include: who the user is, which IP address will be used, if the user is beyond NAT or not, which audio/video codecs the user accepts, etc. When an incoming call arrives, the AMGW first checks if the caller is enabled to call the callee: in the positive case, AMGW tries to retrieve the information about the callee. If the callee is registered, AMGW starts a new session between itself and the callee, by forwarding to it both the SIP messages and the exchanged media stream. As already stated, the decision of scaling-in/-out is taken at the NFVO level, based on monitoring indicators collected by the MCN MaaS. In particular, in the current prototype implementation, also for the sake of simplicity (sophisticated triggering algorithms are put of the scope of this specific paper), we have decided to consider the only CPU average load as the primary monitoring indicator under observation. We use two thresholds for this indicator: one for triggering scale-in and one for triggering scale-out. From the point of view of load-balancing, a simple round-robin algorithm is employed.

4.3 Fog Middleware, a Use Case: ArrowHead Cloud-IoT vertical Architecture for Electro-Mobility Scenario

In this section a real fully use-case of vertical architecture from the top cloud layer to the lowest IoT, passing through the fog middleware. This use case is part of a European project named Arrowhead. This project aimed to provide a framework for the fully integration edge-to-cloud in smart environments, such as smart buildings and infrastructures, electro-mobility and virtual market of energy [149]. My effort was focused in electro-mobility scenario [150]. Following a brief introduction of the Arrowhead project, its framework, and the context in which the use-case is placed.

4.3.1 Arrowhead Project: A General Overview

Arrorhead Framework is a complex fully vertical cloud-fog-IoT architecture for the IoT automation, result of the vast and comprehensive Artemis European project Arrowhead [151].

Arrowhead is addressing efficiency and flexibility at the global scale by means of collaborative automation for five application verticals. That means production (manufacturing, process, energy), smart buildings and infrastructures, electro-mobility and virtual market of energy. The main goal of the framework is to enable collaborative automation by networked embedded devices. The grand challenges are enabling the interoperability and integrability of services provided by almost any device. The assumption at the base of this project is that, a service-based approach will be the feasible technology that enables collaborative automation in an open-network environment connecting many embedded devices. The success of Arrowhead technology depends not only on the technology but also on the capability to create and pursue innovations supported by the core of our technology. If successful the approach is expected to strongly contribute to very significant reduction, 75% or more, in the design and engineering efforts for the predicted multi-billion networked devices. Arrowhead is facing both energy and competitiveness challenges. These challenges are tightly linked and require new dynamic interactions between energy producers and energy consumers, between machines, between systems, between people and systems, etc. Cooperative automation is the key for these dynamic interactions and is enabled by the technology developed around the Internet of Things and Service Oriented Architectures. Under these preambles Arrowhead project poses the following goals to address the technical and applicative challenges associated to cooperative automation:

- Provide a technical framework adapted in terms of functions and performances
- Propose solutions for integration with legacy systems
- Implement and evaluate the cooperative automation through real experimentations in applicative domains: electro-mobility, smart buildings, infrastructures and smart cities, industrial production, energy production and energy virtual market
- Point out the accessible innovations thanks to new services
- Lead the way to further standardization work

The Arrowhead project focus its effort in five macro fields, namely Pilot Domains, such domains are:

- Production
- Smart Building and Infrastructure
- Electro-Mobility
- Energy Production and End-User Services
- Virtual Market of Energy

Successively, the focus will be laid on the Pilot Domain in which i was directly involved, the Electro-Mobility. This domain contains the aforementioned use case of the fully vertical cloud-to-edge architecture for IoT integration [152].

Electro Mobility: a Brief Introduction

Electro mobility is one of the application field in which Arrowhead project is demonstrated. In particular, following an increasing penetration of electric vehicles (either full electric or plug-in hybrid) in the overall mobility market, an increasing energy consumption for recharging such vehicles has to be foreseen and managed in near future. So far, the aim of the project is to include all electro mobility related information and services in Arrowhead Framework, in order to benefit of the interoperability and integrability of the services conceived within such framework. In this scenario, three real world recharge environments and situations have been identified and will be implemented in three real world demonstration sites:

Slow recharge station in private environments This environment consists of a slow recharge station for plug-in electric vehicle recharge. This is the currently most common recharge way, as it is already implemented for private and public AC recharge spots. This application will include end user services such as availability and booking, as well as a possible planning and flexibility management of the recharge event and a remote monitoring of the charging spot itself.

Autonomous recharging stations featuring slow and fast recharging This recharging environment consists of a dual mode recharge station for plug-in electric vehicle recharge, capable of either recharge mode: slow AC and fast DC recharge. This activity includes the development of a recharging spot with an autonomous power source (photovoltaic) and an internal storage system (batteries) in so far that the power interfaces to the energy network is the same as a low power recharge station but it is capable of fast, high power DC recharge. This application will include the same end user services as the slow recharge application plus other energy management features related to the energy storage and fast recharging capabilities.

Longer terms recharging infrastructure Finally, this environment consists of contactless charge-while-drive station. This is a brand new concept that allows an electric vehicle to be recharged while driving over a properly equipped part of the road, without stopping the vehicle. This application will be demonstrated on a short road section in a closed area in which the proper power transfer equipment will be installed. This demonstrator will include end user services such as recharge equipment characteristics (capabilities and details for proper electro-magnetic coupling), availability and booking, as well as a possible planning and flexibility management of the recharge event and a remote monitoring of the charging spot itself.

In the appendix C, a larger and more accurate introduction and overview of the Electro Mobility context is provided.

4.3.2 Electro mobility System of Systems

Electro mobility System of Systems (EM SoSs) are complex SOSs composed by heterogeneous entities that interact, cooperate, and interoperate to provide global services for electro-mobility, including electric vehicle recharge booking and management, charging infrastructure monitoring and maintenance, route planning [153], and EM services oriented to analytics. EM SoSs are based on the Arrowhead Framework for the publication of EM services and for the creation of multi-domain application based on services published by third parties. From a system architecture point of view, EM SoSs are composed of four macro-components that will be described in detail in the following sections:

- The electric vehicle and the recharge infrastructure;
- The Arrowhead SOA electro-mobility solution;
- The Arrowhead Framework EM services;
- The modular co-simulation platform.

The proposed EM solution focuses on three use cases:

- Vehicle recharge “on the move”;
- Vehicle recharge in a private environment;
- Recharge stations in rural environments.

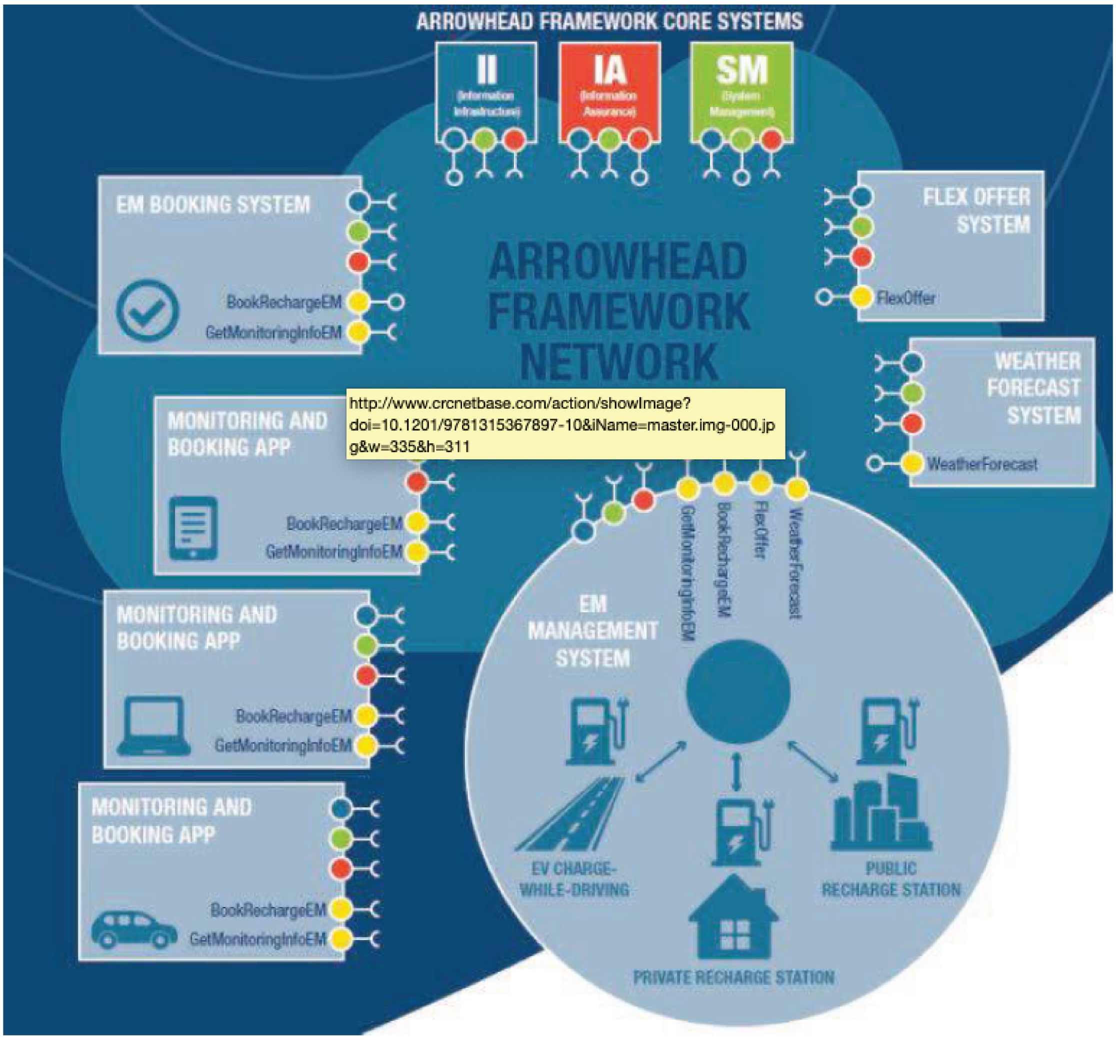


Figure 4.4: System of Systems (SoS) Overview

4.3.3 Electric vehicles and recharge infrastructure

Electric vehicles and charging stations represent the main subsystems of the distributed EM electrical infrastructure. Within the vehicle, the battery, the vehicle main control unit, and, depending on the vehicle, a power converter are the components involved in the recharge process; therefore, they are the vehicle subsystems that interact with the rest of the recharge infrastructure. The amount of energy stored in a battery system can be very different from vehicle to vehicle, according to the vehicle purpose, performance, range, and battery characteristics. The battery's general behaviour can be described by the variable state of charge (SOC), an estimated value that spans from 0% (battery fully depleted) to 100% (battery fully charged). Furthermore, every battery system has its own operating limits, in terms of maximum and minimum operating voltage and operating current, that depend strongly on each battery's characteristics. These limits can never be overcome in any condition. Recharge is mainly managed by a power converter that connects the energy grid (typically AC) and the vehicle battery being charged (a DC storage system). Such a converter is thus mainly an AC/DC converter that can be located either on board a vehicle or off board, depending on its characteristics and power capabilities. From the electrical point of view, the recharge sequence can be roughly described as a sequence of two phases:

First Phase It allows charging the battery from any starting level up to 70–80% SOC level, and is the constant current phase, in which the charger acts roughly as a current generator. The current value is defined by the charger capabilities (power) and the battery acceptance capabilities. The maximum voltage level is defined by the battery characteristics. During this phase, the battery voltage increases up to this maximum level. When this voltage level is reached, then the charger continues with the second phase.

Second Phase It is needed to reach the full charge level, is the constant voltage phase, in which the charger acts as a voltage generator. The current is regulated (i.e., decreased) in order to keep the target voltage level without exceeding it. This phase ends when the current needed to reach the target voltage level decreases to 0 (i.e., full charge condition).

Of course, the charge sequence, from the electrical point of view, is not so straightforward and needs an information exchange between the battery and the charger: the details about this process are out of the scope of this section. In addition to the sequence previously described, it is worth remarking that a charge sequence has to be performed "as a whole", which means it cannot be interrupted but should not be segmented, i.e. once it has been interrupted it should not be restarted. A typical charge profile, regardless of current or voltage values, is depicted in fig.4.5.

It must be highlighted that this kind of profile is applicable to every kind of battery, regardless of the charging technology or charging mode, as described in various international standards. It is a physical/chemical behaviour that cannot be modified. From a simplified point of view, the battery voltage can be described as the sum of two contributions: one depending on SOC (the higher the SOC, the higher the battery voltage) and one depending on the current (coherent with current sign, so increasing voltage while charging

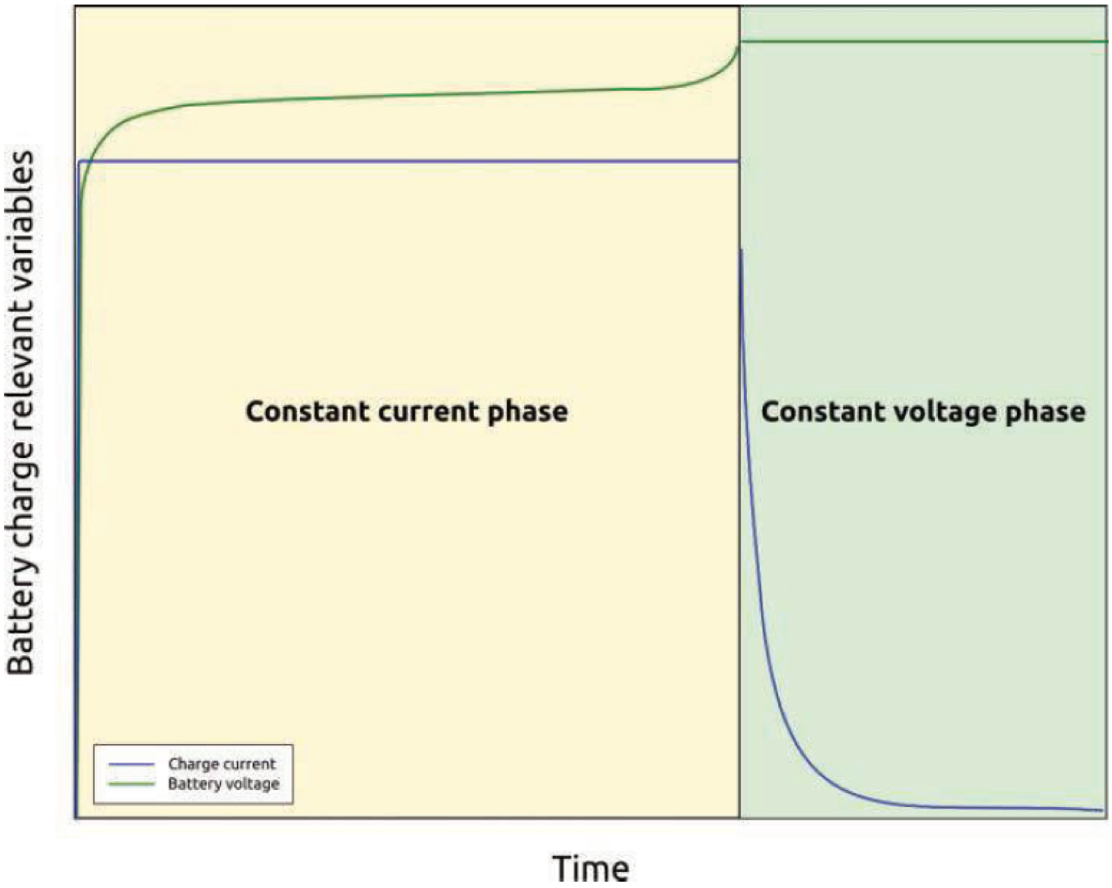


Figure 4.5: Typical Recharge Profile

and decreasing voltage while discharging). So far, the higher the charging current, the quicker the voltage limit is reached, thus preventing achieving a higher SOC level with that current. For a higher SOC level, current has to be decreased, thus slowing down the fast charging process. This means that a full charge can only be achieved by decreasing the charging current as soon as the maximum voltage is achieved. This means also that the charging sequence can be shortened only by increasing the charging current during the constant current phase. So far, fast charging will not be able to achieve full charge of battery, but only some 75-.85% SoC. Furthermore, within these SOC values, the higher the charging current, the lower the SOC level achievable. The technical aspects of the charging process provide the requirements for the identification of significant use cases that represent the state of the art of charging technologies. Starting from these premises, in the Arrowhead project three different recharge use cases have been identified:

- AC slow charge for domestic electric vehicle supply equipment (EVSE);
- DC fast charge for public EVSE in remote locations;
- Contactless charge on-the-move.

These use cases are part of one of the project pilots, because each of them corresponds to a physical pilot showing the usage of the three different recharge technologies. The three recharge scenarios are not intended to be comprehensive, covering all the possible charging scenarios or technologies, as well as it is not intended to go through all the standardization efforts currently running in Europe and worldwide about EV charging scenarios. AC slow charge for domestic EVSE is the most common charge type. It is available to every electric plug-in vehicle on the market. The battery charger is quite compact, is limited to 16A or 32A (at grid level), and it is hosted onboard. The charging time is thus quite long, i.e., several hours, and the EV charge is typically intended to be performed overnight, to take advantage of off-peak low energy rates. DC fast charge for public EVSE in remote locations (i.e., rural areas) is a particular case of DC fast charging that involves a new generation of battery charger that is capable performing a fast DC charge also in situations (remote locations) in which grid capabilities don't allow it. Such over performance is possible by adopting a battery storage system used to accumulate the energy needed for a fast charge. Such a storage system can be charged either from the power grid or from a photovoltaic system also present on site. The battery storage system accumulates energy while no vehicle is charging and then will be quickly discharged during the vehicle recharge process. Of course, charging the auxiliary battery system will take a much longer time than discharging, due to grid or photo-voltic (PV) intrinsic capabilities, so the fast charge will be available only few times a day, according to the battery storage system capacity and to the total amount of energy requested by the incoming vehicles. Some additional considerations regarding fast charging technology are:

- fast charging technology is available only for properly equipped EVs;
- the power converter for fast charging is typically a huge, high-power AC/DC converter located off vehicle, permanently connected to a power grid
- the charging interface to a vehicle is in DC.

The contactless charge on-the-move is a brand new charging technology that exploits jointly two novel concepts. The first is the contactless recharge, which adopts wireless power transfer technology to establish a power connection between one or more transmitting coil(s) located in the road, under the tarmac surface, and one receiving coil located in the lower part of the vehicle. This coil-based system transfers energy from the grid to the vehicle's battery without a wired connection. The second technology is the application of the coil-based approach to a moving vehicle, thus having a series of coils in the road, being activated and deactivated in sequence as soon as the vehicle being charged transits from one coil to the next one. The three recharge use cases have been studied in the context of the Arrowhead project, and three distinct but interoperable pilots have been designed and implemented. The pilots are based and natively integrated within the Arrowhead Framework by means of Arrowhead EM services, which are the main topic of the current section. Starting from the vehicle batteries and the charging stations, the automations of the EM complex system will be achieved through an end-to-end IoT cloud-based solution, an ICT infrastructure of hardware and software components that exploits the state of the art of pervasive systems, telemetry, cloud computing, and service-oriented platforms.

4.3.4 Arrowhead SOA EM solution

The core of the EM SoSs is a service-oriented end-to-end automation solution based on the Arrowhead Framework, on IoT technologies, and on cloud computing. The proposed solution has been successfully applied to EM real contexts identified in the three main use cases: private recharge, public recharge, and recharge on the move. The evaluation of the systems and of the application services, implemented to address the use cases, demonstrated that the required interoperability level among the hardware subsystems, the software pervasive subsystems, the services, and the cloud has been obtained. The analysis also showed that the implemented systems and services, by cooperating together, originate a tangible added value perceivable by both stakeholders and by final users, a positive indicator for the general acceptance and interest for EM. The defined EM automation software architecture may hire an unbounded number of interacting services and systems, as it has been defined to be extensible; thus only the most relevant considerations about system interaction in the defined use cases of interest will be analyzed in detail. Fig.4.6 shows the main components of the EM automation infrastructure from a high-level perspective. The EM cloud platform constitutes the core of the architecture where the information from all the data sources is stored and shared with the other actors and services, according to privacy and information access policies enforced by the cloud management system. Sensible information to be managed in the cloud may come from different sources, i.e., electric vehicle onboard equipment, private or public buildings, energy storage systems based on renewables, charging stations, power grid components, or meters. The cloud platform publishes the EM-Management services on the Arrowhead Framework and uses the services provided by the framework (i.e., the EM-Booking service, the WeatherForecasting service, the FlexOffer service, etc.). With this approach, the technical details are completely hidden and the functionalities of the EM SoSs are efficiently made available to other subsystems of the EM SoSs or to other cross domain applications. At the same time, the EM SoSs, interacting through the cloud with the AF, can exploit the services provided by other domains. The charging stations represent the edge of the EM SoSs and exploit IoT technologies to interact with the cloud platform that, in turns, “converts” their functionalities and collected data in Arrowhead services. The charging stations adopt the Eclipse Kura IoT framework [154] for a pervasive integration with the cloud platform. Kura runs on a control unit inside the charging station that, acting as a multiservice gateway, provides the following functionalities:

- A hardware abstraction layer that simplifies the business logic development;
- Wide support for data collection from the field;
- Edge computing services for local data processing;
- Efficient and secure cloud client that supports MQTT-based telemetry;
- Remote management functionalities that provide full remote control through the entire life cycle of the charging station.

