TECHNISCHE UNIVERSITÄT BERLIN

Application-derived Communication Protocol Selection in M2M Platforms for Smart Cities

- Engineering Doctorate Dissertation -

Asma Abdalla Elmangoush, M.Sc.

genehmigte Dissertation

Berlin 18. April 2016

TECHNISCHE UNIVERSITÄT BERLIN

Application-derived Communication Protocol Selection in M2M Platforms for Smart Cities

vorgelegt von
Asma Abdalla Elmangoush, M.Sc.
geb. in Alexandria

von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -
genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr.-Ing. Sebastian Möller (Technischen Universität Berlin)
Gutachter:	Prof. Dr.-Ing Thomas Magedanz (Technischen Universität Berlin)
Gutachter:	Prof. Dr. Neco Ventura (University of Cape Town)
Gutachter:	Prof. Dr. Axel Küpper (Technischen Universität Berlin)

Tag der wissenschaftlichen Aussprache: 18. April 2016

Berlin 2016

Acknowledgments

First and foremost, praise be to Allah the Almighty through whose mercy all good things are accomplished.

Throughout my past few years, a lot of people supported me professionally and personally, without their help I would not have accomplished this dissertation. Firstly, I would like to express my sincere gratitude to my supervisor Prof. Dr.-Ing Thomas Magedanz for his keen attention, guidance and encouragement. Also for giving me the opportunity to work at both the Technische Universität Berlin (TUB) and the Fraunhofer Institute for Open Communication Systems (FOKUS) and thus let me experience an environment of outstanding applied research. I would also like to thank Prof. Neco Ventura and Prof. Dr. Axel Küpper for the possibility to write this dissertation with their guidance.

I would also like to say thank you to all the people, who have contributed to this dissertation be it by discussion, inspiration, or different kinds of support. I thank all former and current members of the NGNI and AV research group for all the discussions and collaboration leading to numerous papers and research results being fundamental for this dissertation. I was supported by a team of active and inspiring colleagues, especially Adel Al-Hazmi, Niklas Blum, Hakan Coskun, Andreea Ancuta Corici, Ronald Steinke, Fabian Eichhorn and Alejandra Escobar Rubalcava.

I sincerely thank all my friends, who over the years have never failed to ask about my progress and encouraged me to continue my journey.

My special gratitude goes to my parents, sisters, brother, nieces and nephews. Thank you for the unconditional emotional support during the journey towards my goal, for this, I will always be in debt to you.

Berlin, April 27, 2016

Abstract

Machine-to-Machine (M2M) communication provides a new paradigm that aims to increase the level of system automation by enabling every physical and virtual object to be integrated seamlessly into a large-scale Smart City framework. The fact that connected objects represent activities related to all-day applications imposes different challenges to manage the heterogeneity of underlying technologies and application domains. Some applications in the Smart City context have critical requirements in terms of data latency and demanded throughput, such as eHealth and Smart Grid. However, current networks treat traffic generated by different applications in the same way regardless of the content or its source.

Recently, several divergent standards and protocols have been specified for M2M communication and the Internet of Things (IoT) service platforms. Each protocol focuses on a specific aspect of M2M communication. The lack of a protocol that can satisfy the heterogeneous requirements of M2M/IoT applications has resulted in a highly fragmented protocol stack in M2M/IoT systems.

Considering the variety in operating conditions and Quality of Service (QoS) requirements, it's impossible to depend on one protocol for all data streams or all applications. The core question addressed by this dissertation is which transport protocol should be selected for a defined M2M application. In this dissertation, a framework is introduced that enables the dynamically adaptation to transporting heterogeneous traffic of M2M applications and mediation with other M2M platforms. The objective of the proposed framework is to increase the adaptability of M2M nodes in transporting flows of requests from connected objects, be it resource-constrained or resource-rich, and different applications demanding heterogeneous QoS requirements. The concepts were integrated as additions to the Open Machine Type Communication (OpenMTC) platform to prove their value in a standard prototype architecture. An evaluation of the proposed concepts has been carried out providing a practical view on how to realize the proposed functionality as part of IoT systems. Additionally, the work has been verified within several European research projects and testbeds that address different issues related to Smart City realization.

Zusammenfassung

Machine-to-Machine (M2M)-Kommunikation bietet ein neues Paradigma, das darauf abzielt, das Niveau der System-automatisierung zu erhöhen, indem alle physischen und virtuellen Objekte in ein groß angelegtes Smart-City-Framework integriert werden können. Die Tatsache, dass verbundene Objekte Aktivitäten darstellen, die mit alltäglichen Anwendungen in Beziehung stehen, bringt unterschiedliche Herausforderungen mit sich, die Heterogenität der zugrunde liegenden Technologien und Anwendungsdomänen zu verwalten. Einige Anwendungen im Kontext der Smart City haben kritische Anforderungen an die Datenlatenz und den verlangten Durchsatz, wie zum Beispiel eHealth und Smart Grid. Allerdings behandeln die gegenwärtigen Netzwerke den Traffic, der von verschiedenen Anwendungen erzeugt wird, in der gleichen Weise, unabhängig von deren Inhalt oder Ursprung.

In jüngster Zeit wurden mehrere divergierende Standards und Protokolle für die M2M-Kommunikation und Internet-der-Dinge-(IoT)-Service-Plattformen spezifiziert. Jedes Protokoll konzentriert sich auf einen spezifischen Aspekt der M2M-Kommunikation. Das Fehlen eines Protokolls, das die heterogenen Anforderungen der M2M-/IoT-Anwendungen erfüllen kann, führte zu einem stark fragmentierten Protokollstapel in M2M-/IoT-Systemen.

Angesichts der Vielfalt der Betriebsbedingungen und Quality-of-Service-(QoS)-Anforderungen ist es unmöglich, von einem einzigen Protokoll für alle Datenströme oder alle Anwendungen abzuhängen. Die Kernfrage, die in dieser Arbeit gestellt wird, ist, welches Transportprotokoll für eine definierte M2M-Anwendung ausgewählt werden sollte. In dieser Dissertation wird ein Framework eingeführt, das dynamisch angepasst werden kann, um den heterogenen Traffic von M2M-Anwendungen zu transportieren und das die Vermittlung mit anderen M2M-Plattformen ermöglicht.

Das Ziel des AdM2M-Framework ist es, die Anpassungsfähigkeit von M2M-Knoten beim Transportieren von Datenströmen zu erhöhen. Das betrifft Anfragen von verbundenen Geräten, die bei den Ressourcen eingeschränkt oder auch ressourcenreich sein können, und auch verschiedene Anwendungen, die unterschiedliche

Anforderungen im QoS haben. Die Konzepte wurden als Ergänzungen zur Open Machine Type Communication (OpenMTC)-Plattform integriert, um ihren Wert in einer standardisierten Prototyp-Architektur zu beweisen. Eine Leistungsbewertung wurde durchgeführt, die in der Praxis zeigt, wie die beschriebene Funktionalität als Teil eines IoT-Systems zu verwirklichen ist. Darüber hinaus wurde die Arbeit in mehreren europäischen Forschungs¬projekten und Testumgebungen verifiziert, die verschiedene Sachverhalte im Zusammenhang mit der Realisierung von Smart Citys ansprechen.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

The telecommunication world started with the aim of connecting persons to each other through telegraph, telephone voice services (Human-to-Human (H2H)), and later evolved to connect persons to huge content and services over the Internet. Now we are witnessing the era of connecting various physical and virtual objects to the global Internet in a large-scale Internet of Things (IoT) infrastructure. The connected world is extending exponentially including more physical objects besides computers and smart phones in a global IoT. More than nine billion devices around the world are currently connected to the Internet, and estimations show that by the end of 2020, the number of connected objects will range from 50 billion devices to one trillion world-wide [1]. Intel estimates that around 26 smart objects for each human being will be connected to the Internet by 2020 [2].

Enabling the objects in our everyday working or living environment to communicate with each other and elaborate the information collected from their surroundings will make many Smart Services possible. Recently, IoT experts have pointed out Smart City services (e.g., health care, energy management, and transportation) as an emerging market with enormous potential [3, 4]. The main aspects toward devolving a Smart City , are to develop the Instrumented, Interconnected, Interoperated and Intelligent City [5].