The same pervasive IoT solution has been adopted for all the use cases. The clients of the envisioned EM infrastructure may be fixed workstations, mobile applications or on

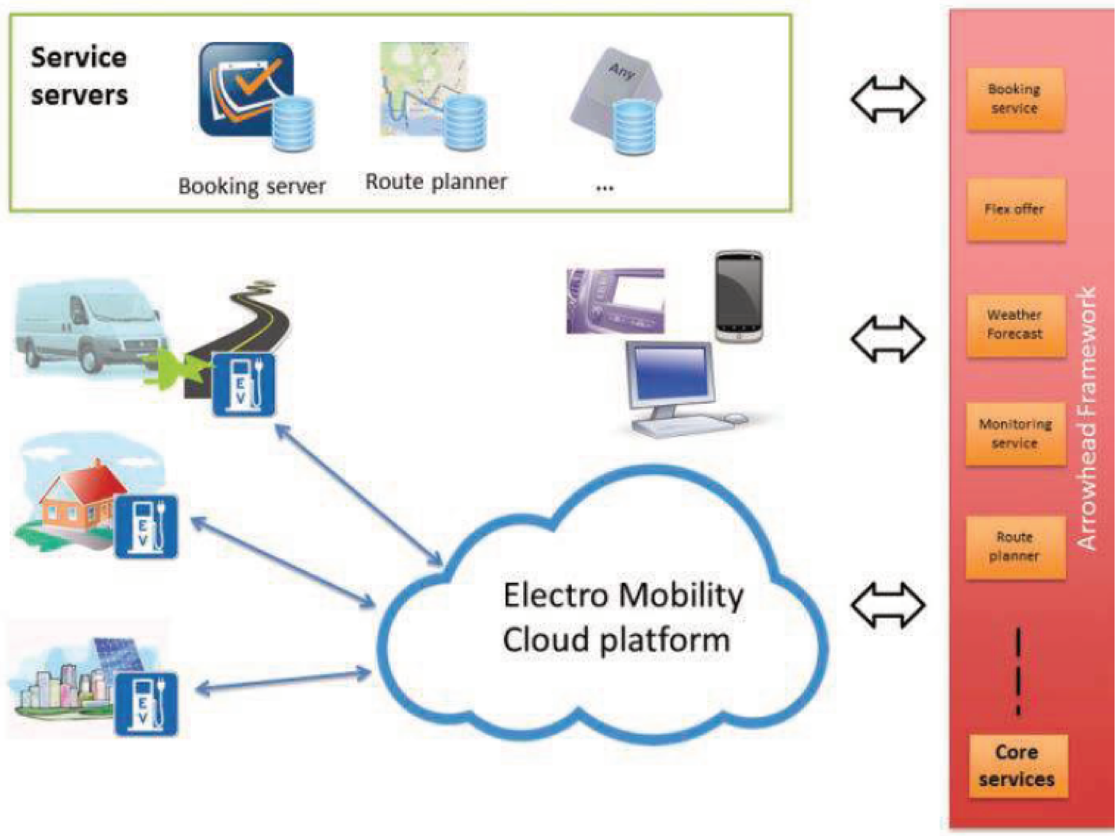


Figure 4.6: EM Scenario: Automation Infrastructure (high-level perspective)

board computers providing services on the move. The services may be schematized as entities with a client side, a server side, and a knowledge base on which they act. All these entities need precise information about the other subsystems in order to interact (i.e., addresses, access rights, information models, protocols, synchronization policies, etc.). The process of adapting the existing heterogeneous and multivendor components in order to properly interact and cooperate in common scenarios is difficult, expensive and implies many non-trivial challenges. In a similar complex scenario, the Arrowhead Framework represents the technological glue that connects together the data sources, the cloud infrastructure, the client equipment, and the services by requiring only simple elements. That is service publication is bound only to the requirements imposed by the authorization service (in order to be trusted) and by the service registry (in order to be discoverable and to discover). The application of the Arrowhead Framework to the EM scenario provides many advantages, among which is worth mentioning the separation of concerns through which each entity is given the opportunity of sharing only the data required to achieve service interoperability. All the other collected information remain completely hidden. As it is possible to see in fig. 4.6, the architecture of the EM infrastructure has been designed to be extensible with new services.

4.3.5 Systems and Services

The possibility to share system characteristics, features, and functionalities through open and interoperable service represents one of the key advantages of the Arrowhead Framework. It enables the simple creation of cross domain service-based application that fully exploits the potentialities of complex System of Systems. With this philosophy, the EM scenario becomes an ecosystem where multi-vendor and multi-domain solutions cooperate to provide new added value services to heterogeneous users and stakeholders. The artifacts introduced in the EM scenario to create this ecosystem are (fig. 4.7) the electro mobility management system (EMMS), a set of services specific of the EM domain and a set of services from other application domains. The systems and services that have been involved are:

- EM Management System;
- Booking Service;
- Route Planner Service;
- Weather Forecast Service;
- FlexOffer Service [155].

The EMMS is the Electro Mobility System of Systems and is responsible for managing and monitoring the charging infrastructure deployed on the field, to manage its interaction with the cloud and to provide the EM service to the Arrowhead Framework. The services from the other application domain cooperate with the EMMS to create the added value application, that have been investigated and demonstrated in the EM use cases. More specifically, the actors and the entities involved in the use cases, through the Arrowhead Framework, are those defined in the three recharge scenarios illustrated in section 9.3.4.3.2 The first example of cross domain service usage is the booking service. The importance and the value of the booking system can be evaluated from three different perspectives. From the user point of view, the booking service allows booking a recharge directly from a mobile device or from the charging station, at home or almost everywhere, also in rural areas, and this feature contributes to the reduction of driver anxiety, one of the biggest obstacles preventing the medium user from choosing an electric vehicle. Due to the long recharging time, it is paramount for users to be able to plan long travels considering the various recharge options. From the charging infrastructure point of view, situations in which users crowd specific charging spots, leaving others underused, will give way to a more balanced and sound resource usage as the customers will be distributed on the available spots. Furthermore, this service allows collecting data that will enable the charging station owners to estimate the future demand and properly manage the local resources to face future usage trends. The booking service provides also an estimation of the minimum number of vehicles that are going to use the charging station, and this information is particularly relevant in the public recharge scenario, with local energy storage system. The public charging station, in fact, has to select the most convenient way to recharge the local storage; two options are available: exploiting solar energy with no added cost or relying on the power grid with the consequent costs, a faster method that is mandatory if there is the risk of completely depleting the local storage. A third perspective under which the booking service may improve the EM scenario is that of the energy provider. The impact of EM on the electric distribution grid is relevant, as demonstrated by several scientific research papers [156, 157]. The power grid has different working modes: depending on the total demand it may range from high efficiency with low demand to low efficiency and high costs with high demand. In this context, the data collected by the booking service from the charging stations allows the energy distributor to improve the power demand estimations and so the power grid component usage, the costs, the quality of service, etc. As anticipated, other services may improve the EM use case, if properly immersed in the AF ecosystem. The weather forecast service can be used in the public recharge scenario to improve the estimation of how much energy it is possible to store through solar panels in the coming hours. A route planner service may advise routes with wireless power transfer equipment in order to augment the probability of completing long trips without the risk of total battery depletion. The EM-Management system provides monitoring and management services that can be exploited at two different levels: directly in the electro mobility scenario and/or by other applications in other domains. The basic role of the EMMS is the orchestration of the distributed charging infrastructure: the main benefit is a shared solution based on IoT and cloud technologies for the integration of every single use case in the electro mobility scenario. This benefit is available at the scenario level and in other domains through EM services. For example, EM services are fundamental for the booking service in order to avoid the reservation of a charging station that is out of service or of a charging station on which a maintenance intervention has been planned. Two important areas of interest for the potential creation of cross domain applications are maintenance and multivendor intercharge. Maintenance activities can have greatly benefit using the EM-Monitoring service, because the remote control of charging stations simplifies their life cycle management, reducing the deployment and maintenance costs with provisioning functionalities, remote monitoring, remote management, and preventive maintenance. Intercharge is another example of cross domain application: EM services provide a common and shared interface that could connect various e-mobility players and provide EV drivers with simple and customer-friendly access to heterogeneous charging infrastructures. Different charging stations managed by different companies become part of the same network, and the new System of Systems does not only hide the complexity caused by this heterogeneity but offers to the drivers the same interface to receive real-time information about the status of the charging station or the charging process. Finally, the FlexOffer service may be used in the private recharge scenario to propose to the end user the best energy cost, thanks to a proper allocation of the effective recharging time, in a time interval where the power grid is not congested or, simply, the contract offers lower prices.

It is worth noticing that the contribution of the services to the quality level of the target scenarios, that is, the concrete added value, is achievable only if the exchanged information is trusted, if the services are able to seamlessly find and use each other's functionalities, and if a good grade of orchestration and cooperation among the services is supposed: in short, if the Arrowhead Framework core services are correctly used to manage the interaction among actors. For example, the booking service only allows booking of a charging stations if the charge consumption data can be received by charging spot owners or energy distributors. Moreover, the quality of service offered will be lower if the booking service can't use the knowledge base of the monitoring service to improve its reliability, e.g., to avoid reserving a charging spot currently in maintenance or to properly advise the end user and propose other recharge options when a fault happens on a previously

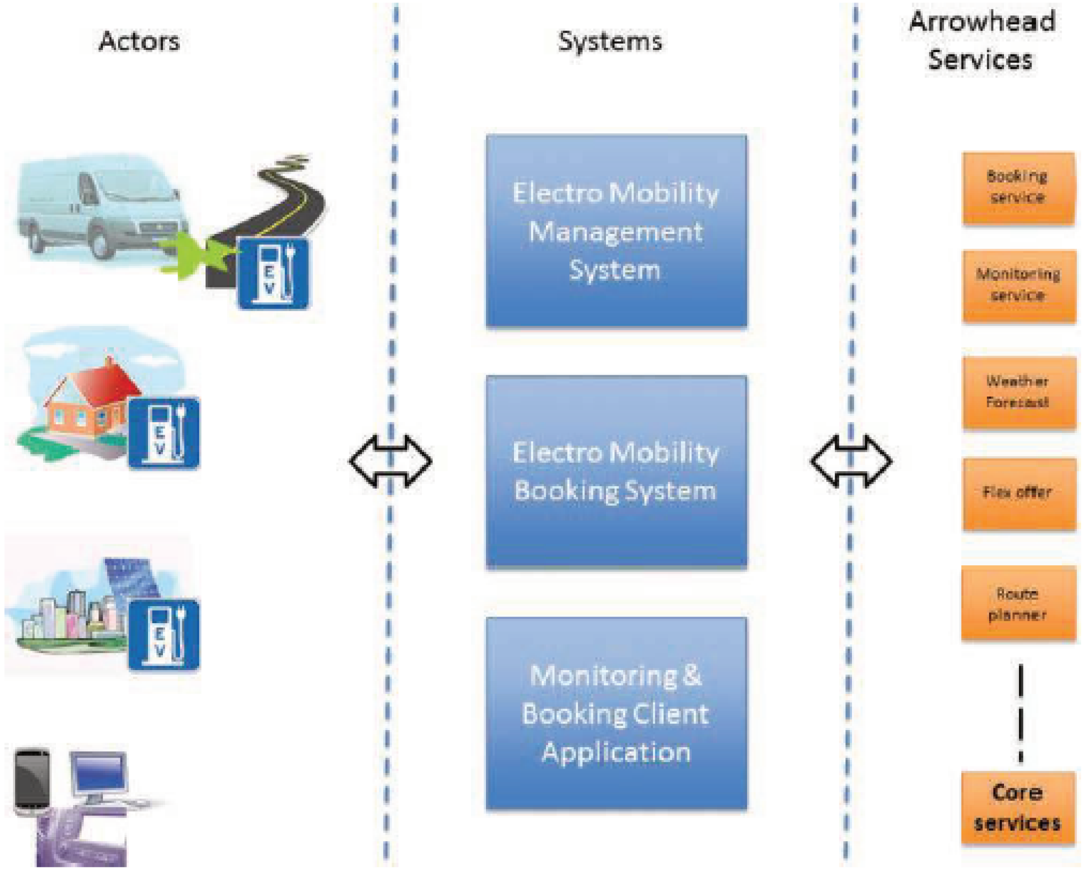


Figure 4.7: Automation Systems and Application Services in the Arrowhead Project for the EM Scenario

reserved charging spot. Similar considerations are valid for the route planner and also for the other services involved in the electro mobility scenario. The route planner, indeed, needs updated information about the new charging station installed on the territory, while the energy distribution company needs the data of the route planner to derive statistics from the user needs and find optimal positions to install new charging stations or charging stations on the move. Each service involved in the electro mobility scenario is provided by a specific subsystem; therefore, there are no specific security issues due to service interactions: the data shared will be only those exposed through the Arrowhead Framework and the interacting entities will communicate using messages encrypted on the basis of security certificates validated by the Arrowhead Framework Authorization service. Fig. 4.7 shows that the integration in the Arrowhead big picture is not invasive as only a REST interface has to be provided for the subsystems to cooperate.

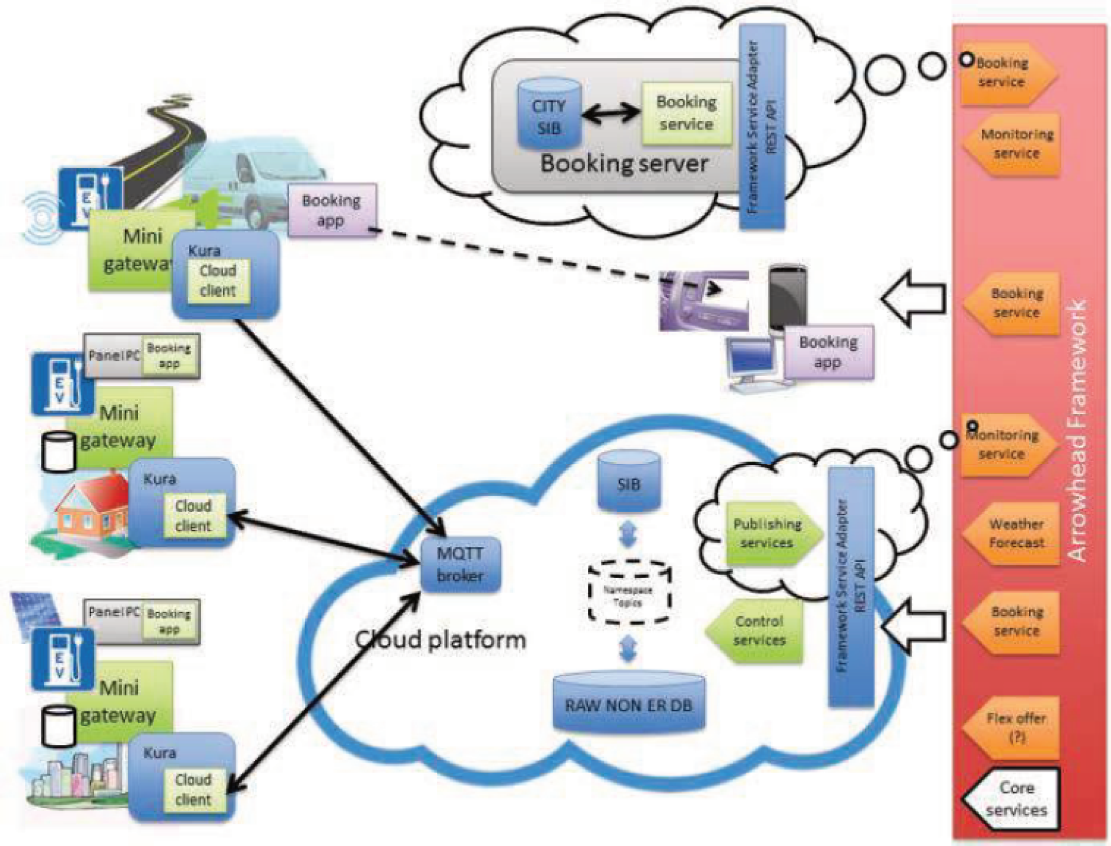


Figure 4.8: EM scenario: Interoperability Components and Software Localization

In regard to the cloud, the main integration components are the publishing and control services, defining which data to share and with whom. The information coming from the charging infrastructure is collected adopting a device-to-cloud approach based on the Eclipse Kura framework and, with the MQTT telemetry protocol, is sent to and stored in the cloud. The raw data are made interoperable and machine understandable through a semantic annotation process using a semantic information broker (SIB) to manage the resulting semantic assertions. Smart-M3 [158, 159] has been adopted as a middleware for interoperability. It is a semantic publish-subscribe data-driven interoperability platform, which stores data encoded as Resource Description Framework (RDF) triples with respect to ontological rules. The data stores are called semantic information brokers

(SIBs) [160, 161] and provide APIs in order to perform operations such as insert, update, delete, query, and subscribe, all of them encoded with the SPARQL [162] or RDF-M3 formalism. Analyzing and comparing Fig.4.8 and 4.9, it is possible to understand how the integration allows two of the main EM subsystems to cooperate. The booking server was a standalone legacy service that has been adapted to the framework through a properly designed REST interface. This interface can be used by both end users wanting to reserve a charging station and the charging station equipment to check the user identity. This second interaction starts from a control unit installed on the charging station which starts an information exchange through the Kura-based device to the cloud platform; the request triggers a cloud internal control service which relies on the Arrowhead Framework to discover the correct instance of the booking service and to check the user credentials. A failure of this check prevents the reservation of the recharge process from starting. The cloud publishing services are used when information about entities connected to the cloud is needed by external agents. The cloud internal logic manages the calls to the *Find Aggregated Data* primitive, a function used by the EMBooking service (after discovering the EM-Management service) to detect malfunctioning charging stations in a given area. A reply is provided only if the caller has the right to access such data.

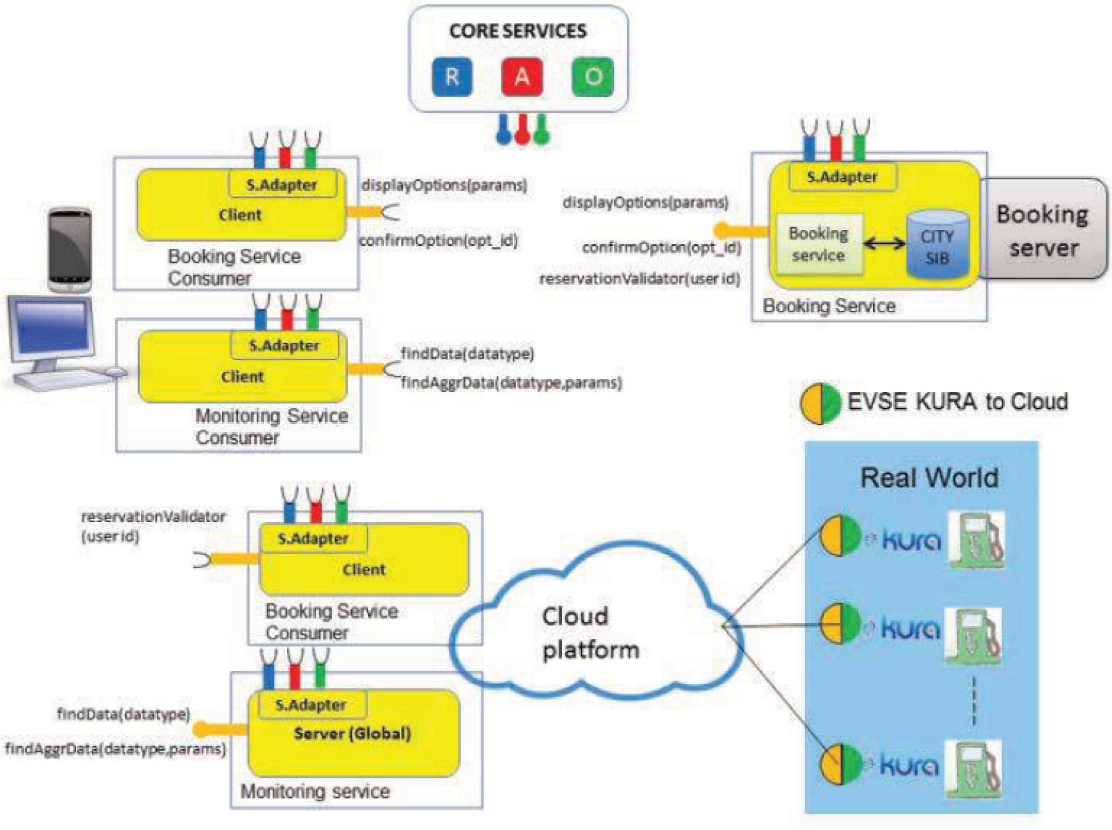


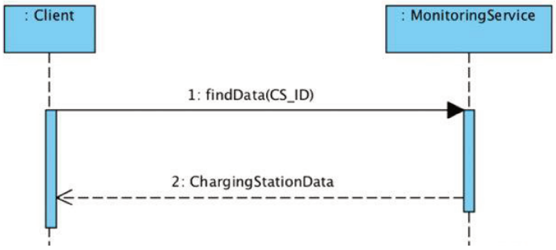
Figure 4.9: Booking and Monitoring Application Systems

4.3.6 Arrowhead EM services and related automation aspects

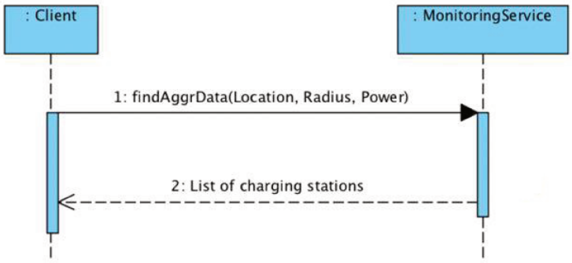
This section describes the arrowhead services designed and implemented in the electromobility scenario. The description is based on the main sequence and class UML diagrams specified in the documentation artifacts available online on the project website.

EM-Monitoring service

The EM-Monitoring service provides two primitives for the interaction with the charging infrastructure (see fig.4.10): findData and findAggrData. The findData primitive is used when an EM subsystem or a third party application needs information about a specific charging station. Using this primitive, a client is able to query the monitoring service about the whole status of an EVSE by simply specifying its identifier in the findData primitive. The findAggrData primitive, instead, is used when the EM agent needs information about a group of charging stations sharing some property, or when it is necessary to refer to a charging station indirectly, through metadata, instead of directly through an identifier. This primitive is particularly relevant because it provides information that is useful in the most common EM use cases. This simple interface generates a rich set of information that corresponds to conceptually complex queries: for example, calling the findAggrData primitive it is possible to identify all the charging spots in a circular area geographically centered on a specific point with a specific radius, identify the closest charging stations where it is possible to recharge 40 kWh in less than 3 hours, or simply detect if a fast charging station is currently in operation.



(a) EM-Management system: Information Retrieval for a single Charging Station



(b) EM-Management system: Information Retrieval for Multiple Charging Station

Figure 4.10

EM-Booking Service

The booking application service operates according to a handshake strategy between the client and the server module. First, the client asks for possible booking options by specifying its preferences in terms of space, time, and amount of energy needed. Then the service, according to its business logic and to the requirements specified by the user, provides a list of charging options. At this point the user may choose one of the provided options, finalizing the reservation, or make a different request. It is not uncommon that, by scaling the end user number, reservation conflicts due to overlapping reservation on the same EVSE at the same time could represent a concrete issue. In this case the user which confirms the last the reservation is informed that the booking option is not available anymore due to a concurrent booking from another user and is invited to choose another option or make a new query to the booking service.

4.3.7 Co-Simulation platform

Energy and smart cities scenarios are considered among the most interesting and powerful environments in which automation systems and services prove their usefulness. However, such systems are often tricky to demonstrate, since the required architecture, especially the hardware components, is not in place. This also motivates simulations as both validation environments and pre-deployment analysis tools. The University of Bologna developed a co-simulation framework capable of depicting an urban scenario with vehicular traffic, electro mobility, charging stations, and smart grid model [163]. In particular, the developed simulator allows the analysis of urban traffic, including a significant percentage of electric vehicles and clusters of charging stations (in particular, fast charging stations concentrated in public parking lots). The co-simulator platform described in this section has been created in order to meet the following goals:

- Design a distributed control able to mitigate the congestion in the network caused by an excess power demand by the recharging of electric vehicles;
- Test the infrastructure from the services' point of view, both as a pre-deployment analysis and a scalability and benchmark test;
- Give to the developer the possibility of building automation systems on top of a sandbox which resembles a realistic scenario.

This section provides a glimpse of the architecture of the co-simulation platform and of its components as well as its Arrowhead services, illustrates the analysis carried out by using the simulator through the presentation of some results, and depicts the possibilities for automation systems and services operating within such an ecosystem.

Architecture of the co-simulation platform

The co-simulation platform integrates the traffic simulator and the power distribution simulator. Fig. 4.11 illustrates the general architecture of the co-simulation environment. Smart-M3 has been adopted as a middleware for interoperability. In such a scenario, the city SIB collects information regarding all the entities coming from different domains, while the dash SIB collects data about the vehicles.

Traffic Simulator

Urban traffic is modeled using VeinS, which is an open source framework for vehicular network simulations based, in turn, on two simulators, namely, discrete event-based simulator OMNeT++ and road traffic simulator SUMO. SUMO (Simulator of Urban Mobility) is

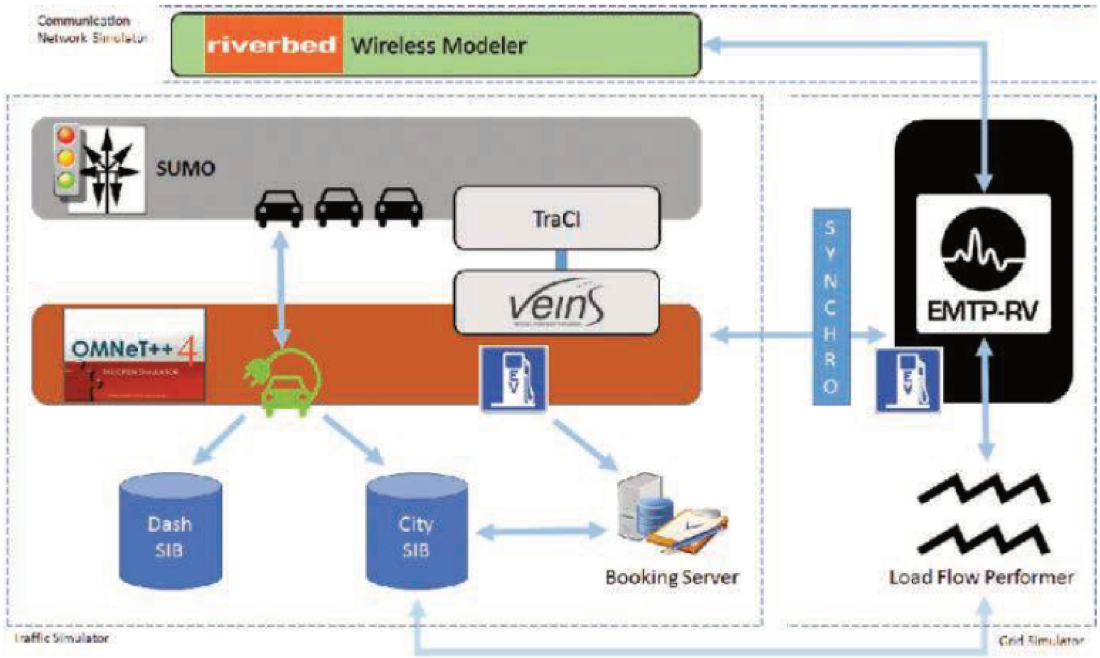


Figure 4.11: Architecture of the Co-Simulation Environment

an open source traffic simulator capable of modeling entities such as roads, vehicles, traffic lights and vehicle routing. Each entity is simulated microscopically; thus it is possible to interact with them separately. OMNeT++ is a general purpose simulation environment for communications, which is able to model customizable and interoperable modules. A large-scale scenario (i.e., downtown Bologna) was considered, with a realistic street map (imported from the OpenStreetMap project). VeinS has been extended with the models of EVs and EVSE units (including the management of the EVs queues) and it has been integrated with the battery charging/discharging models described in [164]. In particular, OMNeT is used to characterize the electric vehicles through the simulation of the battery charging and discharging process, the charging stations, and the previously described reservation mechanism. Depending on the simulated time of the day, a different traffic rate is established (i.e., total number of vehicles running in the scenario at the same time). When an EV reaches a level of battery lower than a predefined threshold it tries to go to recharge. Results show that the use of the reservation mechanism implies significantly less recharging failures.

Power Distribution System Simulator

The power distribution system simulator is based on an EMTP-rv (electromagnetic transient program), which is a time domain simulator. The distribution feeders are modelled including models of the three-phase unbalanced lines, of the three-phase HV/MV substation transformers and of the aggregated unbalanced loads (constant impedance/current/power) that include the EVSE units. The model of the EVSE is based on a triplet of current sources. For each phase the control relies on a feedback regulator in order to inject or absorb the requested values of active and reactive per-phase power, as described in [165, 166]. Each aggregate load is connected to the secondary side of a MV/LV trans-former.

Arrowhead Framework Services

The simulated scenario is intended to resemble the real scenario and the use cases previously defined. In particular, as illustrated in fig.4.12, it implements an instance of the Arrowhead Framework EM-Management system (defined as a requirement for EM scenarios) and it is fully interoperable with the Arrowhead Framework EM-Booking system. Each vehicle in the simulation performs a call to the EM-Booking service whenever it needs to recharge and picks randomly one of the options displayed.

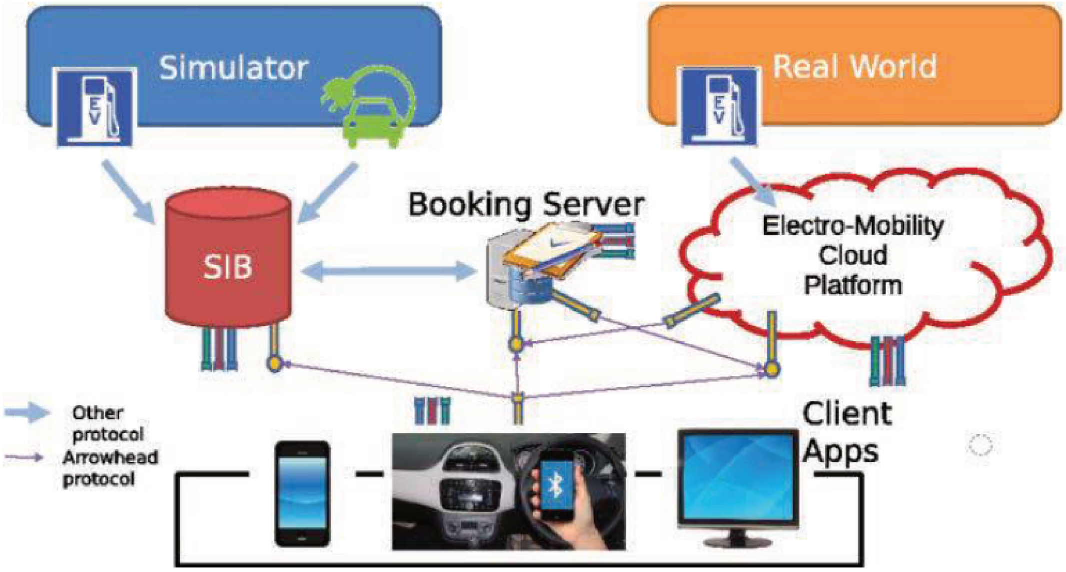


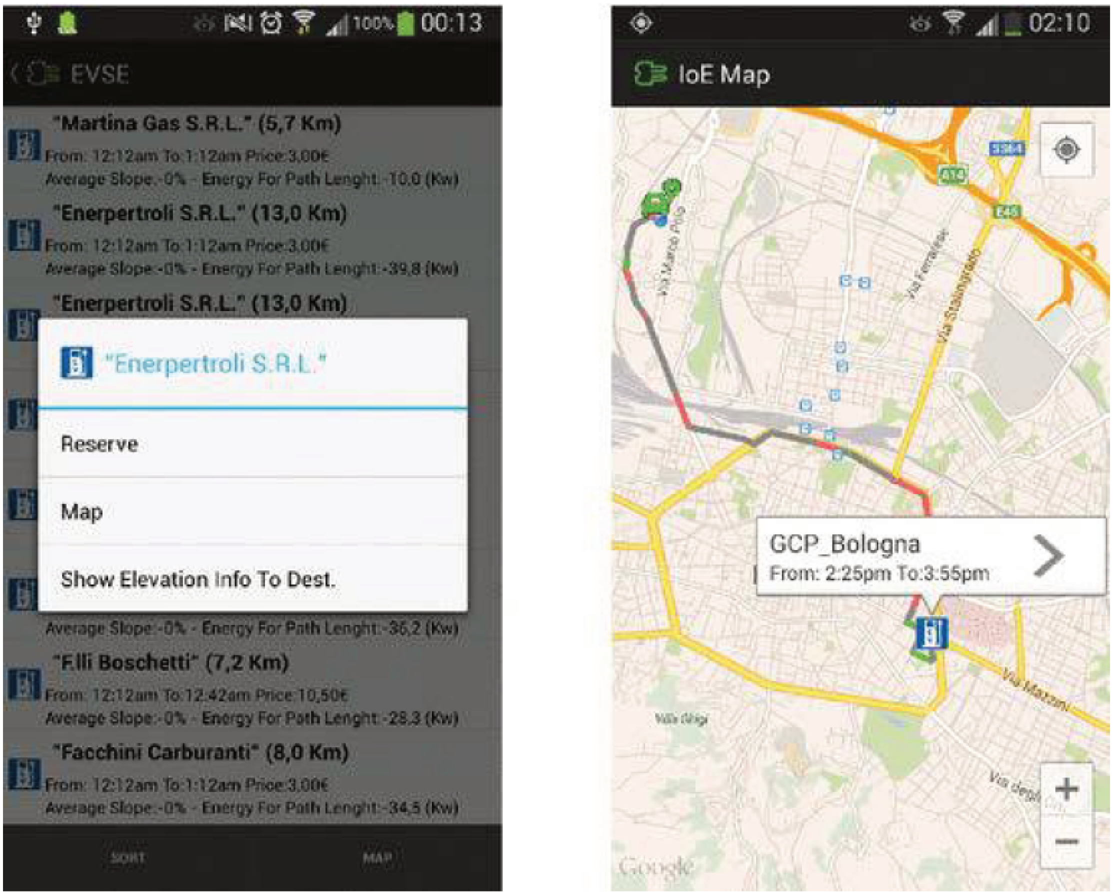
Figure 4.12: Mixed Real-Simulated Scenario

4.3.8 Mobile Service Platforms and Significant Results

In the context of service granularity, in particular when referring to electro mobility, the world of mobile, in-vehicle, and context-aware services has a fundamental role: simplifying the transition to the new infrastructure by both reducing the impact on end users, who feel anxious about new technologies, and also encouraging the transition to EVs. Realistic EV scenarios are characterized by the heterogeneity of the actors involved in them, i.e., different EV models, EVSEs providers and smart-grid operators, each using their data representation techniques, and providing their own services, mobile applications, or APIs for third-party software deployment. In such a chaotic environment the developed mobile services were run, completely unaware of the world that they are running in, meaning that they have the same behaviour when they are deployed in real-world entities or simulated entities. This happens because simulated vehicles can also be controlled remotely through an Arrowhead mobile application, which is indeed the same application that a real user would use while seated in his or her electric car, exploiting both the EM-Booking service and the EM-Monitoring service. Such services show the same interface to consumers regardless of the scenario in which they are deployed. The set of applications having such a property is called “Mobile Application Zoo (MAZ)”. As a result, a mobile application will be able to retrieve and display data of a simulated vehicle (e.g., current charge state), as if the user were driving it. Vice versa, interacting with the mobile application, the user might perform actions that trigger events within the simulation, for instance, a reservation might result in the simulated vehicle changing its route. A MAZ component provides two main advantages. First, it allows testing the correct behaviour of mobile applications under several different conditions that might occur in a real-world scenario. Second, it allows evaluating the performance improvement provided by the utilization of the mobile applications described in the following subsections when all the scenario components are in the loop, i.e., the vehicular traffic, the charging/discharging operations, the impact on the smart-grid.

EM-Booking Service

As in the envisioned real scenario, the EM-Booking service uses the EMMonitoring service in order to get geographical information about the EVSEs nearby and books a recharging time slot in a target area, and during a preferred time frame. In Figure 9.15 is shown screenshots from mobile application client using the reservation service. On the left is the result of a user query, i.e., all the EVSE that match the user preferences and are returned to the client app. When a user selects his preferred recharge option, the application shows the path from the current position to the EVSE and the time at which the reservation is made (fig.4.13 right).



"Martina Gas S.R.L." (5,7 Km)
From: 12:12am To:1:12am Price:3,00€
Average Slope:-0% - Energy For Path Lenght:-10,0 (Kw)

"Enerpertroli S.R.L." (13,0 Km)
From: 12:12am To:1:12am Price:3,00€
Average Slope:-0% - Energy For Path Lenght:-39,8 (Kw)

"Enerpertroli S.R.L." (13,0 Km)

Average Slope:-0% - Energy For Path Lenght:-36,2 (Kw)

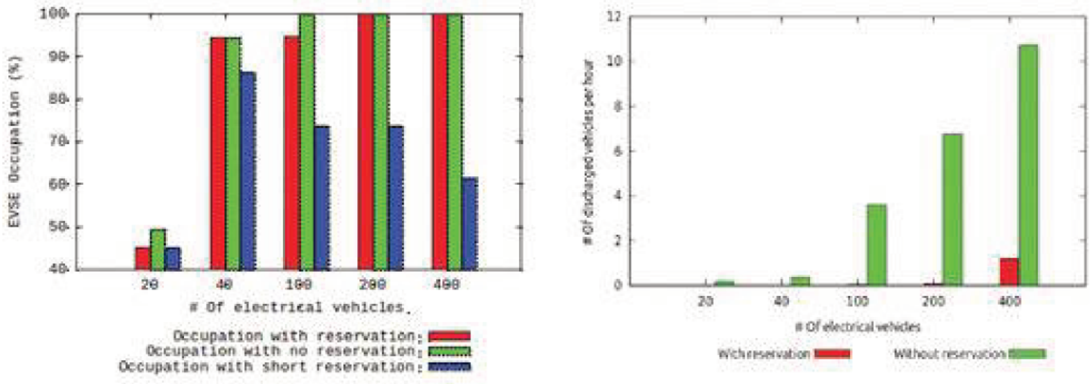
"F.lli Boschetti" (7,2 Km)
From: 12:12am To:12:42am Price:10,50€
Average Slope:-0% - Energy For Path Lenght:-28,3 (Kw)

"Facchini Carburanti" (8,0 Km)
From: 12:12am To:1:12am Price:3,00€
Average Slope:-0% - Energy For Path Lenght:-34,5 (Kw)

GCP_Bologna
From: 2:25pm To:3:55pm

Figure 4.13: Mixed Real-Simulated Scenario

The test of the reservation process within the simulated scenario gave the results shown in Figure 9.16, which depicts the average EVSE utilization when using or not using the reservation service. When a reservation is not used, the simulated EVs seek for an available EVSE, driving around the city. Fig.4.14a shows that the average occupation increases with the number of electric vehicles. The average occupation without a reservation is generally higher than that with reservation because simulated EVs fill the nearest charging spots fast without any coordination. This chaotic behaviour brings battery depletion risks when the number of EVSE is low with respect to that of EVs needing to recharge in a certain zone. In fig.4.14b, it can be noticed that the number of EVs that run out of battery without being able to recharge, is much larger when a reservation is not present. This happens because when all the charging spots are filled, the residual EVs, being unaware of the situation, continue to travel until the residual charge ends. The results provide two important indications for service developers and energy distributors. On one side, they clarify the usefulness of the reservation service, because the EVSEs are utilized in a coordinated manner, which translates into much more efficient scheduling of charging operations. On the other side, they are valid as a pre-deployment analysis, since they provide feedback about the optimal planning of the charging infrastructure, in terms of number and location of EVSEs.