The objective of building a Smart City requires the collaboration of various

stakeholders. Such collaboration is essential in order to increase the efficiency and efficacy of administrative services, and developing environment-friendly applications. Generally, Smart City represents an integrated system of several interoperated intelligent systems. Each system produces its own information and consumes others' information in a well-defined urban planning.

Conventionally, services in telecom were built within the network based on proprietary hardware/software solutions. Using Intelligent technology, the concept of service independent platforms was introduced. Later, object orientation and distributed middleware took off and Application Programming Interfaces (APIs) were introduced to allow for flexible service creation and making services implementation simpler by abstracting the underlying signalling protocols. To enable more advanced services in the Next Generation Network (NGN), intelligence has to be distributed among network elements. The IP-Multimedia Subsystem (IMS) was defined as a core network by the 3rd Generation Partnership Project (3GPP) and has been adopted in many commercial networks, to provide a framework for enhanced and distributed service delivery over IP, independent of the access technology. The Service-Oriented Architecture (SOA) principle extends these concepts. It does not specify any API or overlay architecture but rather refers to the use of services as individual building blocks to create an enriched end-user experience. In the IMS architecture these building blocks or service enablers are hosted on Application Servers (ASs) and the IMS acts as a mere docking station for these services. The use of middleware layer implementation of open interfaces between network and service layer is an acceptable overhead compared to the indisputable gains of exposing network functionality through open APIs [6]. However, the target of the communication services was always human-being with smart end-devices in an H2H/Human-to-Machine (H2M) communication interaction. The recent trend of connecting all physical and virtual objects to the Internet with limited or no human intervention (see Figure 1.1), bumps up new service's characteristics and requirements that enable a higher level of automation.

Typically, the IoT can be seen as an umbrella term for interconnected technologies, devices, objects and services. It is not a single technology, rather a concept in which almost all physical objects are connected and enabled. For example, public transport buses being networked and enabled with embedded sensors, image recognition functionality, and near field communication integrated into situational decision support, asset management and more innovative services. The successful development of such systems requires ubiquitous sensing networks, efficient communication infrastructure over reliable M2M platforms [7], and intelligent data processing capability.

Unlike traditional H2H or H2M services, which mainly involve multimedia sessions, messaging, web browsing and remote control, M2M communication provides a new paradigm that aims to increase the level of system automation. The H2H/H2M traffic are mostly downstream traffic that requires a significant amount of bandwidth and occurs on limited periods controlled by a human. In contrast, M2M traffic is mostly in the upstream direction, generated from connected devices with

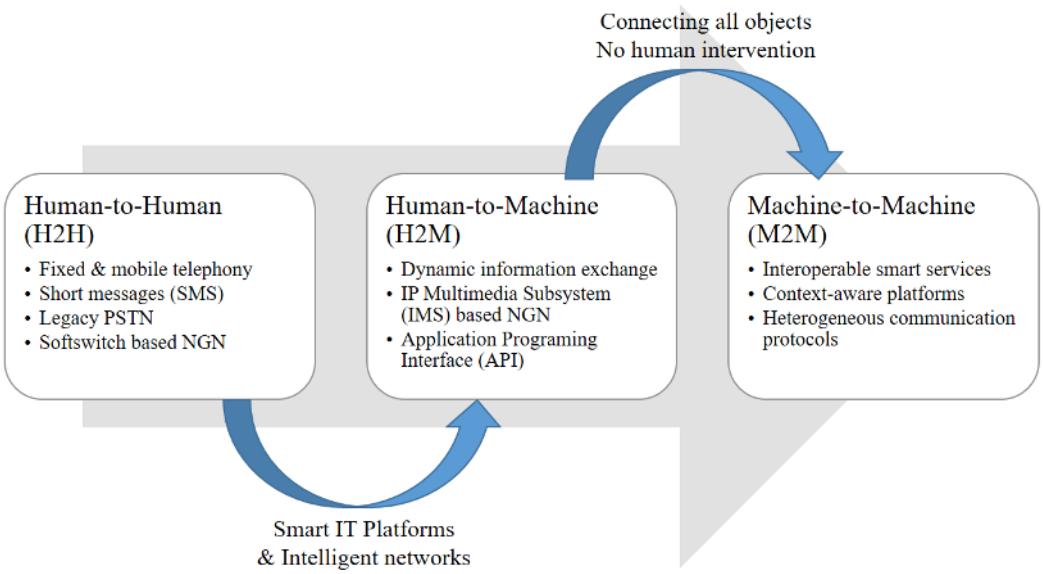


Figure 1.1: Communication Evaluation from H2H to M2M

a huge variety of payload size and frequency rate due to the heterogeneity in both devices' capabilities, and the content nature [8]. Additionally, M2M applications should behave and assume their roles without any human intervention. Therefore, self-sustaining for long periods is an important feature for M2M.

The current Internet paradigm has been built around host-to-host communications, a concept that is now a limiting factor for the current use of the Internet in the context of IoT. Converting existing technologies towards enabling M2M communications is required to overcome the limitations of current frameworks that were designed to support H2H and H2M communication mainly [9]. The majority of available wireless communication systems, including Long Term Evolution (LTE), are intended to support a small number of personal devices with continuous information flow on time-scale [10].

The principle of limiting human interaction in M2M systems as well as integrating a massive number of devices with heterogeneous computational capabilities into it, demands a fully interconnected and application-agnostic system. The main challenges for M2M communication are driven from integrating low-power devices and low-bandwidth networks. The heterogeneity of integrated communications technologies, targeted service domain and data representation, highlights the need to study the communication requirements of various services and the traffic patterns in the system.

In most existing M2M implementations, the application logic and devices/sensors are tightly coupled as the design relies on requirements that were defined on a case-by-case basis. This has resulted in vertical silos of proprietary solutions that are difficult to integrate across application domains [11], as illustrated on the left

side of Figure 1.2. While the specific design per use case has its advantage in terms of security and reliability, it poses a significant limitation in the scalability and interoperability of the M2M market. In order to overcome these limitations, the concept of horizontal service layer is adopted to change the system as illustrated on the right side of Figure 1.2. Different standards try to provide a general solution for M2M applications by proposing a generic service layer with well-defined interfaces between network, devices and applications. On the one hand, access through such interfaces allows applications as well as devices to interoperate seamlessly, by facilitating the discovery process of connected devices and the usage of each other's resources on demand. The need to standardize M2M platforms and enablers to bridge these vertical silos was recognized by various Standards Developing Organizations (SDO) that has recently promoted standardization activities in the M2M domain. Thus, the standard defines a horizontal "middleware" for diverse M2M applications on top of heterogeneous sensor technologies. On the other hand, the concept of connected "Things" has been introduced by some IoT research activities [12, 13], to bring the awareness of Things-level knowledge into the M2M middleware. This will enable applications to interact based on the Things representations (e.g., the room temperature) rather on device-level representations (e.g., a sensor measurement).

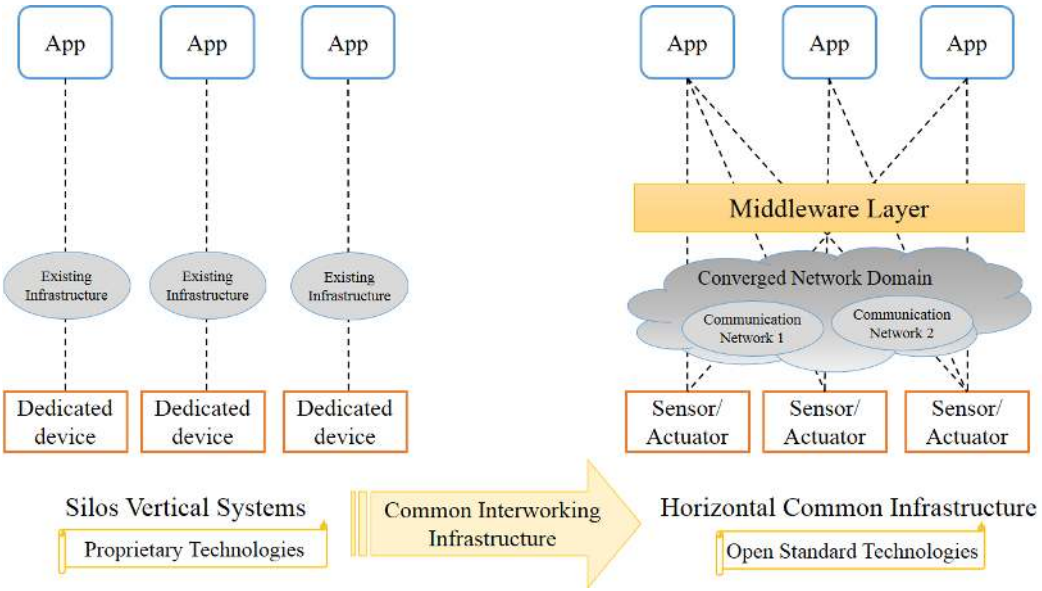


Figure 1.2: From the Existing Silos Towards Interworking Infrastructure

1.2 Related Terms and Definitions

In order to provide the reader the required terminologies and definitions of key fundamental terms related to the topic of this dissertation, this section introduces the most relevant terms and definitions used within this dissertation:

Machine-to-Machine (M2M): *"A form of data communication which involves one or more entities that do not necessarily need human interaction."* [14].

Internet of Things (IoT): the term IoT oriented to the connectivity approach, is defined by ITU - Telecommunication Standardization Sector (ITU-T) in [15] as, *"A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies"*. A very similar definition adopted by the European Research Cluster on the Internet of Things (IERC) states that 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"* [16].

Adaptability: refers to the capability of a system or component to be modified for use in applications or environments other than those for which it was specifically designed [17].

Extendability: refers to the ease in which a system or component can be modified to increase its functional capacity [17] .

Reliability: is defined as *"The ability of a system or component to perform its required functions under stated conditions for a specified period of time"* [17].

Interoperability: is defined as *"The ability of two or more systems or components to exchange information and to use the information that has been exchanged"* [17].

Scalability: is the ability of the system to handle the increased or expanding workload and resource demands.

Smart City: refers to a city connecting the physical infrastructure, the IT infrastructure, the social infrastructure, and the business infrastructure to leverage the collective intelligence of the city [18].

1.3 The Evolution of M2M Communication

As mentioned above, the research on M2M communication focuses on supporting the ubiquitous sensing and autonomous communications of all kind of objects that are capable of connecting to the Internet. In Figure 1.3, a high-level architecture of an M2M system is depicted and highlights the three domains of any typical M2M system. These are: the device field domain, the infrastructure domain, and the application domain.

The device field domain consists of the physical endpoints, which could be simple small tags or more complex items with advanced processing capability and embedded

intelligence. There are multiple connectivity options available that could support the ubiquitous sensing in M2M. In all cases, any object should have connectivity capabilities to interact with the other objects and applications. Although the communication stack on all M2M systems should be Internet-enabled, the deployment of the Internet Protocol (IP) protocol could be very expensive or power consuming for simple battery-powered objects. Based on the connectivity capability of the end-device on this domain, three implementation options are possible:

- Capillary networks relaying the IP connectivity through one or more gateways. The end-objects are mainly non-IP capable depending on point-to-point connectivity such as M-BUS, and IEEE 802.15.4, or on mesh connectivity over ZigBee, 6LowPan or similar technology. This level of connectivity enables the data routing to an IP-enabled gateway that provides network management capabilities and data aggregation, hiding the complexity of this domain from the rest of the M2M layers. Several technologies could be used to provide connectivity to the gateway, such as Wi-Fi (IEEE 802.11), Ethernet (IEEE 802.3), and cellular (GPRS, EDGE, UMTS, HSxPA, and LTE in 3GPP networks). Other technologies like WiMax (IEEE 802.16), PLC (power-line communications), fiber optics (e.g., FTTx and HFC), and xDSL might be used, but these are less popular in emerging M2M deployments [19].

- Direct connection using IP-enabled devices that are able to connect through Wi-Fi, Ethernet or cellular modem. In this case, no gateway is required as the end-devices are equipped with an IP-based connectivity and able to autonomously interact with the next level. Additionally, such devices should have a high-level of computation capability and memory footprint, in order to support more protocols on top of the IP stack handling the transporting and session management.

- Recently, new technical solutions have been introduced in M2M with the objective of simplifying the deployments and gaining wider network coverage without compromising cost or power consumption. New M2M protocols and operators have been implemented that present an IP-based backhaul where the end-devices are provided with direct connectivity to a base station such as Weightless [20], and SIGFOX [21].

The infrastructure domain consists of the M2M capability and service infrastructure. The M2M system involves many stakeholders, such as distinct service providers, and core network providers. In order to enable the management of the overall system consistently, flexible horizontal solutions are needed for sharing skills, network infrastructures, and devices between stakeholders [22]. In 2009, the European Telecommunications Standards Institute (ETSI) created a Technical Committee (TC) whose standardization work is mainly focusing on the service middleware layer. The ETSI M2M Release 1 standards, finalized in 2012, enables integration of different M2M technology alternatives into one managed platform. The ITU established two Focus Groups [23] related to M2M services. The Focus Group on the

M2M service layer (FG M2M), to identify key requirements for a common M2M service layer; and the Focus Group on Smart Sustainable Cities (FG SSC), aims to exchange knowledge in the interests of identifying the standardized frameworks needed to support the integration of ICT services in Smart Cities. Later on mid-2012, a consortium of seven standard development bodies, including ARIB (Japan), ATIS (U.S.), CCSA (China), ETSI (Europe), Telecommunications Industry Association (TIA) (U.S.), TTA (Korea), and TTC (Japan) have set up a new global organisation (oneM2M) to avoid competition between M2M standards. Chapter 3 provides a review of the main efforts and outcome of the standardization activities in M2M.

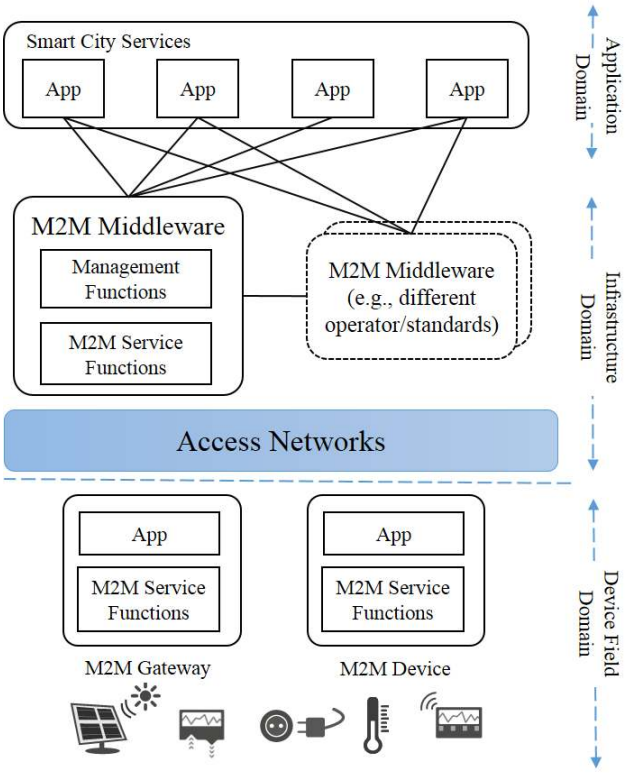


Figure 1.3: High-Level Architecture of M2M system

The application domain consists of interconnected service provider able to collect data, route commands and messages, and manage connected devices. Over this layer, a set of services might be offered in order to build applications that enable the interaction with the implemented system. There are ongoing standardization activities in the M2M service platform and overlaying APIs, which promote the development of M2M Services by abstracting the M2M platform to the application layer. This shields developers from underlying technology and as such reduce the efforts required for service development [24]. The use of Representational State Transfer (REST) architectural style, which makes information available as resources

identified by Uniform Resource Identifiers (URIs), is widely used in M2M applications. On the one hand, standard bodies such as ETSI and oneM2M have specified the M2M service layer based on a REST architecture, where the communication over standard interfaces is independent of the transport protocol. On the other hand, a number of publish/subscribe protocols have been proposed for wireless sensor networks recently [25]. Although they support essential features required for constrained M2M devices, such as low overhead, some are customized protocols, i.e., either built for a particular application without standardization in the communication protocol, or unsuitable for real-time services.