(a) Evaluation of Mobile Services for EVs: Average Occupation

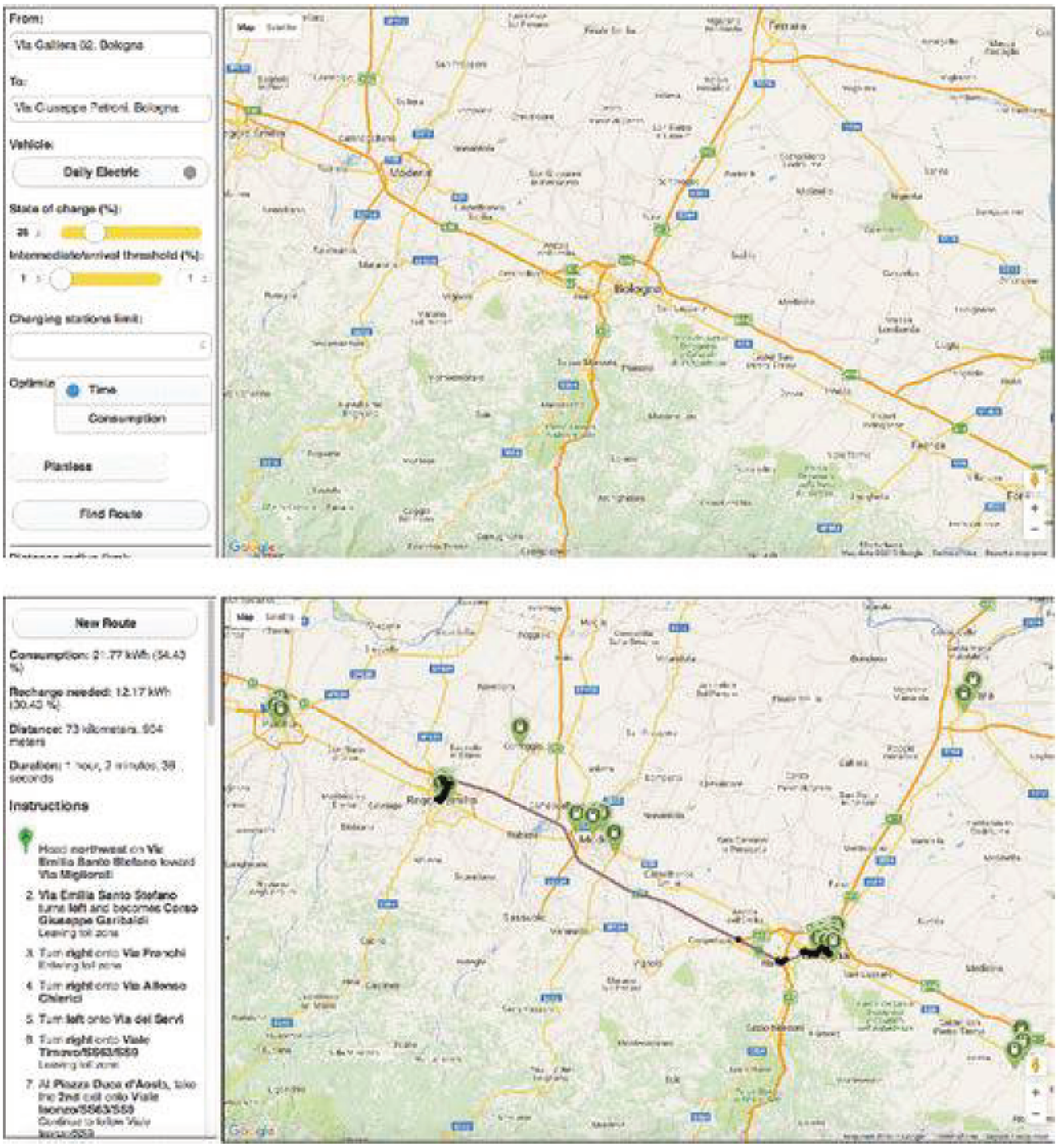
(b) Evaluation of Mobile Services for EVs: Reservation Service Impact on EVs

Figure 4.14: Evaluation of Mobile Services for EVs: EVSE Occupation for Reservation Service.

RoutePlanning service

This section describes the RoutePlanning service, which is depicted in fig.4.15 [167, 153]. One of the most relevant issues of electro-mobility is the uncertainty of successfully traveling through a planned trip. This is due to the complexity of the accurate estimation of the consumption along the path and to the low density of charging opportunities. Hence, a tool to assist the EV driver has been developed: it computes the expected consumption over the desired path, and identifies the needed charging opportunities by minimizing either the total travel time or the total consumption.

The route planner uses EVSE data provided by the EM-Monitoring service and works as follows. First, it computes the expected consumption between the start and the end of



From:
Via Galliera 92, Bologna

To:
Via Giuseppe Petroni, Bologna

Vehicle:
Daily Electric

State of charge (%):
35

Intermediate/arrival threshold (%):
1 1

Charging stations limit:

Optimize: ● Time ○ Consumption

Consumption: 21.77 kWh (54.43 %)

Recharge needed: 12.17 kWh (30.42 %)

Distance: 73 kilometers, 904 meters

Duration: 1 hour, 2 minutes, 38 seconds

Instructions

Head **northwest** on **Via Emilia Santo Stefano** toward **Via Migliorati**

2. **Via Emilia Santo Stefano** turns **left** and becomes **Corso Giuseppe Garibaldi**
   Leaving toll zone
3. Turn **right** onto **Via Franchi**
   Entering toll zone
4. Turn **right** onto **Via Alfonso Chierici**
5. Turn **left** onto **Via del Servi**
6. Turn **right** onto **Viale Timavo/SS63/SS9**
   Leaving toll zone
7. At **Piazza Duca d'Aosta**, take the **2nd** exit onto **Viale Isonzo/SS63/SS9**
   Continue to follow Viale Isonzo/SS9

Figure 4.15: RoutePlanning Service

the path. If a user-defined threshold of intermediate charge is satisfied, then the system simply returns the desired path with the directions to reach the destination. The threshold, called Intermediate State Of Charge (SOCint), was introduced as a safety threshold that users can tune based on their anxiety. If the path is not feasible (due to a violation of the SOCint parameter), the algorithm searches for an available charging spot, minimizing the deviation from the original path. In order to avoid the problem of looking for a charging opportunity farther and ending up in a longer trip, it looks for EVSE closer to the destination compared to the starting point. From each EVSE which can be reached without violating the SOCint threshold, it looks at all the paths to the destination, if feasible. If the algorithm finds one which does not exceed the SOCint parameter, it looks for a feasible path from the EVSE to the destination. Among all the feasible paths, it takes the one that either minimizes the consumption or the travel time, according to the user preference. If it cannot find a path with the previous step, then it again execute the algorithm to find an additional EVSE in which we can charge starting from the previous EVSE, which becomes the starting point for the next step. The algorithm repeats the previous steps until either it finds a feasible path and returns it to the user, or it cannot find any, and thus it returns to the user that it is not possible to travel through the desired path with the chosen parameters. A large-scale scenario has been simulated, i.e., the Italian Emilia-Romagna region, taking into account real EVSE positions and 3D street maps. Random trips were generated within the scenario, and the success probability was studied, i.e., the probability of reaching the destination when following the indications of the route planning service or when using a conventional approach commonly adopted by EV drivers. This approach consists in following the shortest path, and in seeking for an available EVSE, only when the charge is below a given threshold. Figure 9.18 shows that the SP decreases (dashed lines) without route planning but also decreases, in a much slower way, with route planning. This happens because the coverage of EVSEs on the target scenario is not uniform: most of the EVSEs are located on urban areas, while charging opportunities are quite scarce in rural areas. However, as before, fig.4.16 provides useful feedback for service planning, since it demonstrates the performance gain of a planning service for medium and long trips, and for grid planning, since it allows detecting the areas of the Emilia-Romagna region which are mostly uncovered by the current charging infrastructure.

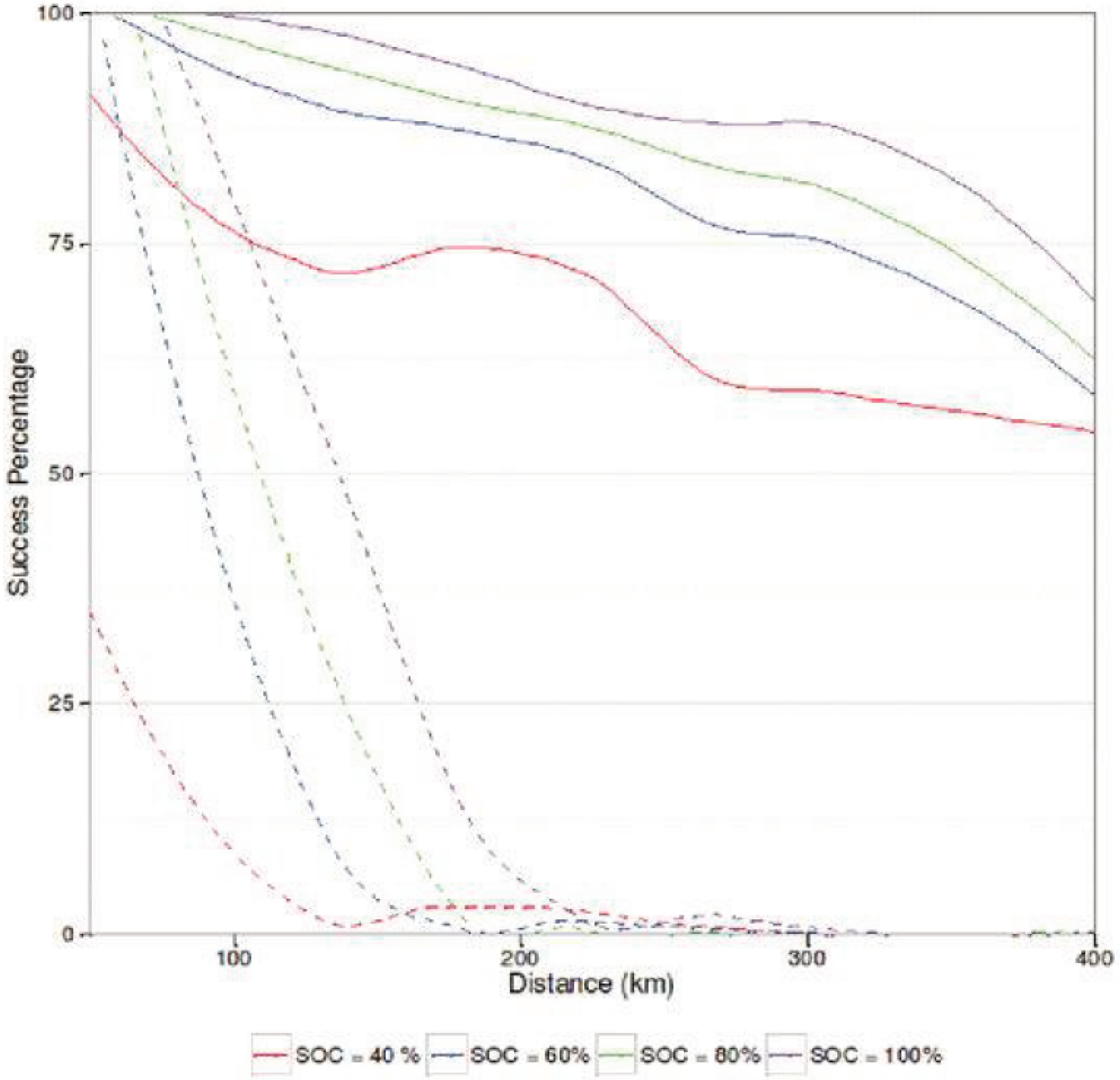


Figure 4.16: Success Probability for the Route Planning Service

Chapter 5

Conclusions and Some Open Research Challenges

The integration with fog and cloud has surely and remarkably leverages all the most common issues of IoT paradigm. The introduction of the fog middleware made feasible the introduction of DaaS concept by enabling all the advantages of fog paradigm, firstly at IoT level, and secondly toward cloud. Although a complete, comprehensive, and fully integrated framework for virtual sensors is still in early stage, this project has proved the complete feasibility and practicability of the idea and the concept. By exploiting the fog middleware the project has overcome:

Device Heterogeneity The fog middleware totally abstracts the physical layer of devices, overcoming all their hardware, technology, and communication protocol differences. The fog middleware makes the IoT objects and their functionalities available and accessible at cloud tier in a standardized and homogeneous manner.

Power Efficiency This project, through fog middleware, enhances the energy sustainability of IoT nodes. This aspect is crucial for devices running on battery. In addition, this research provides a strategy for power saving based on a BLE interface switching on/off mechanism dependent by the advertisers' arrivals.

Device Availability The fog middleware totally decouples the physical layer from the abstraction layer making the virtual sensor fault tolerant and enabling the data recovery. In addition, the research project also provides a Semantic Web-based data access methodology enabling federated query.

Object Data Rawness The adoption of the fog middleware allows multiple interactions IoT-Fog without the need of data/computation offloading to the cloud. In addition, the fog middleware acts as data aggregator and manipulator, providing at cloud level more stable and accurate data.

Moreover this dissertation has proved the feasibility of application of virtual sensor and fog nodes in telecommunication fields. This is only the first step towards the definition of a new paradigm and modus-operandi for telco providers. Starting from fifth generation network on, telco providers will be able to provide services running on virtual sensor and fog nodes through NFV methodology. This will lead at a huge cost reduction and a huge number of available nodes spread on the territory making the telco service pervasive and with much higher performance. In the last chapter is presented the complete application of virtual sensor architecture in a real use case. Nowadays electro mobility is a rapidly spreading scenario with a large number of still open issues. The project has presented the application of a fully vertical cloud-fog-IoT architecture similar to that one presented in this research work for DaaS. From the use case presented, emerges how this type of solutions are feasible and applicable to the real world and usable in common everyday life.

In the following section will be summarized the achieved results, then, finally, in the last section, the future research direction will be introduced.

5.1 Achieved results

This section will address and discuss the achieved results by this research. Firstly, the aspect emerged from the proposed adaptive energy saving strategy will be addressed, then, the focus will be moved on the results achieved by the proposed and adopted application discovery protocol. Finally the discussion will be moved on the abstract sensor data query method, analyzing the federated semantic web service, and, to conclude, the discussion will address the Arrowhead project use case results.

5.1.1 Architecture and Strategy

The fog paradigm enhances the IoT capabilities acting as a middleware between IoT nodes and Cloud. Fog supports IoT environments by providing location awareness, computing resources, mobility support, geo-distribution, and so on. These features are crucial for improving the quality of experience and the efficiency of service providing and seeking in IoT environments. The architecture presented in this work exploits the fog paradigm to improve the BLE node discovery in terms of device discoverability and in terms of power consumption. The proposed model, leveraging the location awareness of the nodes in the proximity, effectively triggers the discovery process thus granting PEND/SPEND full discoverability by overcoming typical issues of the conventional BLE discovery. At the same time, the employed ski-rental optimization allows to save energy by self-adapting the discovery process. From the implementation of the proposed adaptive strategy for the energy saving, has emerged interesting consideration. The maximum number of switching of the BLE interface has limited from the asymptote given by the energy consumed by WiFi connection. This behaviour is reasonable because with the enlarging of the arrival period, the BLE-A arrives ever more sporadically. With a that very occasional arrival, the BLE-S, basically, does not ever switch on its BLE interface, and the energy consumed by WiFi interface becomes the predominant component of the total energy consumption. Indeed, theoretically speaking, If the arrival period is pushed to the infinite, the energy consumption of the BLE interface becomes 0. This happens because the BLE-S never switches the BLE interface on, hence, the consumption of BLE interface is nullify, and the only determining factor of the power consumption of the whole system is the constant value of WiFi interface. Another remarkable aspect has emerged from the implementation of the strategy. By plotting the two separated components of energy, it is clearly observable the intersection between their trends. Those points of intersection projected on the X axis are the threshold arrival period after which is more convenient keep the BLE interface always active and scanning. Therefore, with the arrival period of threshold is possible to calculate the arrival rate limit for the switching on/off strategy.

5.1.2 PEND and SPEND

The introduction of fog middleware MQTT-based coordination of BLE discovery can relevantly help in increasing power sustainability for several IoT application domains where MQTT brokers are required to be employed for data collection. MQTT is an efficient Publish/Subscribe protocol for bandwidth and resource-constrained devices in IoT applications. Device discovery in MQTT results in sustainability concerns for the battery-limited devices as BLE-S always keeps the Bluetooth interface active to scan BLE-A in the vicinity. As this results in tremendous battery drain and –due to synchronization issues between the waveforms of scanner frame and the advertiser frame– high device matching rates may not be achieved. To cope with this, this research project adopted the proposed Power Efficient Node Discovery (PEND) protocol which employs an MQTT Broker to continuously collect the geo-locations of the BLE-A nodes, it wakes-up the BLE-S depending on the location of an approaching node. As the scanning frame starts with the synchronization of the WAKEUP message and the advertisement message, 100% device matching rate is achieved. Moreover, since the Bluetooth scanner interface is only activated whenever a BLE-A node is in the vicinity, significant power savings can be achieved under short to moderate scanning frame durations. However, it has also highlighted how long scanner frame durations may lead to inefficiencies in terms of battery drain due to wake up/sleep power consumption. To overcome to this PEND weak point, it has developed and proposed a new protocol, called Sustainable and Power Efficient Node Discovery (SPEND), which turns the BLE-S scanner interface off as soon as a device has been discovered during a scanning frame. The results under SPEND have shown that the battery drain by the BLE interface can be as low as 60%-70% of a naive-locality based discovery scheme whereas the battery drain of the discovery application under the proposed scheme can be as low as 85%-92% of the node discovery benchmark. In addition to the device sustainability, the proposed and adopted protocol achieved the full IoT nodes discoverability reaching the 100% device match ratio. This means that, with the interaction with the fog middleware, the devices (BLE-S) surely discover an new approaching node, making it available for further interaction or needs. The full device discoverability plays a crucial role in IoT scenario, where, by definition, the nodes have an high mobility.

5.1.3 New PEND

The results emerged from the first run of PEND and SPEND highlighted a slightly overhead in the CPU trend due to the BLE interface switching on/off. This is the key reason that has led to the introduction of the new version of PEND. In this new version, PEND protocol, still have the activation of the discovery process triggered by fog layer but it keeps the BLE interface always on. As it was expected the device discoverability remained guaranteed, since the triggering of discovery process is still driven by fog node. On the other hand, on the power efficiency side, the new version of PEND has proved to be more sustainable, highlighting how the switching on/off mechanism of BLE interface has a remarkable impact on the power drain. The CPU overhead has been leverage with this new version of protocol and that has led to have much better performance even in the CPU power drain due to BLE activities.

5.1.4 PEND and SPEND: Advertiser Dynamic Arrival

In the conventional node discovery approaches for IoT, a scanner node remains in the beaconing mode and continuously scans the environment for an approaching advertiser. In the previous experiments the proposed fog middleware driven discovery protocol have been tested according a static arrival rate of new approaching advertising nodes. As it is already stated the two proposed protocols are fog middleware driven; this means that the discovery process is triggered by fog node via MQTT message, if and only if a new a new advertising device is approaching. This kind of approach makes the two smart protocols advertiser's arrival dependent. Hence, the further test step has been analyzing the impact of advertisers' dynamical arrival on the two proposed protocol.

In section 3.4.2, it is investigated, on the conventional/Benchmark, PEND, and SPEND schemes, the impact of the dynamicity of the advertiser nodes (BLE-A) on the device discovery success and the sustainability of the battery-powered IoT nodes. Through experiments and emulation, it has been confirmed, as it is expected, that the fog middleware driven discovery of IoT nodes can always ensure 100% device discovery regardless of the dynamicity/arrival patterns of the BLE-A. Furthermore, the smart and power efficient node discovery solution SPEND can ensure significant energy savings in the BLE utilization, CPU utilization and the scanning application as a result of self-initiated deactivation of the scanner frames at the BLE-S nodes. While in the previous test set presented the promising behavior of SPEND under static/constant arrival of BLE-A nodes; the experimental results provided in set have further supported the potential of SPEND to be adopted in IoT-Fog environments.

5.1.5 Federated Semantic Web Service

Discovery, aggregation, and manipulation of distributed, diverse sets of data have become key in supporting the availability and the access to the various type of heterogeneous data. Semantic approaches have been proven valid to infer relationships and dependencies from heterogeneous sets of information pieces, however the current de facto standard query language SPARQL falls short when performing truly federated data navigation, with no exact knowledge of data distribution. This research project has proposed and adopted a lightweight Federation Ontology and a Federation Web Service to map information sources across different locations, so as to address current SPARQL limitations in terms of a priori network knowledge. The adopted Federated Web Service represents a portable, non vendor specific solution that relies on the Federation Ontology to infer network endpoints upon with to perform queries. The Federated Web Service then aggregates results via different composition strategies. The applicability of the proposed Federation Web Service and Ontology has been proved by realizing a Federated Education Portal that integrates data sets from both academic institutions and municipalities: the Federation Web Service and Ontology has granted to dramatically ease the development of the portal itself. The system adopted in this project have been also valued and then adopted by some organizations as a reference for their internal projects.

5.1.6 Fog Middleware Use Case: Arrowhead Electro Mobility Scenario

The automation of complex systems is a challenging context which has to deal with many factors, including technology readiness, business opportunities, user appreciation, legal regulations, heterogeneity of legacy systems and privacy issues. From this point of view, the electro-mobility scenario represents a relevant case study where an automation software architecture based on the Arrowhead Framework has been designed and implemented. The identified solution has been conceived to meet the scenario requirements, identified through the analysis of some relevant electro-mobility use cases selected in the Arrowhead project. The resulting software architecture is based on a IoT-to-cloud approach that efficiently exploits the IoT enabling technologies and the computational resources offered by the cloud infrastructure, providing an evident improvement in terms of interoperability, information security, and extendability. Among the heterogeneous set of systems and application services that is possible to plug in the architecture, the EM management system and the booking system have been analyzed in detail, designed, and implemented. With respect to other services provided by the Arrowhead Framework, their importance for the electro-mobility scenario has been highlighted or envisioned with simple considerations: the selection included, but in principle is not limited to, Weather Forecast, FlexOffer and RoutePlanning services. One of the most important results inferred from the performed analysis is that the full potential of a service is available only when it is integrated in a whole automation infrastructure with the ability to interact with other services for the orchestration of new advanced functionalities. The evaluation of the architecture and the services has been performed through a co-simulation framework taking into account the traffic, the variable percentage of electric vehicles, the power grid elements, and the implemented set of services. The simulation environment, according to the mobile application “zoo philosophy,” allowed realistic tests with real services applied to a simulated power grid and a simulated set of electric vehicles and users. The obtained results confirmed the validity of the proposed automation infrastructure, showing a clear improvement of metrics like the load balance of the power grid and the percentage of long trips ended without total battery discharge. Taking into account all the considerations done and the presented results, it is possible to confirm that the application of the Arrowhead Framework to complex scenarios allows their complete automation by improving the interaction between the subsystems and the cloud and preserving, at the same time, the indispensable security and reliability requirements.

5.2 Future Works

To conclude this thesis, in this last section, some future research directions will be addressed. As it is already state, and despite the great result already achieved by this research, this project is only the first stage of a broader and ambitious research. The future directions presented will be divided in two big category:

IoT-Fog Side This category will contain all the extensions and future works targeting the IoT world, the interaction IoT-Fog and the low level networking and device provisioning.

Fog-Cloud Interaction In this category will fall all the future work extensions aiming to improve the abstract resource provisioning, the complex multi virtual sensor service composition, and other business-oriented side services.

This final section is structured as following, for each category will be addressed firstly the future work direction, extension and improvement of the solution already adopted, and then totally novel features, solution, service or application will be introduced.

5.2.1 IoT-Fog Side Future Directions

In this field very important results have been already achieved, this research project has already proposed two solid device discovery protocol solutions runnable on any IoT device. These two solution are protocol that work at application layer, in this way the heterogeneity of the IoT devices is overcome. Despite these protocols result to be a good, solid and stable solution, they still present some open challenges, and provide some starting points for further improvements.

Discovery Protocol

The protocol already proposed, targeted BLE as reference communication model and technology mainly because it is the currently most used. The fist possible development would be make the discovery protocol completely independent by the technology and discovery approach. The discovery protocols adopted in this research projects are focused on device-to-device interaction, and BLE technology. A valuable extension of this work would be to implement and to test the proposed protocol with other relevant device-to-device technology such as WiFi direct or LTE direct. Moreover, it would be interesting to develop an hybrid long-ranged mid-short-ranged discovery protocol able to switch communication technology as soon as the approaching device gets closer. For example an interesting application of the concept of hybrid discovery protocol would be Wifi direct BLE discovery protocol, that exploits the long range capability of WiFi direct for fetching the first information regarding the new approaching device, and then exploits the low energy consumption of BLE technology, on the based of the information already gathered in the first step, to complete the discovery and to invoke functionalities. Another point to address as a future development regarding the discovery process is a fully study on the power efficiency. This branch offers a quite large plethora of ideas for future development. For example, the already proposed study is only focused on BLE scanner, this study might be extended as well to the BLE advertiser. By scenario definition, it has been assumed that the BLE-S had a static location, while the BLE-A was free to move. A question might be risen, is this the most power efficient solution? In other words, which does process consume more energy, the scanning or the advertising? Those questions will be addressed by a full and comprehensive power consumption study. Another interesting future research direction would be the application of real mobility patter of several real scenario to the advertisers' arrival. In this way, it would be possible to have a real feedback on the performance of the two proposed protocols applied to everyday real scenario. In stack of that, a new approach for discovery protocol might be proposed, an adaptive fog middleware driven device discovery protocol. It is possible to develop a discovery protocol that though a machine learning algorithm running on the fog middleware, is able to adapt itself to the advertisers mobility pattern minimizing the power consumption and optimizing the device discoverability.

Advanced Efficient Handoff of IoT Nodes

It has already been stated that one of problem of the sensor network in the IoT worlds is the node mobility. IoT is a computer paradigm highly dynamic, its network topology is in continuous reshaping, and its nodes might have an extremely high mobility. Cause of that, in IoT scenario, a lot of effort and energy is drained by node discovery process. This process would be drastically leveraged if the new approaching node would have been already known by the arrival network. This idea might be achieved by perform an IoT node handoff. This handoff might be feasible by exploiting the fog middleware. The first time a new IoT node approach to any IoT node network, it is discovered, it joins the network and all the information and services exposed by the new node are stored in a fog node. Then, when this node move to another IoT node network, all the information regarding that node are migrated by the fog middleware from the original fog node the fog node in the new locality, in this way the discovery process in the new network might be drastically reduced if not totally avoided. In IoT-Fog environment the IoT node handoff is possible by migrating all the IoT node information, services, context, and status, from a fog node to another. In literature, there already are some proposal regarding IoT node handoff, this future work direction will address the still open challenges in this area, such as IoT node status live migration, or proactivity IoT object migration based on location prediction algorithm. In both the above mentioned future development, the migration is meant live without any stop of service provisioning. This IoT node handoff is feasible by exploiting the proposed fog middleware, and its realization would be a very interesting and promising future research.

5.2.2 Fog-Cloud Side Future Directions

On this side of the architecture this research has already proposed a valuable method to access abstract sensors data and to query them. The main future research directions in this field would be

- Improving the methodologies already proposed

- The introduction of a Naming Service for virtual sensors
- the introduction of a billing and accounting service and a service compositor for business

The future work cloud side are mostly focused on improving the virtual sensor availability and accessibility and providing service for building complex services on top of virtual sensors. The first step for the future development is to investigate a strategy to mitigate the current limitation of Federal Semantic Web Service in terms of endpoint time-outs and unavailability. It is possible to think of developing a specific type of virtual sensor aimed to act only as endpoint of the Federated Web Service. These special virtual sensor would not be directly accessible, and available by end user application and other services, but their purpose would be only to be employed as endpoint for Federated Web Service infrastructure.

The other interesting aspect of cloud side future work would be the development of a Naming Service for virtual sensor. This service would have the same characteristics of the traditional naming service already presented in literature, but applied to virtual sensors. Naturally, this service will not contain the physical location of the abstracted sensor, but it would act more as a catalogue in which are listed all the available virtual sensor with their characteristics, available information and the list the exposed service to invoke. This virtual sensor catalogue will offer also the opportunities of composing virtual sensors for creating a more complex and personalized one. This will open the door to the adoption of virtual sensor by the companies. In addition, it is possible to develop a billing and accounting service. This will lead to the profiling of each user or entity that uses the virtual sensors, enabling a pay-per-use interaction model. The name service and the billing and accounting service will lead to the massive adoption of virtual sensor by the companies that will be able to build business on top of virtual sensors. By the exploiting of all these side services would be possible to develop, as a future work, a fully integrated framework that would aggregate all these services and would make easy to access, query, gather information, compose, list, and account virtual sensor and services.

Appendix A

Semantic Web: Evaluated Approaches and Platforms

This appendix gives an three main architectural alternatives that have been evaluated to realize a fully scalable and extensible SPARQL endpoint federation. Several qualitative Key Performance Indicators (KPIs), have been also defined to help the comparison, the organization, benefits and shortcomings of these alternatives, and to ultimately facilitate the choice of the further adopted solution. Any listed solutions has been checked against those KPIs to compare their properties and ease the choice. Furthermore, the same approach has been used in the middleware platform selection process. Last section of this appendix gives an overview of all the evaluated middleware platforms and all the KPI used to compare such platforms.

A.1 KPIs

Ease of development SPARQL federation should require limited development efforts, no matter the specific Semantic framework implementation.

Integration SPARQL federation solution should easily integrate with other framework features, with little or no additional effort.

Maintenance SPARQL federation solution should be conceived so as to limit development effort in case of bug fixing or feature adaptation activities

Evolution The solution should be easily extensible in order to cope with new requirements.

A.2 Architectural Approaches

Endpoint Extension This solution extends the internal logic of a SPARQL endpoint, from the classic logic to the execution of the query for extracting data to a model for the interception of the query, the query rewriting through the wired application logic, by inserting the statement required to query all the nodes in the federation, and finally performing on the various endpoints, returning the result in accordance with the provisions of the specific SERVICE clause SPARQL 1.1. This solution allows to provide input to any type of SPARQL query, with the only constraint of not being able to use the SERVICE clause, both because this is used as the main construct for the manipulation, either because it would make the handling and the high complexity of final query. This aspect is important as it does not allow the use of all the constructs that conform to the standard SPARQL 1.1, giving the value added to the product that more to the solution. Another aspect to consider is the difficulty in handling the query itself, which could lead to not-trivial query implementation and poor performance. In fact approach is extremely platformdependent from both a technical point of view (need to modify existing SPARQL endpoint source code), and a management one (modifying source code from other vendors and distributing it may pose legal and organizational challenges in terms of distribution process and code ownership), therefore achieving a low score on each KPI.

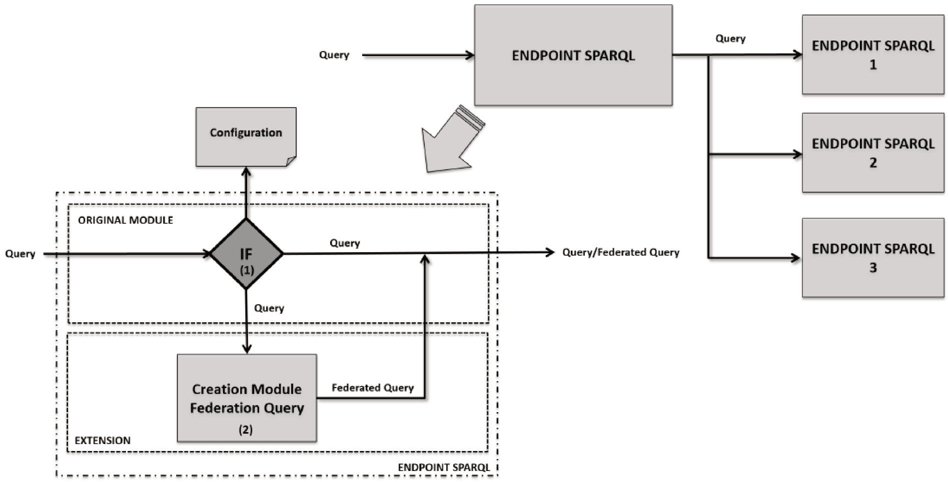


Figure A.1: Endpoint Extension Alternative

Plugin Some Semantic platforms and SPARQL implementations typically allow developers to extend platform features via plugin modules. Data federation may be realized as a dedicated plugin that transparently handles data federation across nodes, and overcomes current SPARQL limitations. This approach poses non-trivial technology issues:

- plugin implementation strictly depends on the actual semantic platform, therefore allowing to federate only nodes that rely on the same semantic platform
- only a subset of currently available, production-grade semantic platforms support a plugin model.

These limitations adversely impact all KPIs, thus making this option viable only for controlled environments where the semantic platform is shared across the federation and supports a plugin model.

Federation Web Services The Federation Web Service solution is based on creating a Web Service, which realizes all the functions related to federation. The Web Service approach makes this solution portable to any semantic platform implementation,

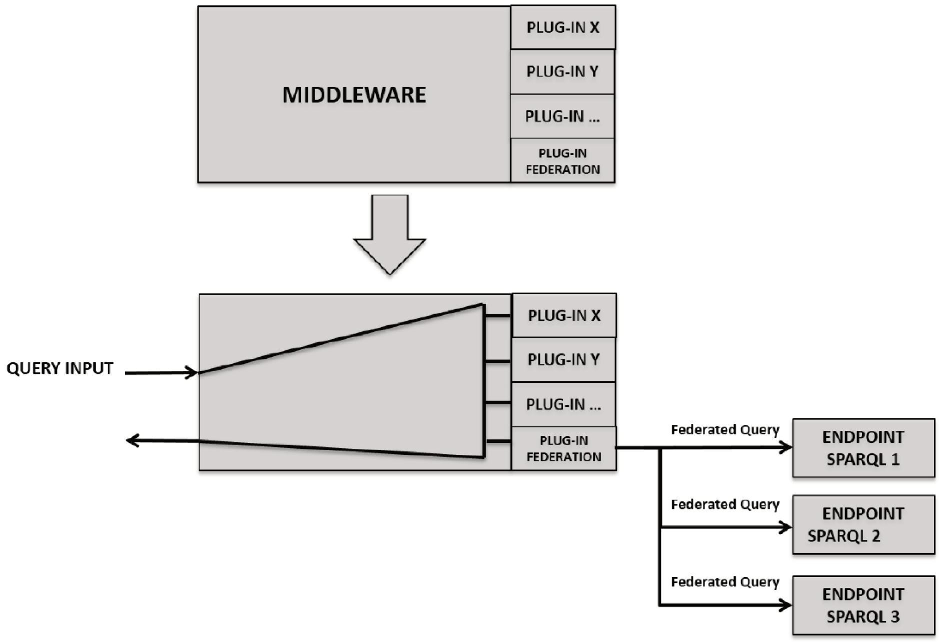


Figure A.2: Plugin Alternative

thus fostering openness and interoperability. Furthermore, to facilitate the management of large semantic data networks, we developed a specific Network Federation Ontology that facilitates the definition and navigation of network topology. The Federation Web Service relies on such a ontology to transparently determine which nodes and endpoints should be involved, thus facilitating the definition of federated semantic queries.

The logical flow of the Federation Web Service is as follows:

1. Federation resolution: thanks to the Federation Ontology, the Federation Web Service determines the actual endpoints involved in the federation;
2. Query execution: the Federation Web Service performs queries on all individual nodes involved in the federated query;
3. Result aggregation: the Federation Web Service aggregates the results obtained from single nodes and returns them to the caller. The aggregation techniques can be managed through the typical constructs of the SPARQL language, such as the UNION clause.

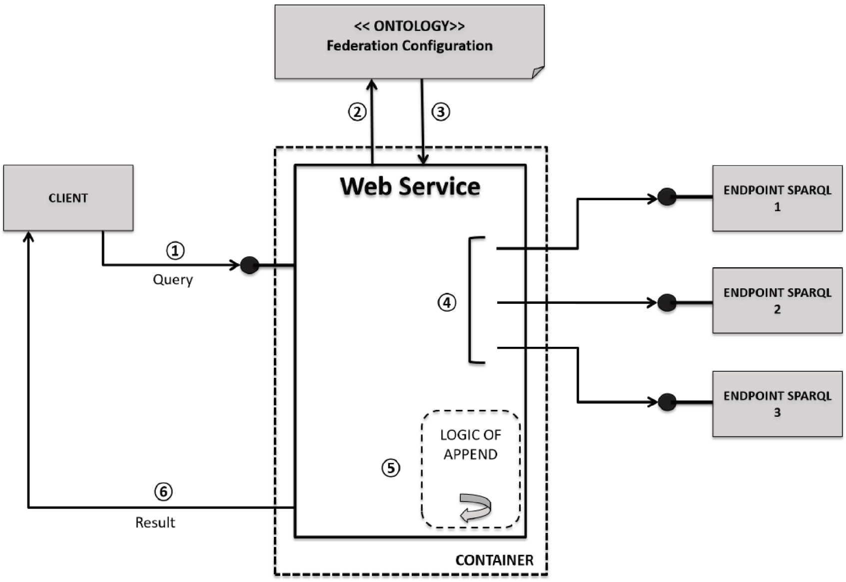


Figure A.3: Federation Web Service Alternative

In the following table A.1 are summarized the KPIs for each type of approach.

Table A.1: Solution Comparison according the KPIs

	Endpoint Extension	**Plugin Development**	**Federation WB**
Ease of Development	LOW - Platform dependent	LOW - Platform dependent	HIGH
Integration	LOW - requires distributing a new version of the endpoint	LOW - Platform dependent	HIGH – no vendor lock-in
Maintenance	LOW - Platform dependent	LOW - Platform dependent	HIGH
Evolution	LOW - Platform dependent	LOW - Platform dependent	HIGH – no vendor lock-in

Openness and portability of the Federation Web Service model grant this solution high KPI values, from both a development (ease) and management (integration, maintenance, evolution) point of view. The architectural choice has relied on the definition and implementation of a Federation Web Service.

A.3 Semantic Middleware Platforms

The first step in realizing the SPARQL federation model relates to the selection of a Semantic middleware platform on top of which the adopted solution has been built. The main and widely-diffused Semantic middleware solutions, again on the base of a set of relevant qualitative KPIs.

Apache JENA . Apache JENA is one of the most widespread, open-source semantic middleware on the market today. It is supported by the Apache community and is written entirely in Java programming language. JENA does not natively support SPARQL and SPARQL endpoints, though specific external libraries exist that provide both features (namely, the ARQ package and the Joseki extension) [168, 169].

Sesame . Sesame is an open source, Java-based RDF storage that natively supports SPARQL and exposes REST services to facilitate integration with external systems and services [170].

AllegroGraph . Allegrograph is an open source, Java-based RDF storage that natively supports SPARQL and exposes REST services to facilitate integration with external systems and services. SPARQL is the default query language in Allegrograph, and a implementation of the TWINQL specification is also available; this specification provides relevant query optimization features [171, 172].

Openlink Virtuoso . Virtuoso is a multi-model data storage solution that combines traditional RDBMS storage features, with advanced Semantic storage, reasoning support (e.g., RDF), and an extensive set of integration features
(e.g., REST/WebServices integration). SPARQL support is native in Virtuoso [143].

The above semantic middleware above described have been evaluated according qualitative KPIs. KPIs provided in the following are qualitative evaluation criteria relevant for any middleware choice. Our scoring attribution is based on the Capability Maturity Model Integration framework (CMMI), that provides a common reference model to assess the maturity of a system with respect to a given evaluation aspect [173]. Each KPI is evaluated against a set of values between 0 (no feature/characteristic support) to 5 (full, enterprise-grade feature/characteristic support).