1.4 Problem Statement

The vision of the Future Internet is to enable objects to be connected any-time, any-place, with any-thing and any-one ideally using any-path/network and any-service. From a technical point of view, this vision could not be accomplished by implementing one novel technology; instead, several complementary technical developments shall provide functionalities and capabilities to assist in bridging the gap between the virtual and physical world. The success of IoT can be unleashed based on three components: Content, Things, and Connections, as shown in Figure 1.4.

The M2M communication intends to promote seamless interaction processes between connecting objects (things) to enable the automation of decision making based on aggregated data. Numerous challenges in all IoT components are still open to provide reliable Smart City services; the authors of [26, 27, 28] discussed a number of these challenges and open research issues.

The main limitations of existing systems to enable the deployment of reliable Smart services are:

- The fact that connected objects represent activities related to all-day applications (e.g., health care, energy management, etc.), imposes additional challenges to provide reliable services. Some applications in the Smart City context have critical requirements in terms of data latency and throughput, such as EHealth or SmartGrid [29]. However, the current networks treat traffic generated from different applications in the same way regardless of the content or the content's source. Data generated by EHealth applications require different handling than other applications, due to the variety of demanding in terms of QoS. Thus, understanding the M2M/IoT traffic characteristics is essential to implement reliable Smart services.

- Due to the nature of many objects connected with M2M systems being simple tags or resource-constrained devices, low power consumption protocol stack is required to allow battery-powered objects to plug-into the Internet flexibly. The current M2M protocol stack is highly fragmented. Typically, M2M applications utilize IP-compatible open protocols that are standardized, in order to be widely deployable. Recently, various protocols have been proposed

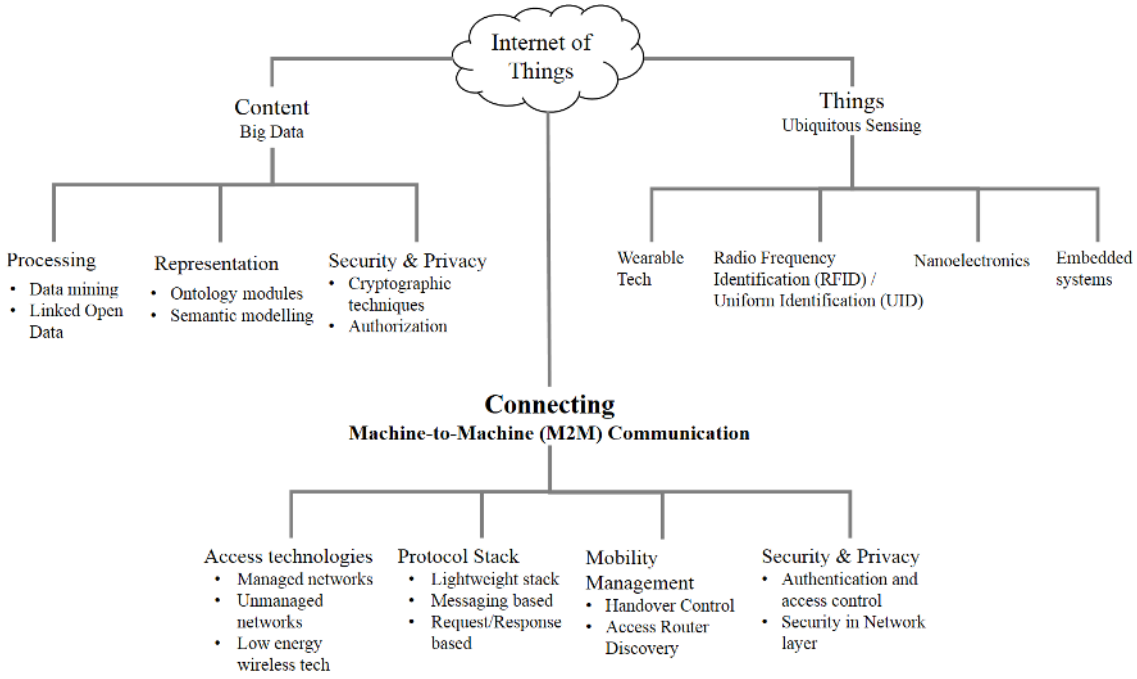


Figure 1.4: Taxonomy of Internet of Things Components

to the M2M development, aiming to address the requirements of integrating resource-constrained devices and supporting ubiquitous access. Nevertheless, there are no sophisticated guidelines for the protocol stack deployment for the future M2M services.

- From the connected object perspective, there are a number of interaction models to the system, communication styles, and transport protocols that can be applied for data pushing; one-to-one, one-to-many and many-to-many. However, changing the interactions models of an object can have more impact on the performance and the efficiency than the communication protocol used for that interaction.

- The estimations of the numbers of connected objects in the coming decade are showing a rapid growth. This promotes the need of scalable technologies for managing and analyzing collected information from billions of objects. Current systems are not designed to simultaneously serve the aggregated traffic accrued from a large number of devices. For instance, current systems could easily serve five devices at 2 Mb/s each, but not 10,000 devices each requiring bandwidth of 1 Kb/s [30].

- There are many standard organizations working in specifying a horizontal M2M middleware layer which combines various technologies. While the use of standard-based APIs supports interoperability across heterogeneous plat-

forms [31], appropriate binding to transport protocols is needed to provide an efficient service, considering the variety of communication requirements from different applications.

This dissertation focuses on overcoming the most fundamental of these limitations, which are eventuated from the fragmented technological landscape and protocol stack for M2M communication. To cope with these challenges, the next sections discuss the scope of this dissertation which deals with problems related to the increasing variety protocols and standards, the analysis of interaction models and the selection of the suitable protocol for M2M applications.

1.5 Dissertation Target and Scope

The previous Section 1.4 creates a broader picture for understanding the challenges of deploying large-scale M2M systems. This section defines the scope of the work in this dissertation and positions the scientific contribution.

Following the vision of large-scale sensing framework, the objectives of this dissertation are twofold: first, gain a better understanding of the available standardized protocols for M2M in order to define guidelines for selecting a proper communication channel and transport protocol based on the application's communication criteria and requirement. Second, to enable the interoperability of the M2M service layer by defining interworking proxies to other M2M/IoT platforms.

By examining the communication requirements of some Smart City services, we aim here to provide an accurate description of key protocols that are being standardized and used in the M2M implementation recently. Along with basis of selecting the proper one that satisfies the requirements of a given Smart Service. To allow the deployment of reliable Smart City service in large scale level, the following major aspects should be considered:

1. The dissimilarity in resources and capabilities of the connected objects. This includes the computation power, storage and energy power.

2. The different interaction models that are possible between M2M nodes (i.e. devices, gateways and core servers).

3. The heterogeneous traffic patterns generated from different M2M nodes.

4. Various transmission channels and access technologies emerging to support the seamless mobility of M2M nodes.

5. The semantic analysis of the aggregated data in order to create situation awareness and enable applications and machines to understand their surrounding environments.

As the first three points represent the core requirement to handle the M2M traffic delivery, they are the main subjects of this research, while the last two research

aspects are considered for future work. Motivated by the special requirements of M2M applications, characteristics of M2M protocols and the lack of comprehensive measurements studies on the protocol deployment, this dissertation develops a framework for Adaptable M2M (AdM2M) Transportation. Figure 1.5 illustrates the scope of the overall research framework following the IoT reference model defined by the ITU-T [15].

The scope of this thesis relates to the definition of an adaptable transport framework, named AdM2M which dynamically manages the transport capabilities of M2M nodes and address the interoperability with incompatible M2M platforms. The transport capabilities focus on providing connectivity for the transport of M2M/IoT service and application specific data information, as well as the transport of control and management information. The developed functionality aims to address the transport stratum of the M2M middle and infrastructure nodes, leaving out of scope the networking access technologies of the M2M node and platform. As a proof-of-concept, the implementation of the system will cover only two protocols that address needs of both resource-constrained resource-rich devices. Additionally, platforms interoperability is considered as one specific capability required to support large-scale deployment of variant M2M/IoT standard platforms.

A number of domains in the Smart City context demand the timely and immediately actionable services. Therefore, it is important to support reliable data exchange between inter-operated domains for efficient large-scale deployments.

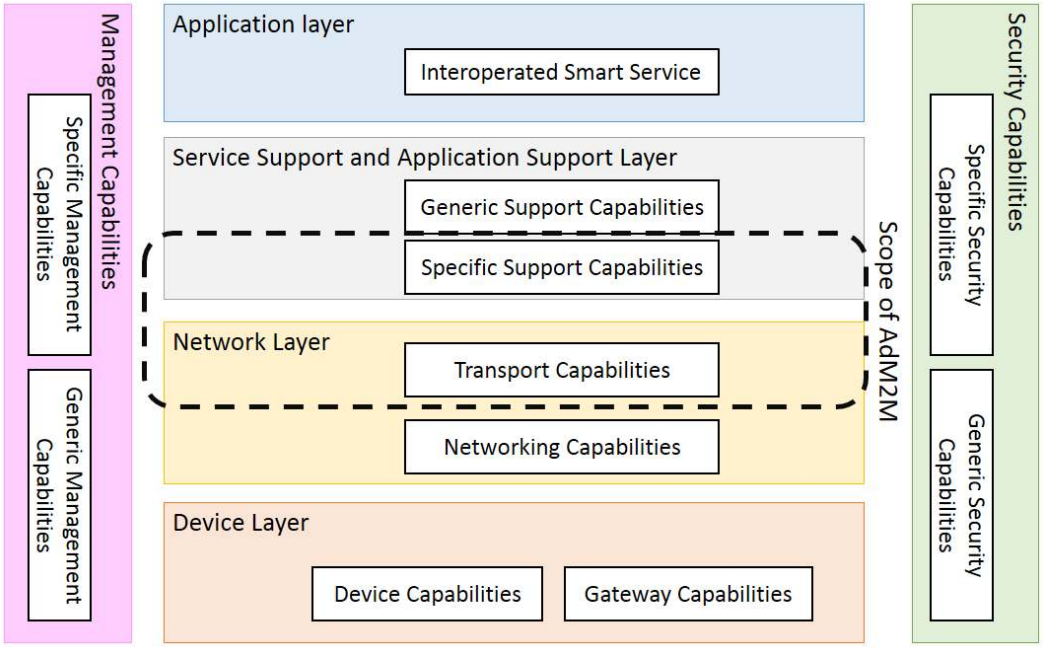


Figure 1.5: Overall IoT Architecture and Scope of Research Framework (based on [15])

Two main research questions are intended to be answered by this thesis:

Q1: *Are the current standardized protocols sufficient to support different kinds of Smart City services including those which comprise real-time M2M streams?*

Q2: *To which level could the interoperability of M2M application-agnostic platforms be realized?*

To answer these questions, this dissertation makes the following contributions:

1. Gap analysis of service's requirements and the protocols performance in order to define a guideline for selecting the proper protocol.
2. Design and specification of basic core functionalities for the AdM2M framework, as depicted in Figure 1.6, enhancing the M2M event and data message exchanging.
3. The implementation of a validated generic M2M platform that has been used deployed within different research projects in the context of Smart City services.

The concept development, experimentation and evaluation results will be published in scientific proceedings and participation with specific chapters to technology related books.

1.6 Research Methodology

It is envisaged the research include a practical research element to investigate questions of ensuring reliable Smart Services using M2M platforms. Figure 1.7 illustrates the trends and influences on the thesis, which are basically the M2M/IoT ecosystem and new requirements, the state-of-the-art technologies and the current research trend toward future ubiquitous sensing. The research methodology will provide a fivefold approach addressing problems within the scope described above:

1. State of the Art review of existing M2M/IoT standardization activities and M2M middleware specifications toward a common M2M service layer that provides end-to-end service delivery and integrates heterogeneous devices and technologies. Findings of this step are published in [5, 32, 33, 34]
2. Study of Smart City service's requirements from the perspective of the network operator, service provider, service developer, and end user, which characterize an executable practice for Smart City. Findings of this step are published in [24, 35, 36, 37, 29].
3. Identification of key traffic patterns and interaction models required for selected Smart services use cases, and propose a guideline of using M2M transport protocols in order to fulfill operation requirements. Findings of this step are published in [38, 39, 40].

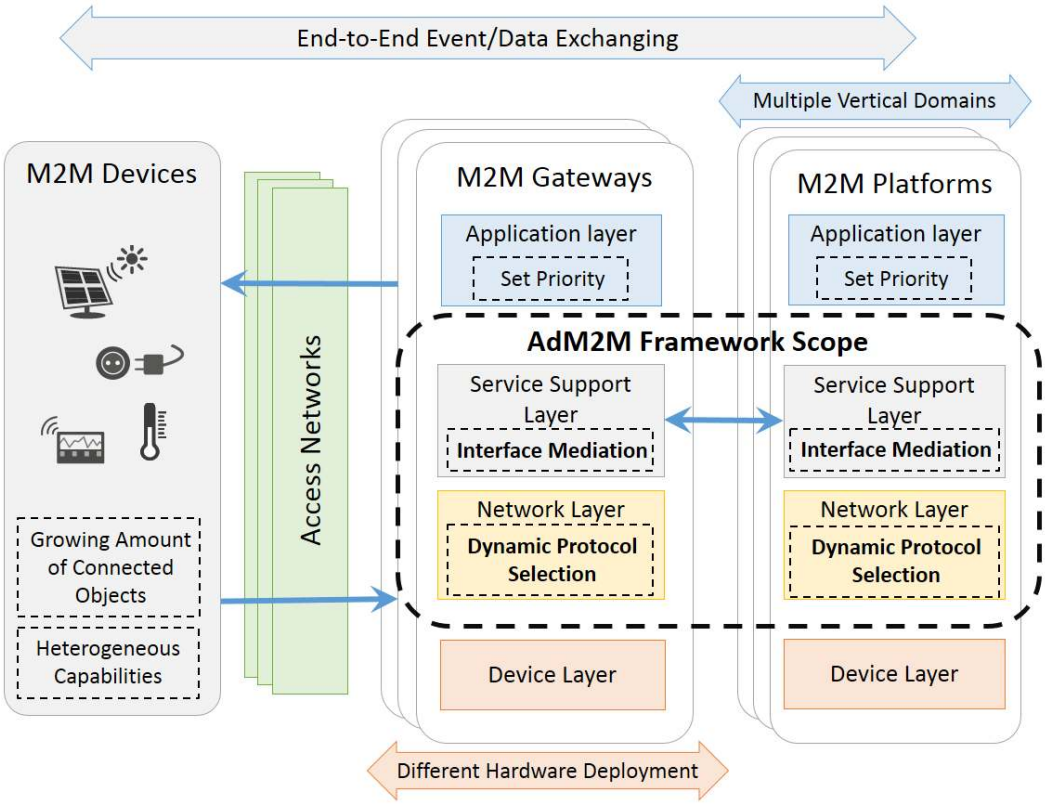


Figure 1.6: Overall AdM2M Framework Scope

4. Specification and implementation of the AdM2M framework providing different transport protocol stacks as plug-ins to support the communication with M2M devices using the proper stack to each use case, and integrating into the Fraunhofer FOKUS Machine-Type Communication (OpenMTC) platform [41]. The work conducted for this step is published in [42, 43, 44, 45, 46].

1.7 Dissertation Outline

The rest of this dissertation is structured as follows:

- **Chapter 2 - Fundamentals of M2M Communication:** provides an overview of the M2M communication concept and fundamentals. Definitions of M2M communication from the literature are provided followed by discussing the basis of the M2M middleware service layer and M2M traffic models. Finally, the summary highlights the challenges and related technologies in this type of communication.

- **Chapter 3 - State-of-the-Art:** surveys the State-of-the-Art in context of this dissertation. Standardization efforts in the field of M2M service capabili-

ties, transport and application protocols, and access technologies are reviewed. Additionally, an overview of some research projects activities in M2M/IoT context are presented.