Federation Support This KPI relates to the maturity of federation features provided by the candidate framework, and is the most relevant KPI in the adopted evaluation model (i.e., highest weight).
Jena, Virtuoso, and Allegrograph provide no support for federation besides the limited SPARQL SERVICE directive, hence scoring an evaluation of 0 (no support) for this KPI. Sesame, instead, supports some sort of data sources federation, but data model sharing is still at an early stage, hence scoring an evaluation of 1.

Community Support This KPI relates to the maturity of the community involved in the development and maintenance of the candidate framework. Typical community support features relate to bug tracking and resolution processes, documentation, wiki, online support, etc, and are crucial for long-term manageability and stability of the platform. Allegrograph has virtually no active community and provides only a static documentation of the platform, hence scoring a lowest result. Jena and Sesame provide a fairly active community that involves bug tracking/resolution, and product roadmap definition. Virtuoso features the most active community and provides bug tracking and resolution, product roadmap definition and direct on-call (chat) support.

Commercial Support Besides community support, commercial support is generally required in enterprise-grade software solutions to guarantee long-term stability. Commercial support typically provides highly skilled, special-purpose consulting resources as well as the ability to provide specific developments/extension and integrations. Jena has no commercial support, Allegrograph and Sesame provide some level of consulting support, and Virtuoso provides the most comprehensive commercial ecosystem that offers also specific developments by need.

SPARQL Endpoint Support This KPI describes the availability of SPARQL endpoint APIs within the candidate middleware. Virtuoso provides full endpoint exposition and support, whereas other candidates have only limited functionality.

Java Support This KPI relates to product compatibility with the Java programming language. This KPI is particularly relevant due to the widespread adoption of Java (and its enterprise counterpart JEE) in developing large enterprise applications. Jena implementation is fully based on the Java programming language, and exposes fully workable Java APIs; on the contrary Virtuoso is entirely written in C# and has limited API support for Java.

Market Adoption This KPI describes the availability skilled professional resources on the candidate framework. This KPI is especially relevant to guarantee manageability and evolvability of the solution. Virtuoso and Jena are leaders in terms of market adoption, hence featuring a larger market workforce share.

Product Maturity This KPI defines the overall maturity of the product, specifically in terms release process: products with frequent and long-term planned release schedules and roadmaps tend to be more reliable from a business point of view.

The KPIs above described are grouped in table A.2, where for each eligible middleware are listed the given grades based on a weighted sum of KPIs. The weights for the qualitative KPIs reflect their relevance according the fog middleware scopes. Federation Support and Market Adoption are particularly crucial in ac hiving the pre-fixed goals. KPI weight values are in a range between 0 and 1.
Aggregate results are obtained as a weighted sum of all KPIs: despite its poor Java support and lack of native federation support, Openlink Virtuoso was the Semantic platform that best fitted the project's business needs and provides a stable and reliable option.

Table A.2: Semantic Middleware Comparison

	Weight	**Jena**	**Virtuoso**	**Sesame**	**Allegrograph**
Federation Support	0,3	0	0	0	1
Community	0,1	4	2	5	4
Commercial Support	0,1	0	2	5	3
SPARQL Endpoint Support	0,1	1	2	5	1
Java Support	0,1	4	2	1	3
Market Adoption	0,2	3	1	3	1
Product Maturity	0,1	3	2	4	2
	1	1,8	1,2	**2,6**	1,8

Appendix B

Open Baton Overview

In the very recent period there was an increasing interest in having Orchestration solutions compliant with the ETSI NFV MANO information model in order to reduce the fragmentation between different existing management platforms. The Orchestrator selected is Open Baton, supported by the Fraunhofer FOKUS Institute and the Technical University of Berlin, implementing a compliant ETSI NFV MANO Framework using the Information Model specified in the initial phase of the standardization; its architecture is shown in fig. B.1.

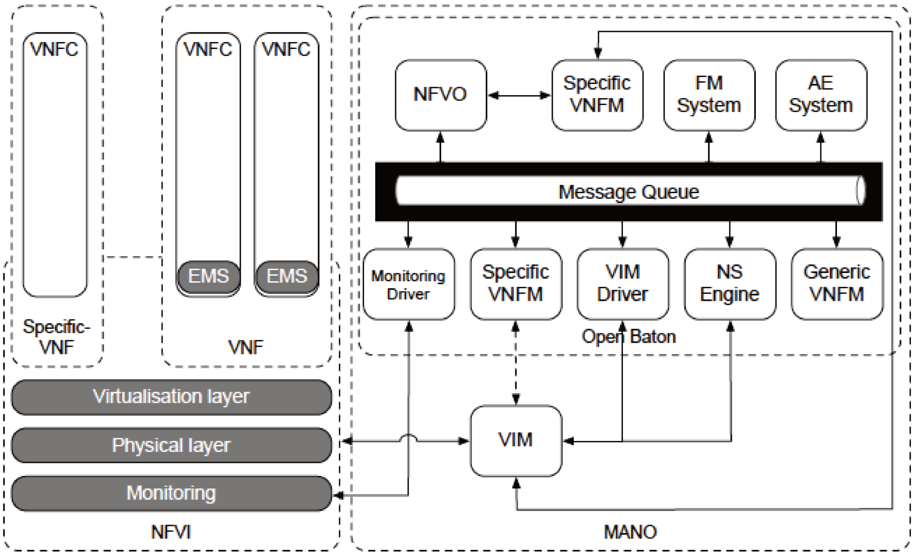
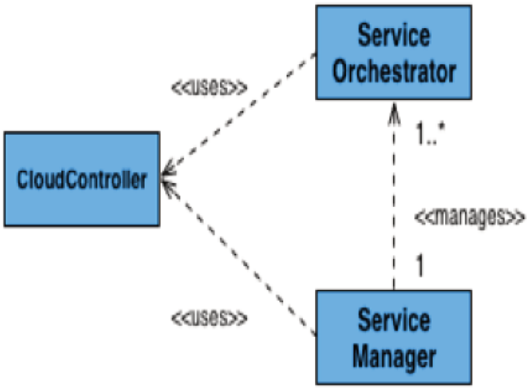


Figure B.1: Open Baton Framework

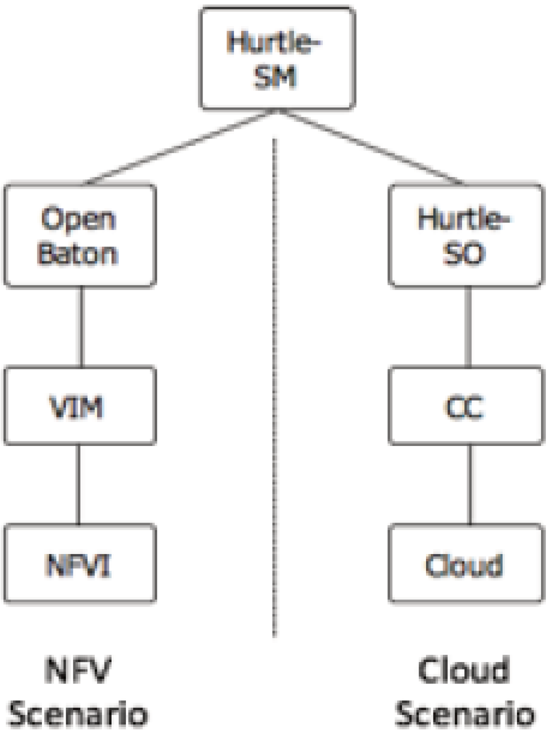
Open Baton follows the ETSI NFV Architecture, providing the NFV Orchestrator (NFVO) for managing and orchestrating combination of multiple VNFs, defined as Network Services. The NFVO makes use of multiple VNF Managers for the specific lifecycle of VNFs. Open Baton provides several capabilities for extending the set of use cases which can be implemented on top of it. It provides also a Generic VNFM for ease lifecycle management, and a Software Development Kit (SDK) for implementing new specific VNFMs. Furthermore, it provides a plugin mechanism which allows the simple integration of external components, like the Monitoring system and the Virtualized Infrastructure Manager (VIM). Finally, its powerful Event Engine allows the loosely coupled integration of third party components enhancing the set of functionalities the Framework provides. It communicates with one or more VNF Managers via a REST API or messaging system. The messaging system uses the AMQP protocol and its implementation is based on RabbitMQ. In the context of this work it was employed only the Generic-VNFM which provides generic capabilities for controlling the lifecycle events of a VNF based on its definition in the Network Service Descriptor (NSD). This VNFM was used for controlling all the IMS Service Instance Components (SICs). Some scripts have been implemented as part of the Element Management System (EMS) for integrating the IMS SICs into the Open Baton NFV Catalogue. Using the MCN terminology, IMS corresponds to a MCN Service. In order to provide the functionality “as a Service”, MCN Services are managed through a combination between Service Manager (SM) and Service Orchestrator (SO). Through the SM the Enterprise End User (EEU) can use the OCCI APIs to deploy, provision and dispose Service Instances (SIs) and retrieve their status-information. Furthermore, the MCN framework relies on the Cloud Controller entity as intermediate component between different Cloud Management systems. Fig. B.2a shows the key entities of the MCN architecture. The Hurtle framework, implemented by the InIT Cloud Computing Lab (ICCLab) at the Zurich University of Applied Sciences (ZHAW), represents a reference implementation of the MCN architecture. In order to manage the lifecycle of each Service Instance, the Hurtle approach proposes to instantiate SO on demand via the SM interface. This combination allows the SM to provide a gateway to numerous service instances for multiple consumers while the SO itself remains responsible for one specific deployment, only being accessed by the overlaying management. To let Open Baton be used as a SO by the SM, an API providing the same functionality with nearly identical signatures is being exposed. It offers deploy, provision, status information and disposing of service instances. Fig. B.2b shows how the integration has been realized.

While in the hurtle-SO approach one SO would manage only one service instance, in the Open Baton scenario, the NFVO manages multiple service instances, each identified by a universally unique identifier, which is required for every request except for the initialization. For this reason, it is not necessary to deploy on demand a new Service Orchestrator instance. Corresponding to this, the responsible SM is using a specific Service Orchestrator Manager implementation. It includes the mentioned unique service instance identifier in its requests so that it will do the requests to the correct service instances. While the reference manager implementation makes usage of PaaS like OpenShift to deploy and dispose service orchestrators into containers on demand, this is not needed for the Open Baton approach, container creation is totally skipped and the whole lifecycle management is passed over to Open Baton. The usage of a service manager using this manager implementation does only differ to the generic implementation in the need to specify host and port for the Open Baton in the service manager configuration. As can be seen in fig.B.3, the Enterprise End User requests the instantiation of the IMS SIC at the H-SM endpoint. Once the request is received by the SM, it calls internally the Open Baton Service Orchestrator Manager class which has been leveraged for supporting multiple types of Service Orchestrators. fig.B.3 shows the sequence diagram of the operations which are executed when a user requests the instantiation of a network service according to the following five main steps:

1. The Enterprise End User requests the instantiation of a IMS Service Instance via a



(a) MCN Main Entities



(b) Open Baton - Hurtle Integration

Figure B.2

POST request to the respective Service Manager endpoint (H-SM).

2. The H-SM requests the instantiation of the IMS SI to the Open Baton NFVO via the OCCI-Adapter. In particular, it first sends a PUT request for initializing it, and then sends a POST request for deploying the IMS SI.
3. Considering that the Open Baton NFVO is capable of orchestrating multiple types of Network Services, the OCCI-Adapter should request the instantiation of a particular Network Service via its unique identifier. This identifier is configured on the OCCI-Adapter at the installation time, but can be also passed in a OCCI-Attribute while doing the instantiation procedure.
4. Once the POST request has been received by the OCCI-Adapter, it requests the instantiation of the IMS SI NSD, which turns into a series of internal operations executed at the NFVO level. Those operations are usually associated with the execution of lifecycle events: instantiation of virtual resources required as described by the ITG, and execution of installation and configuration of each Service Instance Component with the support of the Generic VNFM (G-VNFM).
5. Once all the operations are concluded successfully, the IMS SI status is set to ACTIVE and the H-SM set the status to deployed.

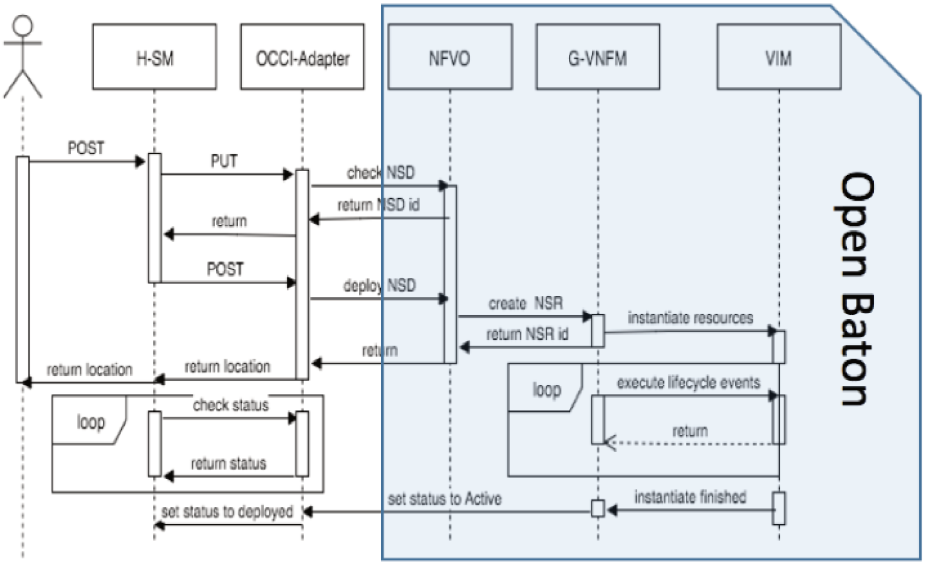


Figure B.3: Sequence Diagram of IMS SI deployment

Appendix C

The Electro Mobility Use Case: Context Overview and Introduction

A complex system represents a scenario where individual components cooperate and interact to create new functionalities, which cannot be provided by single components themselves. A complex system typically features a large number of interacting actors and components which have to efficiently interact for the sake of user satisfaction, for security reasons, or for specific industrial business purposes: heterogeneous hardware, different devices, software components, entire ICT subsystems, and a large number of user categories. Automating a complex system means permeating all the involved actors with the automation infrastructure components. The purpose is to achieve full interoperability at low costs and with simple interaction primitives. The Arrowhead Framework provides all the functionalities, services and tools to achieve a high grade of automation even with the challenges that characterize complex systems [174], among which are:

- Products from many industries that have to mutually understand and actively interact;
- Support for existing standards for data serialization and information models;
- Different computational capabilities of the involved devices: from supercomputers to autonomous energy harvesting embedded electronics;
- System requirements that range from maximum quality of service to maximum performances, depending on the analyzed subsystem;
- Cloud support with simplified interfaces for application development;
- System evolution and adaptability: new situations have to be managed adapting the system configuration to the new needs;
- System extension is to be taken into consideration: when new functionalities are added to an existing complex SoS, they should be discoverable by the other devices/systems enabling proper and efficient use;
- Privacy issues: when multiple HW/SW components from multiple vendors extensively interact in a common scenario, outstanding privacy issues arise for informa-

tion belonging to end users and to that belonging to the industries; reliable measures for preserving information ownership are needed to achieve complete automation and interoperability;

- Security issues on information must be reliably managed; stored information cannot be deleted or modified by unauthorized agents; all the interacting systems have to be properly authenticated when requesting or performing a service.

The complex system domains are a universe so rich and vast that cannot be treated exhaustively in this section. The study has been focused on a specific scenario that is relevant in terms of system complexity, market impact, and industrial interest. The objective is to show how, adopting the approach proposed in the Arrowhead project, it is possible to satisfy the requirements previously mentioned by relying on its framework for interaction among systems. The complex automation scenario that has been selected is the electro mobility (EM) domain, with a particular interest in the vehicle charging infrastructure and related ecosystem. The EM scenario is one of the main branches investigated by the Arrowhead project because of the industrial and academic relevance, the societal impact, business opportunities and eco-sustainability. EM is also sufficiently complex and diversified to claim that the resulting SoSs have been obtained by directly facing most of the issues existing for complex multi-vendor multi-player systems. Very briefly, the scenario considers an ecosystem where electric vehicles recharge their batteries from a network of charging stations connected to the power grid or powered through renewable sources (i.e., solar panels). The study and engineering of EM involve many industrial and academic players and have potential for significant benefits considering the impact on society, economy, transportation, and environment. Furthermore, EM represents a radical change from the current consolidated mobility panorama based on petrol or diesel vehicles, a change that is strongly influenced by the great investments required for the construction and deployment of distributed charging infrastructure and for the creation of the ecosystem of involved actors. EM, moreover, being a substantial innovation, could potentially be exposed to the difference of the end user, nowadays used to combustion engines reality, thus significantly reducing the EM potential acceptance. The concretization of future EM on a large scale and the creation of the corresponding market represent a serious challenge and the solution proposed in the Arrowhead project. can significantly reduce the effects of EM criticality, dramatically increasing the acceptance of EM transport solutions and contributing to their future diffusion. Arrowhead objectives in the EM domain are two-fold:

- Allow a simple, effective, reliable, and secure interaction among multiple heterogeneous multi-vendor components involved in the EM scenario.
- Improve the EM services quality, usability, and efficiency, in order to achieve a critical mass for incentivising governmental financing and attracting the interest and the investments of the important industrial players.

The following sub sections illustrate the Arrowhead Framework based solution for the EM domain, starting from the description of the main technologies involved and the automation plan identified, i.e., including the SoS architecture conceived to manage the scenario requirements. The SoS components are successively analyzed in detail. As a fully operational EM scenario is still not available on the large scale in European countries, at the current level of progress, a specific section describes a modular co-simulation platform that has been designed and integrated into the EM SoSs. The available information, merged with the data originated by the co-simulator is finally used to test some of the implemented application services and show the potential benefits they introduce in future EM scenarios. Lastly, conclusions are drawn to summaries the goals achieved by adopting the Arrowhead Framework and to describe the issues still pending in the automation process for EM and, more generally, for complex systems.

C.1 Vision from an Automation Perspective

Automation is considered one of the most promising enabling technologies in the EM market in order to minimize the environmental impacts of transportation, create new business opportunities, and increase efficiency, quality of service, and comfort for the final user. Automation is the one of the hottest research topics in the field of smart transportation and, more specifically, in the area of electro mobility. The evolution of electro-mobility is driven by the convergence and integration of different technologies, business models, and trends, including automation, green solutions, mobility as a service, intelligent transport systems, IoT and other key trends. The evolution of these factors will not continue isolated and the effects and synergies of their convergence have not been exploited yet. The Arrowhead Framework represents an enabling technology that will speedup this convergence process, providing tools for the simplification of the electro mobility infrastructure management, the optimization of resource usage, and a significant improvement of the perceived quality of service. Arrowhead will simplify also the removal of legal barriers and the adoption of new business models through the adoption of the “everything as a service” model, a widely accepted solution that exploits the vast potential for on-demand cloud-based services intended to fully displace the delivery of a commodity service. It is well known that in several cases the technology is already available but the application of automation to electro-mobility is prevented by unsolved legal and liability issues. Furthermore, current automation solutions have evolved independently for the vehicle, the recharge infrastructure and the ICT related solutions, thus not fully exploiting the potential benefits of electro mobility. From this point of view, the Arrowhead Framework represents an efficient service oriented integration framework, allowing the electro mobility Systems of Systems (SoSs) to spontaneously emerge from the cooperation of heterogeneous and complex systems. This vision, based on a service-oriented framework, intrinsically interoperable, allows the harmonized and orchestrated collaboration between the actors of the electro-mobility SoSs: automated and connected vehicles, recharge infrastructures, final users, freight, road and fleet operators, public transports, energy suppliers, ICT service providers, etc. The Arrowhead vision for electro-mobility will lead to a paradigm shift in the way future electric transport systems are envisaged, implemented, and deployed. The immediate impact of the adoption of the Arrowhead Framework in the electro-mobility domain is the possibility to set up an ICT solution for the integration and management of the distributed recharge infrastructure. Perhaps the most urgent need in the electro-mobility market is financing the charging infrastructure. An intense debate has been focused on the dilemma about the priorities in electro-mobility market development: should manufacturers focus on electric vehicles, or should they focus on the charging infrastructure first? Currently, the two components of the market are being developed simultaneously, with governments supporting the financing of the charging infrastructure: a study of the electric vehicle infrastructure estimates a global expense for the charging infrastructure of 2.5 billions dollars between 2008 and 2014, with 4.5 billion dollars of fiscal incentives for both vehicles and infrastructure development. In this context, the cloud integration platform based on the Arrowhead Framework could dramatically reduce the expenses for the development of the infrastructure and optimize the investments. The advantages could influence all the levels of the infrastructure development and future management:

- from the infrastructure design, development, and implementation;
- to its monitoring, maintenance, and optimization;
- to the creation of new electro-mobility services and related business opportunities.