- **Chapter 4 - Communication Requirements Towards Reliable Smart Service Deployment:** presents the process of identifying the functional and non-functional requirements. Different M2M services are studied to define the challenges and requirements toward realizing reliable deployment based on M2M platforms.
- **Chapter 5 - Design and Specification:** presents the design and specification of the Adaptable M2M Transport (AdM2M) framework based on the outcomes and findings of previous chapters.
- **Chapter 6 - AdM2M Framework Implementation:** describes the implementation of the framework and the applied tools and technologies.
- **Chapter 7 - Evaluation:** presents the evaluation work and comparison with other approaches. The implementation evaluation is carried out at Fraunhofer Institute for Open Communication Systems (FOKUS) within the context of several research projects.
- **Chapter 8 - Conclusions and Further Work:** gives insights of this dissertation and introduces open research issues for future work.

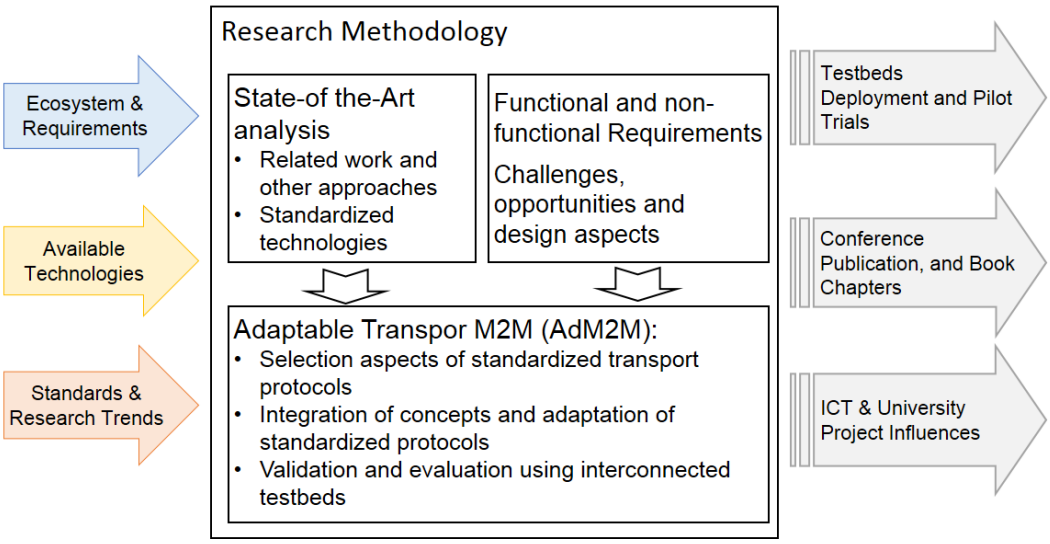


Figure 1.7: Dissertation Trends, Influences and Methodology

2

Fundamentals of M2M Communication

This chapter gives an overview of the M2M communication concept, challenges and related technologies. Ongoing research and activities in the M2M field are related to application and data handling, connectivity management, and device controlling. In order to elaborate on the targeted problem and place the contribution of the dissertation in context, the main focus of this chapter will be on the connectivity aspect and its related issues. After presenting the definition of M2M Communication, the principles behind the M2M middleware service layer and its main functionalities are provided. The focus is subsequently narrowed down by presenting an overview of the M2M traffic taxonomy. Parts of this work have been published in [39, 40].

2.1 Introduction

The concept behind the M2M communication is highly related to the Supervisory Control and Data Acquisition (SCADA) systems that have been utilized in controlling industrial processes since the 1970s. SCADA refers to a system that gathers and analyses real-time data from various equipment at a factory, plant or in other

remote locations. The data is then presented through the Human Machine Interface (HMI) to end users in graphical form so they can control and monitor the SCADA system as needed. The term has been used broadly to portray control and management solutions in a wide range of industries. Usually, SCADA systems are based on proprietary technologies that make them costly and difficult to maintain in widespread deployments [47], while M2M communication is based on IP and a big set of standardized access technologies such as IEEE 802.11 wireless LANs and cellular communications. However, there is ongoing work to enable SCADA protocols over IP protocol and allowing SCADA components to scale up [48].

Figure 2.1 shows the continuous workflow performed by a typical M2M system. First, the sensors devices perform the data acquisition task from the sensed environment (e.g. temperature, humidity and flow measurement). Then, the devices process the aggregated data and take decisions, which requires computational power capabilities to manage the decision-making functionalities. Finally, some devices (actuators) execute actuation tasks (e.g., alerts/information, or commands to actuators).

In this regards, the M2M related technologies could be classified into three aspects as shown in Figure 2.2; these are application and data handling, connectivity, and device controlling. For the scope of this dissertation we will concentrate on the connectivity aspects.

The M2M communication is emerging in a wide range of domains and shaping the development of services in the Future Internet (FI) sector. The M2M communication technology is representing an essential part of the IoT concept, where machines communicate to each other with limited or no human interaction. This enhances the connectivity of Any-thing or Any-one, Any-time and Any-place via