C.1.1 Infrastructure Design, Development, and Implementation

During the design and development, phases the adoption of a unified, open and interoperable framework simplifies the design of the architecture of the charging infrastructure, both in terms of offered functionalities/services and in terms of ICT solutions, from the field up to the cloud. The electro-mobility solution based on the Arrowhead Framework (AF), at fog layer, adopts a pervasive IoT framework (Eclipse Kura [154]) that is responsible for the edge operations: it abstracts and isolates the developer from the complexity of the hardware and the networking subsystems, and redefines the development and reusability of integrated hardware and software solutions. Kura simplifies the design and development of the edge computing part of the charging infrastructure and contributes abstracting and hiding the technical details of the services exposed through the AF. On the cloud side, in turn, it simplifies the design and development of the core logic of the electro-mobility application, the use of existing services from third parties, and the creation of new business logic for specific electro-mobility applications and services. The presence of a pervasive IoT framework on the charging stations simplifies the deployment of the whole charging infrastructure, providing provisioning functionalities and full remote control during the installation process.

C.1.2 Maintenance and Optimization

The Arrowhead Framework introduces a significant reduction of maintenance costs: the remote control of charging stations based on cloud and IoT technologies enables continuous monitoring, remote, predictive and planned maintenance, reduction of technical interventions on the field, optimization of maintenance procedures, etc. The Arrowhead Framework also allows the optimization of the usage of the charging infrastructure. The capability to collect near real-time data from the charging infrastructure represents a huge opportunity in terms of analytics. The study of this huge amount of information allows the optimization of the use of the charging infrastructure in terms of electric load balancing, costs for recharges, number of users served, and optimization of the territorial coverage.

C.1.3 Monitoring

Another important source of cost reduction is related to the electricity distribution network that is required by the charging infrastructure. The availability of autonomous monitoring stations capable to recharge their internal batteries using alternative and green energy sources (i.e., solar panels) can significantly reduce the required territorial coverage of the electricity network, in particular in rural areas where only low power lines are available and the costs for improving grid infrastructure are higher. The charging stations based on the Arrowhead Framework are fully autonomous also from an ICT point of view, allowing them to seamlessly become part of the global charging infrastructure and services. The Arrowhead Framework IoT-based solution enables the creation of services that allow the connection of different electro-mobility players, providing a friendly access to charging stations. These services offer a solution for creating a provider-independent network of electric vehicle charging stations, a unified multivendor charging infrastructure (intercharge). A similar interchange platform could operate like roaming between different network operators on the international mobile communications market: the final user has one single, unified, simple way to access a variety of different charging stations spread over the territory. This is a significant example of the potential of the Arrowhead Framework in terms of new service creation and new business opportunities exploitation. There is, however, one obstacle to electric vehicle diffusion: the so-called range anxiety. Although all EVs have a driving range suitable for most users, this does not totally fulfill the common driver's expectation. Except for some high-end vehicles, EVs' range is lower than traditional vehicles and recharge time is significantly longer than gasoline/diesel refuel time. This is widely perceived as a lower usability of EVs in comparison to traditional vehicles. Arrowhead Framework electro-mobility infrastructure aims to overcome such obstacles by deploying a pervasive recharge network, characterized by the widespread availability of autonomous and remotely controlled charging stations. The pervasive coverage of the recharge infrastructure, especially in rural areas, is a crucial factor to compensate for the limitations of vehicle range, thus eliminating an important obstacle that will characterize electro-mobility for several years to come. The adoption of a service-oriented solution also introduces indirect positive impacts. An important obstacle for electric vehicle diffusion is the cost of the batteries that, currently, has a significant impact on the vehicle price, with respect to equivalent petrol or diesel models. The 30%–40% of the value added in purely electric vehicles is still due to the batteries. The availability of an electro-mobility infrastructure that will encourage the purchase of electric vehicle, is directly linked to their mass production and the cost reduction is a consequence of mass production.

Bibliography

[1] 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, no. 4, pp. 2347–2376, 2015.

[2] D. Evans, “The internet of things: How the next evolution of the internet is changing everything,” *CISCO white paper*, vol. 1, no. 2011, pp. 1–11, 2011.

[3] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787 – 2805, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128610001568

[4] L. D. Xu, W. He, and S. Li, “Internet of things in industries: A survey,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov 2014.

[5] R. Van Kranenburg, “The internet of things. a critique of ambient technology and the all-seeing network of rfid, network notebooks 02. institute of network cultures,” 2011.

[6] F. Alam, R. Mehmood, I. Katib, N. N. Albogami, and A. Albeshri, “Data fusion and iot for smart ubiquitous environments: A survey,” *IEEE Access*, vol. 5, pp. 9533–9554, 2017.

[7] X. Li, Q. Huang, and D. Wu, “Distributed large-scale co-simulation for iot-aided smart grid control,” *IEEE Access*, vol. 5, pp. 19 951–19 960, 2017.

[8] M. Pouryazdan and B. Kantarci, “The smart citizen factor in trustworthy smart city crowdsensing,” *IT Professional*, vol. 18, no. 4, pp. 26–33, Jul 2016.

[9] M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, and S. Andreescu, “Health monitoring and management using internet-of-things (iot) sensing with cloud-based processing: Opportunities and challenges,” in *IEEE Intl. Conf. on Services Computing (SCC)*, Jun 2015, pp. 285–292.

[10] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, “Iot middleware: A survey on issues and enabling technologies,” *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 1–20, 2017.

[11] C. Patsakis, R. Venanzi, P. Bellavista, A. Solanas, and M. Bouroche, “Personalized medical services using smart cities’ infrastructures,” in *Medical Measurements and Applications (MeMeA), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 1–5.

[12] E. Welbourne, L. Battle, G. Cole, K. Gould, K. Rector, S. Raymer, M. Balazinska, and G. Borriello, "Building the internet of things using rfid: the rfid ecosystem experience," *IEEE Internet computing*, vol. 13, no. 3, 2009.

[13] J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, and A. Marrs, *Disruptive technologies: Advances that will transform life, business, and the global economy*. McKinsey Global Institute San Francisco, CA, 2013, vol. 180.

[14] S. Muthuswamy and P. Ganapathi, "Internet of things – an overview," 02 2016.

[15] I. Gartner, "Gartner report. available at urlhttp," *cloudtimes. org/2013/12/20/gartner-theinternet-of-things-will-grow-30-times-to-26-billion-by-2020*, 2016.

[16] S. Krco, B. Pokric, and F. Carrez, "Designing iot architecture (s): A european perspective," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 79–84.

[17] Z. Yang, Y. Yue, Y. Yang, Y. Peng, X. Wang, and W. Liu, "Study and application on the architecture and key technologies for iot," in *Multimedia Technology (ICMT), 2011 International Conference on*. IEEE, 2011, pp. 747–751.

[18] M. A. Chaqfeh and N. Mohamed, "Challenges in middleware solutions for the internet of things," in *Collaboration Technologies and Systems (CTS), 2012 International Conference on*. IEEE, 2012, pp. 21–26.

[19] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges," in *Frontiers of Information Technology (FIT), 2012 10th International Conference on*. IEEE, 2012, pp. 257–260.

[20] L. Da Xu, "Enterprise systems: State-of-the-art and future trends." *IEEE Trans. Industrial Informatics*, vol. 7, no. 4, pp. 630–640, 2011.

[21] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad hoc networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[22] S. Wang, Z. Zhang, Z. Ye, X. Wang, X. Lin, and S. Chen, "Application of environmental internet of things on water quality management of urban scenic river," *International Journal of Sustainable Development & World Ecology*, vol. 20, no. 3, pp. 216–222, 2013.

[23] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, "Vision and challenges for realising the internet of things," *Cluster of European Research Projects on the Internet of Things, European Commision*, vol. 3, no. 3, pp. 34–36, 2010.

[24] J. Wan and J. D. Jones, "Managing it service management implementation complexity: from the perspective of the warfield version of systems science," *Enterprise Information Systems*, vol. 7, no. 4, pp. 490–522, 2013.

[25] R. Roman, P. Najera, and J. Lopez, “Securing the internet of things,” *Computer*, vol. 44, no. 9, pp. 51–58, 2011.

[26] L. Li, “Technology designed to combat fakes in the global supply chain,” *Business Horizons*, vol. 56, no. 2, pp. 167–177, 2013.

[27] T. H. Szymanski, “Security and privacy for a green internet of things,” *IT Professional*, vol. 19, no. 5, pp. 34–41, 2017.

[28] A. Dastjerdi, H. Gupta, R. Calheiros, S. Ghosh, and R. Buyya, “Chapter 4 - fog computing: principles, architectures, andapplications,” in *Internet of Things*, R. Buyya and A. V. Dastjerdi, Eds. Morgan Kaufmann, 2016, pp. 61 – 75. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128053959000046

[29] 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.

[30] M. Mukherjee, L. Shu, and D. Wang, “Survey of fog computing: Fundamental, network applications, and research challenges,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 1826–1857, thirdquarter 2018.

[31] M. Ketel, “Fog-cloud services for iot,” in *Proceedings of the SouthEast Conference*, ser. ACM SE ’17. New York, NY, USA: ACM, 2017, pp. 262–264. [Online]. Available: http://doi.acm.org/10.1145/3077286.3077314

[32] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[33] M. Aazam and E.-N. Huh, “Fog computing: The cloud-iot\/ioe middleware paradigm,” *IEEE Potentials*, vol. 35, no. 3, pp. 40–44, 2016.

[34] H. Atlam, R. Walters, and G. Wills, “Fog computing and the internet of things: a review,” *Big Data and Cognitive Computing*, vol. 2, no. 2, p. 10, 2018.

[35] W. Wang, Q. Wang, and K. Sohraby, “Multimedia sensing as a service (msaas): Exploring resource saving potentials of at cloud-edge iot and fogs,” *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 487–495, April 2017.

[36] OpenFog Consortium. (2017) OpenFog Consortium fog scale characteristics. [Online]. Available: https://www.openfogconsortium.org

[37] D. Willis, A. Dasgupta, and S. Banerjee, “Paradrop: a multi-tenant platform to dynamically install third party services on wireless gateways,” in *Proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture*. ACM, 2014, pp. 43–48.

[38] A. Majd, G. Sahebi, M. Daneshtalab, J. Plosila, and H. Tenhunen, "Hierarchal placement of smart mobile access points in wireless sensor networks using fog computing," in *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, March 2017, pp. 176–180.

[39] C. Chang, M. Liyanage, S. Soo, and S. N. Srirama, "Fog computing as a resource-aware enhancement for vicinal mobile mesh social networking," in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, March 2017, pp. 894–901.

[40] S. Sarkar and S. Misra, "From micro to nano: The evolution of wireless sensor-based health care," *IEEE Pulse*, vol. 7, no. 1, pp. 21–25, Jan 2016.

[41] F. Jalali, S. Khodadustan, C. Gray, K. Hinton, and F. Suits, "Greening iot with fog: A survey," in *2017 IEEE International Conference on Edge Computing (EDGE)*, June 2017, pp. 25–31.

[42] Cisco Systems. (2016) Fog computing and the internet of things: Extend the cloud to where the things are. [Online]. Available: http://www.cisco.com

[43] M. Satyanarayanan, V. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE pervasive Computing*, 2009.

[44] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 workshop on mobile big data*. ACM, 2015, pp. 37–42.

[45] Cisco Systems. (2014) Iox overview. [Online]. Available: http://goo.gl/n2mfiw

[46] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

[47] IBM. (2016) What is fog computing? [Online]. Available: https://www.ibm.com/blogs/cloud-computing/2014/08/fog-computing/

[48] OpenFog Consortium. (2017) Definition of fog computing. [Online]. Available: https://www.openfogconsortium.org/resources/definitionof-fog-computing

[49] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[50] R. Mahmud, R. Kotagiri, and R. Buyya, *Fog Computing: A Taxonomy, Survey and Future Directions*. Singapore: Springer Singapore, 2018, pp. 103–130. [Online]. Available: https://doi.org/10.1007/978-981-10-5861-5_5

[51] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of internet of things," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 46–59, 2018.

[52] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, “Fog computing: A platform for internet of things and analytics,” in *Big data and internet of things: A roadmap for smart environments*. Springer, 2014, pp. 169–186.

[53] I. Stojmenovic, “Fog computing: A cloud to the ground support for smart things and machine-to-machine networks,” in *Telecommunication Networks and Applications Conference (ATNAC), 2014 Australasian*. IEEE, 2014, pp. 117–122.

[54] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, “Applying nfv and sdn to lte mobile core gateways, the functions placement problem,” in *Proceedings of the 4th workshop on All things cellular: operations, applications, & challenges*. ACM, 2014, pp. 33–38.

[55] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, “Network function virtualization: Challenges and opportunities for innovations,” *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, 2015.

[56] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, “Fog computing may help to save energy in cloud computing,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1728–1739, 2016.

[57] M. A. Hassan, M. Xiao, Q. Wei, and S. Chen, “Help your mobile applications with fog computing,” in *Sensing, Communication, and Networking-Workshops (SECON Workshops), 2015 12th Annual IEEE International Conference on*. IEEE, 2015, pp. 1–6.

[58] L. Pu, X. Chen, J. Xu, and X. Fu, “D2d fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted d2d collaboration,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3887–3901, 2016.

[59] R. Mahmud, F. L. Koch, and R. Buyya, “Cloud-fog interoperability in iot-enabled healthcare solutions,” in *Proceedings of the 19th International Conference on Distributed Computing and Networking*. ACM, 2018, p. 32.

[60] Y. Wang, T. Uehara, and R. Sasaki, “Fog computing: Issues and challenges in security and forensics,” in *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, vol. 3. IEEE, 2015, pp. 53–59.

[61] S. Yi, Z. Hao, Z. Qin, and Q. Li, “Fog computing: Platform and applications,” in *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*. IEEE, 2015, pp. 73–78.

[62] P. Hu, S. Dhelim, H. Ning, and T. Qiu, “Survey on fog computing: architecture, key technologies, applications and open issues,” *Journal of Network and Computer Applications*, vol. 98, pp. 27–42, 2017.

[63] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, “Fog computing: Focusing on mobile users at the edge,” *arXiv preprint arXiv:1502.01815*, 2015.

[64] A. Bader, H. Ghazzai, A. Kadri, and M.-S. Alouini, “Front-end intelligence for large-scale application-oriented internet-of-things,” *IEEE Access*, vol. 4, pp. 3257–3272, 2016.

[65] F. Van den Abeele, J. Hoebeke, G. K. Teklemariam, I. Moerman, and P. Demeester, “Sensor function virtualization to support distributed intelligence in the internet of things,” *Wireless Personal Communications*, vol. 81, no. 4, pp. 1415–1436, 2015.

[66] 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, no. 5, pp. 508–521, 2014.

[67] K. W. Yang, M. Wang, K. J. Zou, M. Hua, J. J. Hu, J. Zhang, W. Sheng, and X. You, “Device discovery for multihop cellular networks with its application in lte,” *IEEE Wireless Communications*, vol. 21, no. 5, pp. 24–34, 2014.

[68] K. W. Choi and Z. Han, “Device-to-device discovery for proximity-based service in lte-advanced system,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 1, pp. 55–66, 2015.

[69] A. Prasad, K. Samdanis, A. Kunz, and J. Song, “Energy efficient device discovery for social cloud applications in 3gpp lte-advanced networks,” in *Computers and Communication (ISCC), 2014 IEEE Symposium on*. IEEE, 2014, pp. 1–6.

[70] T. Kushida and M. Yuriyama, “Sensor-cloud infrastructure - physical sensor management with virtualized sensors on cloud computing,” in *2010 13th International Conference on Network-Based Information Systems(NBIS)*, vol. 00, 09 2010, pp. 1–8. [Online]. Available: doi.ieeecomputersociety.org/10.1109/NBiS.2010.32

[71] A. P. Jayasumana, Q. Han, and T. H. Illangasekare, “Virtual sensor networks - a resource efficient approach for concurrent applications,” in *Fourth International Conference on Information Technology (ITNG’07)*, April 2007, pp. 111–115.

[72] S. Madria, V. Kumar, and R. Dalvi, “Sensor cloud: A cloud of virtual sensors,” *IEEE Software*, vol. 31, no. 2, pp. 70–77, Mar 2014.

[73] N. Raveendranathan, S. Galzarano, V. Loseu, R. Gravina, R. Giannantonio, M. Sgroi, R. Jafari, and G. Fortino, “From modeling to implementation of virtual sensors in body sensor networks,” *IEEE Sensors Journal*, vol. 12, no. 3, pp. 583–593, March 2012.

[74] J. Stephant, A. Charara, and D. Meizel, “Virtual sensor: application to vehicle sideslip angle and transversal forces,” *IEEE Transactions on Industrial Electronics*, vol. 51, no. 2, pp. 278–289, April 2004.

[75] C. Y. Liang, S. Srinivasan, and E. E. Jacobson, “Nox emission-control system using a virtual sensor,” Apr. 19 2005, uS Patent 6,882,929.

[76] H. Myeong and Y. Hong, “Method and/or apparatus for navigating mobile robot using virtual sensor,” Apr. 29 2014, uS Patent 8,712,588.

[77] M. Oosterom and R. Babuska, “Virtual sensor for fault detection and isolation in flight control systems - fuzzy modeling approach,” in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, vol. 3, Dec 2000, pp. 2645–2650 vol.3.

[78] P. Buschka and A. Saffiotti, “A virtual sensor for room detection,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, Sept 2002, pp. 637–642 vol.1.

[79] R. Rallo, J. Ferre-GinÃ©, A. Arenas, and F. Giralt, “Neural virtual sensor for the inferential prediction of product quality from process variables,” *Computers & Chemical Engineering*, vol. 26, no. 12, pp. 1735 – 1754, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098135402001485

[80] J. Park, M. Moon, S. Hwang, and K. Yeom, “Cass: A context-aware simulation system for smart home,” in *5th ACIS International Conference on Software Engineering Research, Management Applications (SERA 2007)*, Aug 2007, pp. 461–467.

[81] D. D. Finlay, “Smart environments: Technology to support healthcare,” *Technology and Health Care*, vol. 17, no. 3, pp. 159–160, 2009.

[82] L. Coyle, S. Neely, G. Stevenson, M. Sullivan, S. Dobson, and P. Nixon, “Sensor fusion-based middleware for smart homes,” *International Journal of Assistive Robotics and Mechatronics*, vol. 8, no. 2, pp. 53–60, 2007.

[83] S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta, “Role of middleware for internet of things: A study,” *International Journal of Computer Science and Engineering Survey*, vol. 2, no. 3, pp. 94–105, 2011.

[84] T. Gu, H. K. Pung, and D. Q. Zhang, “A service-oriented middleware for building context-aware services,” *Journal of Network and Computer Applications*, vol. 28, no. 1, pp. 1 – 18, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804504000451

[85] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain, “A survey on sensor-cloud: architecture, applications, and approaches,” *International Journal of Distributed Sensor Networks*, vol. 9, no. 2, p. 917923, 2013.

[86] C. X. Mavromoustakis, G. Mastorakis, J. M. Batalla, and P. Chatzimisios, “Social-oriented mobile cloud offload processing with delay constraints for efficient energy conservation,” in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–7.

[87] A. Liendo, D. Morche, R. Guizzetti, and F. Rousseau, “Efficient Bluetooth Low Energy Operation for Low Duty Cycle Applications,” in *IEEE International Conference on Communications (ICC’2018)*, Kansas City, MO, United States, May 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01775064

[88] G. Shan and B. Roh, “Advertisement interval to minimize discovery time of whole ble advertisers,” *IEEE Access*, vol. 6, pp. 17 817–17 825, 2018.

[89] J. Liu, C. Chen, and Y. Ma, “Modeling neighbor discovery in bluetooth low energy networks,” *IEEE Communications Letters*, vol. 16, no. 9, pp. 1439–1441, September 2012.

[90] C. Drula, C. Amza, F. Rousseau, and A. Duda, “Adaptive energy conserving algorithms for neighbor discovery in opportunistic bluetooth networks,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 1, pp. 96–107, Jan 2007.

[91] P. H. Kindt, M. Saur, M. Balszun, and S. Chakraborty, “Neighbor discovery latency in ble-like protocols,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 617–631, March 2018.

[92] T. Renzler, M. Spörk, C. A. Boano, and K. Römer, “Improving the efficiency and responsiveness of smart objects using adaptive ble device discovery,” in *Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects*, ser. SMARTOBJECTS ’18. New York, NY, USA: ACM, 2018, pp. 7:1–7:10. [Online]. Available: http://doi.acm.org/10.1145/3213299.3213306

[93] M. Alam, M. Albano, A. Radwan, and J. Rodriguez, “Candi: context-aware node discovery for short-range cooperation,” *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 5, pp. 861–875, 2015.

[94] A. V. Dastjerdi and R. Buyya, “Fog computing: Helping the internet of things realize its potential,” *Computer*, vol. 49, no. 8, pp. 112–116, Aug 2016.

[95] A. HernÃ¡ndez-Solana, D. Perez-Diaz-de Cerio, A. Valdovinos, and J. L. Valenzuela, “Proposal and evaluation of ble discovery process based on new features of bluetooth 5.0,” *Sensors*, vol. 17, no. 9, 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/9/1988

[96] P. Serrano, A. Garcia-Saavedra, G. Bianchi, A. Banchs, and A. Azcorra, “Per-frame energy consumption in 802.11 devices and its implication on modeling and design,” *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1243–1256, Aug. 2015. [Online]. Available: https://doi.org/10.1109/TNET.2014.2322262

[97] G. Foddis, R. G. Garroppo, S. Giordano, G. Procissi, S. Roma, and S. Topazzi, “Lte traffic analysis for signalling load and energy consumption trade-off in mobile networks,” in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 6005–6010.

[98] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, “A close examination of performance and power characteristics of 4g lte networks,” in *Proc. of the 10th Intl. Conf. on Mobile Systems, Applications, and Services*, ser. MobiSys ’12, 2012, pp. 225–238.

[99] M. S. Manasse, *Ski Rental Problem*. Boston, MA: Springer US, 2008, pp. 849–851. [Online]. Available: https://doi.org/10.1007/978-0-387-30162-4_378

[100] A. R. Karlin, M. S. Manasse, L. A. McGeoch, and S. Owicki, “Competitive randomized algorithms for nonuniform problems,” *Algorithmica*, vol. 11, no. 6, pp. 542–571, Jun 1994. [Online]. Available: https://doi.org/10.1007/BF01189993

[101] A. R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator, “Competitive snoopy caching,” *Algorithmica*, vol. 3, no. 1, pp. 79–119, Nov 1988. [Online]. Available: https://doi.org/10.1007/BF01762111

[102] G. Cardone, A. Corradi, and L. Foschini, “Cross-network opportunistic collection of urgent data in wireless sensor networks,” *The Computer Journal*, vol. 54, no. 12, pp. 1949–1962, 2011. [Online]. Available: http://dx.doi.org/10.1093/comjnl/bxr043

[103] A. R. Biswas and R. Giaffreda, “Iot and cloud convergence: Opportunities and challenges,” in *2014 IEEE World Forum on Internet of Things (WF-IoT)(WF-IOT)*, vol. 00, March 2014, pp. 375–376. [Online]. Available: doi.ieeecomputersociety.org/10.1109/WF-IoT.2014.6803194

[104] F. Li, M. Voegler, M. Claessens, and S. Dustdar, “Efficient and scalable iot service delivery on cloud,” in *2013 IEEE Sixth International Conference on Cloud Computing*, June 2013, pp. 740–747.

[105] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, “Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services,” *IEEE transactions on Services Computing*, no. 3, pp. 223–235, 2010.

[106] M. M. Hassan, B. Song, and E.-N. Huh, “A framework of sensor-cloud integration opportunities and challenges,” in *Proceedings of the 3rd international conference on Ubiquitous information management and communication*. ACM, 2009, pp. 618–626.

[107] M. Yuriyama and T. Kushida, “Sensor-cloud infrastructure-physical sensor management with virtualized sensors on cloud computing,” in *2010 13th International Conference on Network-Based Information Systems*. IEEE, 2010, pp. 1–8.

[108] S. Alam and J. Noll, “A semantic enhanced service proxy framework for internet of things,” in *Proceedings of the 2010 IEEE/ACM Int’l Conference on Green Computing and Communications & Int’l Conference on Cyber, Physical and Social Computing*. IEEE Computer Society, 2010, pp. 488–495.

[109] S. Alam, M. M. Chowdhury, and J. Noll, “Senaas: An event-driven sensor virtualization approach for internet of things cloud,” in *Networked Embedded Systems for Enterprise Applications (NESEA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1–6.

[110] X. M. Zhang and N. Zhang, “An open, secure and flexible platform based on internet of things and cloud computing for ambient aiding living and telemedicine,” in *Computer and Management (CAMAN), 2011 International Conference on*. IEEE, 2011, pp. 1–4.

[111] D. Gachet, M. de Buenaga, F. Aparicio, and V. Padrón, “Integrating internet of things and cloud computing for health services provisioning: The virtual cloud carer project,” in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*. IEEE, 2012, pp. 918–921.

[112] Y. Bo and H. Wang, “The application of cloud computing and the internet of things in agriculture and forestry,” in *Service Sciences (IJCSS), 2011 International Joint Conference on*. IEEE, 2011, pp. 168–172.

[113] J. Soldatos, M. Serrano, and M. Hauswirth, “Convergence of utility computing with the internet-of-things,” in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*. IEEE, 2012, pp. 874–879.

[114] IETF. (2012) Constrained application protocol (coap). [Online]. Available: http://tools.ietf.org/html/draft-ietf-core-coap-08

[115] G. Antunes, J. Borbinha, and A. Caetano, “An application of semantic techniques to the analysis of enterprise architecture models,” in *System Sciences (HICSS), 2016 49th Hawaii International Conference on*. IEEE, 2016, pp. 4536–4545.

[116] N. A. Rakhmawati and M. Hausenblas, “On the impact of data distribution in federated sparql queries,” in *2012 IEEE Sixth International Conference on Semantic Computing*. IEEE, 2012, pp. 255–260.

[117] A. Zimmermann, M. Pretz, G. Zimmermann, D. G. Firesmith, I. Petrov, and E. El-Sheikh, “Towards service-oriented enterprise architectures for big data applications in the cloud,” in *2013 17th IEEE International Enterprise Distributed Object Computing Conference Workshops*. IEEE, 2013, pp. 130–135.

[118] OGC - Open Geospatial Consortium. (2018) sensor web enablement (swe). [Online]. Available: http://www.opengeospatial.org/standards

[119] J. Sung, Y. Kim, T. Kim, Y.-J. Kim, and D. Kim, “Internet metadata framework for plug and play wireless sensor networks,” in *Sensors Applications Symposium, 2009. SAS 2009. IEEE*. IEEE, 2009, pp. 320–324.

[120] S. Tilak, K. Chiu, N. B. Abu-Ghazaleh, and T. Fountain, “Dynamic resource discovery for sensor networks,” in *International Conference on Embedded and Ubiquitous Computing*. Springer, 2005, pp. 785–796.

[121] A. Sheth, C. Henson, and S. S. Sahoo, “Semantic sensor web,” *IEEE Internet computing*, vol. 12, no. 4, 2008.

[122] D. J. Russomanno, C. R. Kothari, and O. A. Thomas, “Building a sensor ontology: A practical approach leveraging iso and ogc models.” in *IC-AI*, 2005, pp. 637–643.

[123] M. Compton, C. A. Henson, H. Neuhaus, L. Lefort, and A. P. Sheth, “A survey of the semantic specification of sensors,” in *SSN*, 2009.

[124] J.-H. Kim, H. Kwon, D.-H. Kim, H.-Y. Kwak, and S.-J. Lee, “Building a service-oriented ontology for wireless sensor networks,” in *Computer and Information Science, 2008. ICIS 08. Seventh IEEE/ACIS International Conference on*. IEEE, 2008, pp. 649–654.

[125] V. Huang and M. K. Javed, “Semantic sensor information description and processing,” in *Sensor Technologies and Applications, 2008. SENSORCOMM’08. Second International Conference on*. IEEE, 2008, pp. 456–461.

[126] H. Patni, C. Henson, and A. Sheth, “Linked sensor data,” in *Collaborative Technologies and Systems (CTS), 2010 International Symposium on*. IEEE, 2010, pp. 362–370.

[127] J. Pschorr, C. Henson, H. Patni, and A. Sheth, “Sensor discovery on linked data, kno. e,” sis Center Technical Report, Tech. Rep., 2010.

[128] S. Madria, V. Kumar, and R. Dalvi, “Sensor cloud: A cloud of virtual sensors,” *IEEE software*, vol. 31, no. 2, pp. 70–77, 2014.

[129] 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, no. 4, pp. 2347–2376, Fourthquarter 2015.

[130] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, “A survey on application layer protocols for the internet of things,” *Transaction on IoT and Cloud Computing*, vol. 3, no. 1, pp. 11–17, 2015.

[131] J. Y. Huang, P. H. Tsai, and I. E. Liao, “Implementing publish/subscribe pattern for coap in fog computing environment,” in *8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Oct 2017, pp. 175–180.

[132] M. H. Asghar and N. Mohammadzadeh, “Design and simulation of energy efficiency in node based on mqtt protocol in internet of things,” in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, Oct 2015, pp. 1413–1417.

[133] A. Bhattacharyya and S. Bandyopadhyay, “Lightweight internet protocols for web enablement of sensors using constrained gateway devices,” in *Proc. of the Intl Conf. on Computing, Networking and Communications (ICNC)*, ser. ICNC ’13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 334–340.

[134] D. H. Mun, M. L. Dinh, and Y. W. Kwon, “An assessment of internet of things protocols for resource-constrained applications,” in *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1, June 2016, pp. 555–560.

[135] R. M. Karp, “On-line algorithms versus off-line algorithms: How much is it worth to know the future?” in *IFIP Congress (1)*, vol. 12, 1992, pp. 416–429.

[136] G. Peralta, M. Iglesias-Urkia, M. Barcelo, R. Gomez, A. Moran, and J. Bilbao, “Fog computing based efficient iot scheme for the industry 4.0,” in *2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*, May 2017, pp. 1–6.

[137] Mosquitto Server, “,” https://mosquitto.org.

[138] Android Debug Bridge, “,” https://developer.android.com/studio/command-line/adb.html.

[139] R. Venanzi, B. Kantarci, L. Foschini, and P. Bellavista, “Mqtt-driven node discovery for integrated iot-fog settings revisited: The impact of advertiser dynamicity,” in *2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, March 2018, pp. 31–39.

[140] W3C. (2015) W3c recommendation: “owl web ontology languange”. [Online]. Available: https://www.w3.org/2001/sw/

[141] ——. (2004) W3c recommendation: "owl web ontology languange". [Online]. Available: http://www.w3.org/TR/owl-ref/

[142] ——. (2013) W3c recommendation: "sparql 1.1 federated query". [Online]. Available: http://www.w3.org/TR/2013/REC-sparql11-federated-query-20130321/

[143] OpenLink Software 2015. (2018) Openlink virtuoso. [Online]. Available: http://virtuoso.openlinksw.com/

[144] MCN. (2016) The eu fp7 mobile cloud networking (mcn) project. [Online]. Available: http://www.mobile-cloud-networking.eu/site/

[145] ETSI. (2016) Etsi network function virtualization (nfv). [Online]. Available: http://www.etsi.org

[146] P. Bellavista, “Mobile cloud networking: Lessons learnt, open research directions, and industrial innovation opportunities,” in *2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*. IEEE, 2016, pp. 79–80.

[147] 3GPP. (2016) Ts 23.228. ip multimedia subsystem (ims). [Online]. Available: http://www.3gpp.org/DynaReport/23228.htm

[148] G. Carella, L. Foschini, A. Pernafini, P. Bellavista, A. Corradi, M. Corici, F. Schreiner, and T. Magedanz, “Quality audit and resource brokering for network functions virtualization (nfv) orchestration in hybrid clouds,” in *Global Communications Conference (GLOBECOM), 2015 IEEE*. IEEE, 2015, pp. 1–6.

[149] J. Delsing, *Iot automation: Arrowhead framework*. CRC Press, 2017.

[150] M. Ornato, T. S. Cinotti, A. Borghetti, P. Azzoni, A. D’Elia, F. Viola, F. Montori, and R. Venanzi, “Application system design: Complex systems management and automation,” in *IoT Automation*. CRC Press, 2017, pp. 317–352.

[151] ARROWHEAD. (2013) The arrowhead framework. [Online]. Available: http://www.arrowhead.eu

[152] Arrowhead Framework. (2016) Pilot domain: Eletro mobility. [Online]. Available: http://www.arrowhead.eu/about/pilot-domain-electro-mobility/

[153] S. Mehar, S. M. Senouci, and G. Rémy, "Ev-planning: Electric vehicle itinerary planning," in *International conference on smart communications in network technologies*, vol. 1, 2013, pp. 17–19.

[154] Eurotech. (2016) Ecliple kura: The extensible open source java/osgi iot edge framework. [Online]. Available: https://www.eclipse.org/kura/

[155] L. L. Ferreira, L. Siksnys, P. Pedersen, P. Stluka, C. Chrysoulas, T. Le Guilly, M. Albano, A. Skou, C. Teixeira, and T. Pedersen, "Arrowhead compliant virtual market of energy," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE, 2014, pp. 1–8.

[156] K. Clement-Nyns, E. Haesen, and J. Driesen, "The impact of charging plug-in hybrid electric vehicles on a residential distribution grid," *IEEE Transactions on power systems*, vol. 25, no. 1, pp. 371–380, 2010.

[157] L. P. Fernandez, T. G. San Román, R. Cossent, C. M. Domingo, and P. Frias, "Assessment of the impact of plug-in electric vehicles on distribution networks," *network*, vol. 16, p. 21, 2011.

[158] D. G. Korzun, A. M. Kashevnik, S. I. Balandin, and A. V. Smirnov, "The smart-m3 platform: Experience of smart space application development for internet of things," in *Conference on Smart Spaces*. Springer, 2015, pp. 56–67.

[159] J. Honkola, H. Laine, R. Brown, and O. Tyrkkö, "Smart-m3 information sharing platform," in *Computers and Communications (ISCC), 2010 IEEE Symposium on*. IEEE, 2010, pp. 1041–1046.

[160] E. Ovaska, T. S. Cinotti, and A. Toninelli, "The design principles and practices of interoperable smart spaces," in *Advanced Design Approaches to Emerging Software Systems: Principles, Methodologies and Tools*. IGI Global, 2012, pp. 18–47.

[161] F. Morandi, L. Roffia, A. D'Elia, F. Vergari, and T. S. Cinotti, "Redsib: a smart-m3 semantic information broker implementation," in *Open Innovations Association (FRUCT), 2012 12th Conference of*. IEEE, 2012, pp. 1–13.

[162] A. Seaborne, G. Manjunath, C. Bizer, J. Breslin, S. Das, I. Davis, S. Harris, K. Idehen, O. Corby, K. Kjernsmo *et al.*, "Sparql/update: A language for updating rdf graphs," *W3c member submission*, vol. 15, 2008.

[163] A. D'elia, F. Viola, F. Montori, M. Di Felice, L. Bedogni, L. Bononi, A. Borghetti, P. Azzoni, P. Bellavista, D. Tarchi *et al.*, "Impact of interdisciplinary research on planning, running, and managing electromobility as a smart grid extension." *IEEE Access*, vol. 3, pp. 2281–2305, 2015.

[164] L. Bedogni, L. Bononi, M. Di Felice, A. D'Elia, R. Mock, F. Montori, F. Morandi, L. Roffia, S. Rondelli, T. S. Cinotti *et al.*, "An interoperable architecture for mobile smart services over the internet of energy," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*. IEEE, 2013, pp. 1–6.

[165] R. Bottura, A. Borghetti, F. Napolitano, and C. A. Nucci, "Ict-power co-simulation platform for the analysis of communication-based volt/var optimization in distribution feeders," in *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*. IEEE, 2014, pp. 1–5.

[166] R. Bottura and A. Borghetti, "Simulation of the volt/var control in distribution feeders by means of a networked multiagent system," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2340–2353, 2014.

[167] L. Bedogni, L. Bononi, A. D'Elia, M. Di Felice, M. Di Nicola, and T. S. Cinotti, "Driving without anxiety: A route planner service with range prediction for the electric vehicles," in *Connected Vehicles and Expo (ICCVE), 2014 International Conference on*. IEEE, 2014, pp. 199–206.

[168] APACHE Software Foundation. (2018) Apache jena. [Online]. Available: https://jena.apache.org/index.html

[169] ——. (2018) Apache jena arq. [Online]. Available: https://jena.apache.org/documentation/query/

[170] Eclipse Foundation. (2018) Sesame. [Online]. Available: http://rdf4j.org/

[171] Franz, Inc. (2018) Allegrograph. [Online]. Available: https://allegrograph.com

[172] CLiki the Common Lisp Wiki. (2018) Twinql extension. [Online]. Available: https://www.cliki.net/twinql

[173] CMMI Institute LLC. (2018) Cmmi model. [Online]. Available: http://cmmiinstitute.com/

[174] P. Varga and C. Hegedus, "Service interaction through gateways for inter-cloud collaboration within the arrowhead framework," *5th IEEE WirelessVitae, Hyderabad, India*, 2015.

List of Publications

R. Venanzi, B. Kantarci, L. Foschini and P. Bellavista, "MQTT-Driven Sustainable Node Discovery for Internet of Things-Fog Environments," *2018 IEEE International Conference on Communications (ICC)*, Kansas City, MO, 2018, pp. 1-6.
doi: 10.1109/ICC.2018.8422200
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8422200&isnumber=8422068

R. Venanzi, B. Kantarci, L. Foschini and P. Bellavista, "MQTT-Driven Node Discovery for Integrated IoT-Fog Settings Revisited: The Impact of Advertiser Dynamicity," *2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, Bamberg, 2018, pp. 31-39.
doi: 10.1109/SOSE.2018.00013
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8359146&isnumber=8359133

P. Bellavista, L. Foschini, R. Venanzi and G. Carella, "Extensible Orchestration of Elastic IP Multimedia Subsystem as a Service Using Open Baton," *2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, San Francisco, CA, 2017, pp. 88-95.
doi: 10.1109/MobileCloud.2017.31
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7944877&isnumber=7944854

M. Ornato, T. S. Cinotti, A. Borghetti, P. Azzoni, A. D'Elia, F. Viola, F. Montori, R. Venanzi, "9 Application system design: Complex system management and automation ", Delsing, J. (Ed.). (2017). IoT Automation. Boca Raton: CRC Press. Feb 2017, 281 - 316

M. Casoni, S. Monti, F. Sprotetto, A. Corradi, L. Foschini, R. Venanzi,"Semantic SPARQL queries: a novel federation model and implementation towards Enterprise Data Governance" *10th EAI International Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS 2016, Taormina, Italy, 25th-28th Oct 2016. ACM 2016, ISBN 978-1-63190-141-6. Pages 202-208
doi: 10.4108/eai.25-10-2016.2267042

C. Patsakis, R. Venanzi, P. Bellavista, A. Solanas and M. Bouroche, "Personalized medical services using smart cities' infrastructures," *2014 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, Lisboa, 2014, pp. 1-5.
doi: 10.1109/MeMeA.2014.6860145
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6860145&isnumber=6860015

Under Review

R. Venanzi, B. Kantarci, L. Foschini and P. Bellavista, "Fog-Driven Sustainable Context-Aware Architecture and Node Discovery for Internet of Things Environments," *IEEE Access Journal, regular transaction*.

R. Venanzi, B. Kantarci, L. Foschini and P. Bellavista, C. Stefanelli "Fog-Driven Context-Aware Architecture for Node Discovery and Energy Harvesting Strategy for Internet of Things Environments," *Hindawi, Wireless Communications and Mobile Computing,* special issue on *Fog-Cloud Computing Cooperation*

Recognition

Selected by IBM as best student for "*IBM Best Student Recognition Event 2015: Smart Food*", 8th - 10th July 2015, Amsterdam, Netherlands.