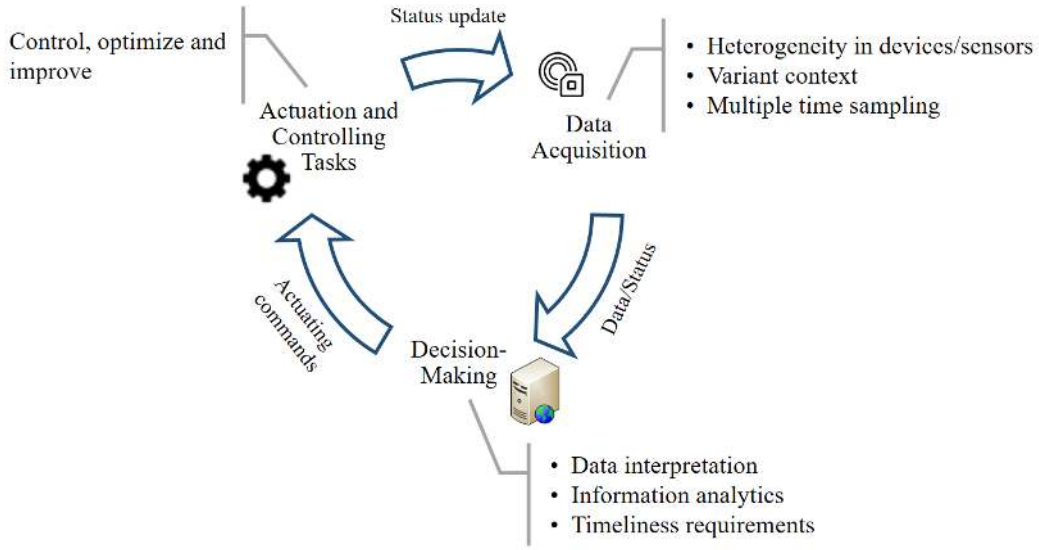


Figure 2.1: A General overview of M2M Workflow

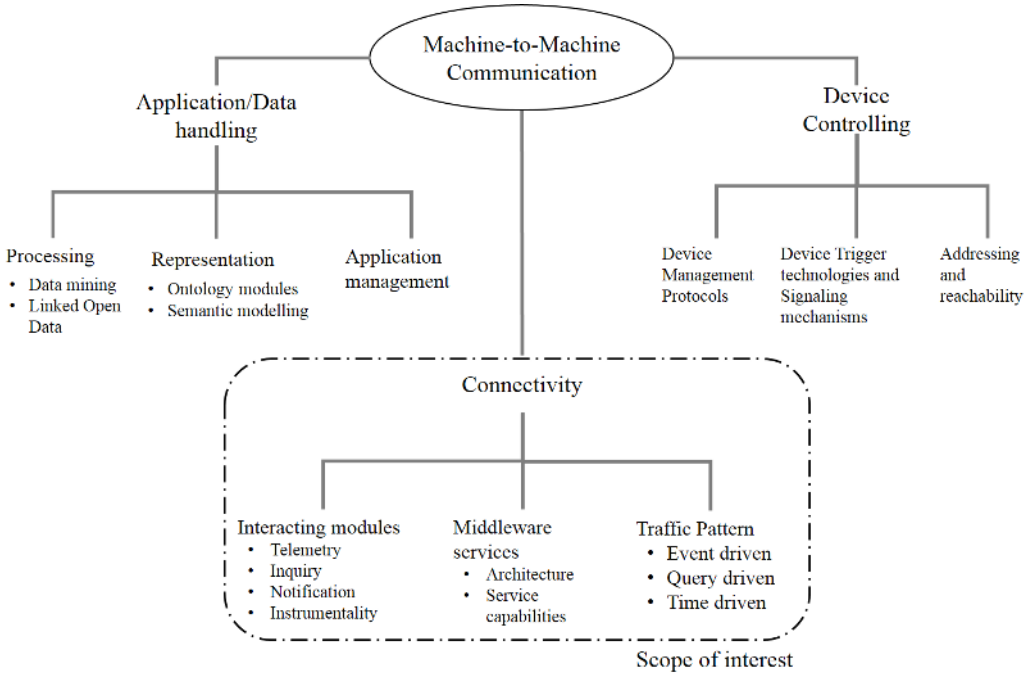


Figure 2.2: A Taxonomy of M2M Aspects

Any-path/network to use Any-service. The primary value that M2M creates is a direct result of the data that can be captured from connected things - and the resulting insights that drive business and operational transformation.

The motivation of this new trend is two-fold: technical and economic. On one hand the advancement of semiconductor industry shrinking lithography continues to reduce chipset cost and power consumption and embeds more sensors into devices used in different aspects of our daily life. On the other hand, the technology evolution in the Internet and advanced wireless networks make it possible to provide broadband data service at a significantly lower cost per bit transferred than in the past. In addition, recently the mobile market has become saturated and highly competitive, which raises the need to introduce new potential services to fill the revenue gap.

2.2 M2M Communication Definition

The term Machine-to-Machine (M2M) has been used for a long time to describe quite broad machine (nodes) connections across the wireless and wireline domains, where the connectivity for deployed endpoints can be provided via various technologies, i.e. cellular/mobile, satellite, wireline/Internet, etc.

IEEE 802.16p task group defines M2M communication as: *"The information exchange between user devices through a Base Station, or between a device and a*

server in the core network through a Base Station that may be carried out without any human interaction." [49].

More generally, ETSI defines M2M as: *"The communication between two or more entities that do not necessarily need any direct human intervention."* [50].

The 3GPP uses the term MTC instead and define it as: *"A form of data communication which involves one or more entities that do not necessarily need human interaction."* [14].

All these definitions emphasise the limited human intervention to this paradigm of communication that aims to increase the level of system automation and facilitate data exchange. Thus, the self adaptation in M2M nodes is a major requirement. In this dissertation, the terms M2M and MTC are used equivalently.

2.3 M2M Middleware Service Layer

The main goal of M2M/IoT platforms is to connect efficiently the growing number of devices, and associate them to a set of applications addressing use cases from different industrial domains such as energy, automotive, health, transportation etc. As discussed in Section 1.1, the need to exchange information between actors at different domains in a Smart city, has motivated the approach of integrating an M2M/IoT service layer middleware that mediate the communication between these systems and enable interoperability of heterogeneous services and technologies.

As illustrated in Figure 2.3, developing a large-scale Smart environment based on M2M communication, demands interoperability at all layers, between objects/devices, platforms and gateways, and Smart services. Considering the massive amount of objects to be part of the Internet in the next decade [2] and the wide range of access technologies to be supported, the management of the connectivity channels and data traffic of each device is challenged. It is important to solve the problems emerging from the existing M2M solutions, which are incompatible and difficult to extend for integrating Any-Thing and/or Any-One.

To this end, the need of a middleware M2M platform in FI infrastructure has been recognized in research Institutions and Academia in order to provide reliable transportation and data exchange via Any-Path in the M2M system. Several standards are emerging in this direction, but the compatibility with other solutions remains as a main requirement to support the seamless integration of M2M/IoT services. Since the essence of the IoT is the interconnection of the physical world of things with the virtual world of Internet, the software and hardware platforms as well as the standards commonly used for enabling such interconnection may become the core of an IoT ecosystem [51].

Furthermore, the IoT has huge potentiality for developing new innovative applications in Smart Cities as well as in many other fields. Authors in [4] presented the idea of a smart city as a "System of systems", where the integrated systems form a closed loop and are characterized by functions: sensing, information management, analytics and modeling, and influencing outcomes. Each system produces its own

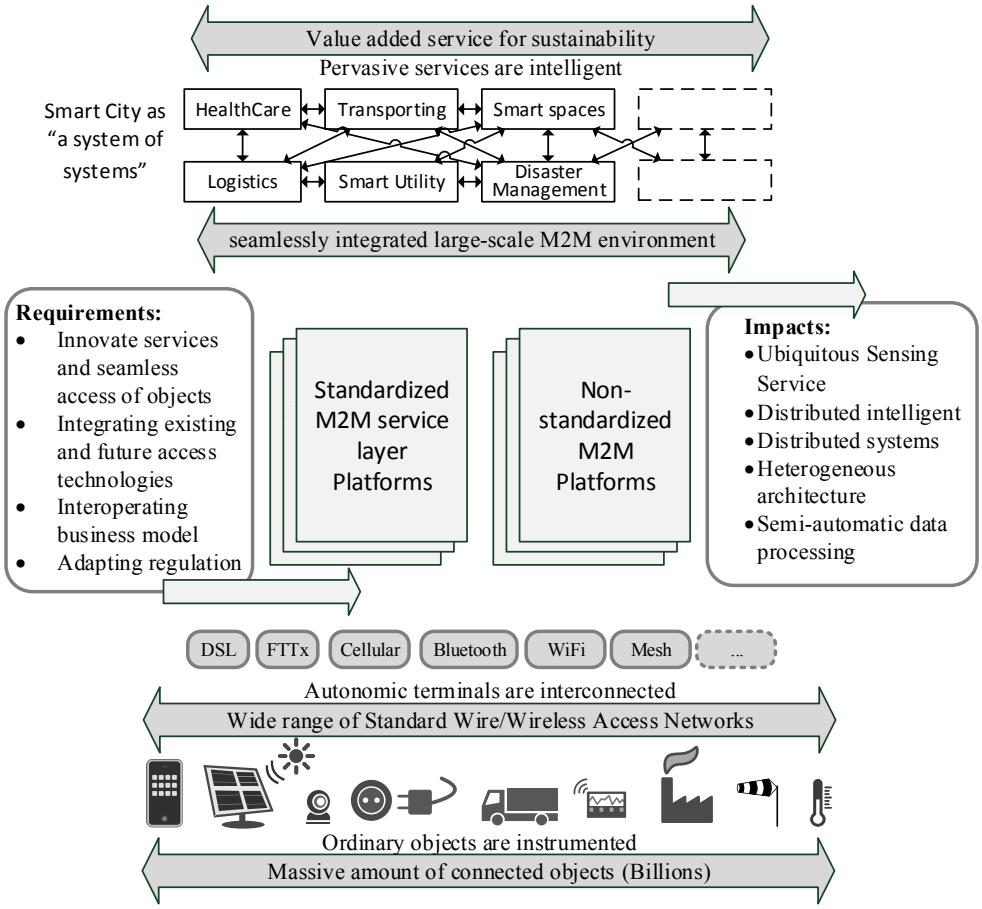


Figure 2.3: High-Level Overview of Smart City System

information and consumes others' information in a well-defined urban planning.

2.3.1 M2M Stakeholders

M2M systems are inherently multiple stakeholder systems. The stakeholders in an M2M system are entities that facilitate and/or participate in the legitimate operation of the system [52]; examples include:

- Manufacturer of M2M Devices, motivated by the possibility to expand their trading relationship to cover a completely new demand level, by providing M2M-ready hardware able to sense, act, or manage something on one side and connect over a capillary network or the Internet on the other.

- M2M Service Platform, as an enabler for different domains integration by providing the linking tools between developers and users.
- Network operators challenged by the fact that providing connectivity alone will not be sufficient to generate revenues.
- M2M Service Providers utilizing the deployments of previous actors to develop a full vertical M2M service.
- User application developer, who develops M2M applications over custom implementations or a given middleware.
- Device owner who buys and maintains the connected objects.
- User/Consumer of the M2M solution either directly by using the M2M device or indirectly by using the service or data.

2.3.2 Architecture Principles

Horizontal Middleware: The horizontal middleware layer approach enable building a coherent framework valid across a large variety of business domains, networks, and devices. Middleware layer interposed between the networking and the application level hides the details of different technologies to keep developers away from issues that are not directly pertinent to their focus, which is the development of specific application. Additionally, this middleware layer makes the data independent from the network access, which in turn facilitates the interoperability among different network technologies.

All-IP Open Architecture: Despite the fact that IP-based protocols requires large memory footprint, the IP-base approach is gaining more momentum in large-scale deployments in the context of M2M. The use of IP in M2M systems enables connected objects to be addressed from anywhere and use existing Internet services, such as email, Internet telephony and video streaming. Besides, new features could be built on existing functionality such as global interoperability, network-wide data packet delivery (forwarding and routing), data transport across different physical media, naming services (URL, DNS) and network management [53]. Due to the unprecedented growth in the number of connected objects, it will be important to provide these objects with automatic capabilities for self-management, self-healing, and self-discovery. Using IPv6 will provide a large space address and allows the automatic address configuration, enabling smart objects to assign their own addresses.

Application Agnostic Framework Offering services to control connected devices and handle the aggregated data in an interoperability manner, raise many challenges in designing open and standardized service enablers. In M2M systems, service enablers should facilitate the harmonization of services into interoperable applications via standard open interfaces. This shall enable the remote deployment of new services on run-time satisfying application's needs.

2.3.3 Main Functionalities of an M2M platform

The M2M operator platform should offer a set of functionalities that allows M2M applications to access, manage and configure devices and gateways. These functionalities act as enablers, since they enable third parties, such as service providers, to offer added value services based on the functionality of the operator platform. Main functionalities required are as follows [32]:

- Connectivity management: Providing ubiquitous computing and communications is the main object of M2M platforms, therefore supporting multiple protocols and sensor technologies is essential. Additionally, communications networks shall be optimized to support the new M2M interactions and traffic patterns. M2M applications will have bursty traffic at regular intervals or in the trigger of node's events. Furthermore, various transport protocols are used to carry M2M data according to diverse traffic patterns and profiles.

- Device management: New technologies are needed to facilitate the interaction between a decision making server and actuator nodes (clients) to replace the SMS-based protocols, which are still used for controlling devices over legacy systems. The solution shall offer convenient methods to discover, control and manage massive number of devices (sensors and actuators). Furthermore, an abstraction level of representing and virtualizating these devices is required, in order to enhance the interoperability over heterogeneous platforms.

- Application management: Offering innovative services to control connected devices in an interoperable manner, raises various challenges in designing open and standardized service enablers for M2M communication. The M2M platforms shall support abstract application development and ongoing management over open APIs to facilitate solution development. Today service providers are building an eco-system with 3rd party partners to offer new innovative services. The relationship between the application and the end devices shall be decoupled through an abstraction layer, which exposes the sensor's data in comprehensible format and the actuator's command as a service.

- Data and context processing: M2M platforms share many of the key challenges similar to large scale data initiatives, in terms of handling the data streams aggregated from billions of devices, and make them usable by various applications. As huge amounts of data and information are provided to the system, methods must be involved to understand, combine, and process the content aggregated from different sources and in different formats, in order to address the Internet of Content (IoC) challenges. Through this horizontal middleware, proper governance can be realized avoiding the mishandling of data and unsuitable assigning of rights.

- Security and privacy: M2M systems are extremely vulnerable to attacks as they are used in many sensitive sectors in home and industry. Furthermore,

most integrated components are characterized by low power and computation capabilities, and therefore cannot implement complex security mechanisms. Therefore, an M2M platform has to be secured directly from the design. The communication of the devices and the network core should be secured against a large variety of security threats. Authors in [54] categorize the main security vulnerabilities of M2M systems in physical attacks, configuration attacks, protocol attacks and user data and identity privacy attacks. It is essential to construct secure and efficient M2M communication systems against such attacks.

2.4 M2M Traffic Taxonomy

The design of robust and reliable networks and network services is difficult in today's world. Developing a detailed understanding of the traffic characteristics of the network is very important to achieve the goal of reliable services. The current cellular mobile networks are designed for human communication, and therefore are optimized for the traffic characteristics of human-based communication applications, i.e. communication with a certain session length, data volume, interaction frequency and patterns. A study in an operational 3G network with the Busplus application has shown that even a few tens of M2M terminals can lead to a significant network accessibility degradation (up to 60%) [55].

To overcome the shortcomings, improve the network performance and further guarantee the application's QoS, reliable traffic models and reference scenarios are necessary. They could help in better understanding the M2M traffic pattern and characteristics.

2.4.1 Classification of M2M Use Cases

M2M applications can be divided into three main classes according to the traffic generation pattern and its origin:

1. **Event driven applications**: the communication from the M2M node takes place when a specific event occurs and a corresponding notification or report has to be transmitted to inform other nodes about the event. While no event occurs, M2M devices are detached from the network, and remain in idle state. An event may either be caused by a measurement parameter passing a certain threshold or be a command sent by the server controlling a device remotely. Mainly, event-driven applications produce real-time traffic with a variable time pattern and data size in both uplink and downlink direction, for example cardiac arrest in the case of health home monitoring irregular heartbeats or cardiac arrhythmia of a person.

2. **Query driven applications**: in this kind of applications, the M2M node get triggered by a remote M2M server device sending requests to aggregate

collected data or to execute a control command. This traffic is more likely to be uplink-dominant and can either be of constant size as in telemetry, or of variable size like a transmission of an image, or even of data streaming triggered by an alarm. This traffic may be real time or non-real time, depending on the sensor and the event type.

3. **Time driven applications**: they follow the Event driven applications principle, but in this case, M2M devices send data periodically, e.g., every minute. Generally, the traffic of these applications is non-realtime and has a regular time pattern and a constant data size. The transmitting interval might be reconfigured by the server. A typical example of a time-driven message is Smart Meter reading.

2.4.2 M2M Traffic Characteristics

M2M communication provides the opportunity for deploying new services and engaging various kind of connected objects into different domain-specific applications. This leads to new type of traffic profiles and patterns emerging in the future Internet. Analysis of emerging M2M application scenarios such as smart metering, E-health, and Smart transportation has disclosed that in the majority of cases, the M2M traffic includes all or a subset of the following specific features [55]:

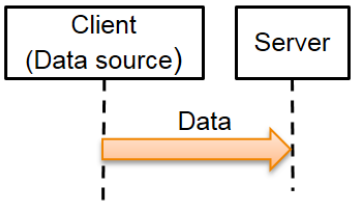
- Short and small number of packets.
- Low duty-cycle packets (i.e. long period between two data transmissions).
- Uplink-dominant packets (i.e. more packets in the uplink than the downlink).
- Real time and non-real time packets.
- Periodic and event-driven packets.
- Raw and aggregated packets (i.e. combining traffic of multiple sources into a single packet, relevant for specific nodes such as gateway).
- Unsynchronized and synchronized packets (i.e. simultaneous access attempts from many devices reacting to the same/similar events).

In each use case the interaction between the M2M nodes, i.e. data source, actuator, server, and data user, could follow four different patters. Figure 2.4a depicts the **telemetric pattern** that includes an object (data source) automatically transmitting measurement of data to the server in a telemetry manner for monitoring proposes. The data could be saved for a specific time or forwarded to a data processing unit for analysis in a time-driven bases.

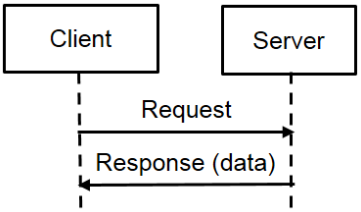
In the query driven applications, the data shall be provided in an inquisitive manner; the request could be generated periodically from the client to inquire data from the server which might forward the request to another node. As illustrated in Figure 2.4b, the **inquiry pattern** is used to inquire data from a node with server

capabilities. However, the availability of the data could not be predicted most of the time, and therefore the response could be empty or returning the same data each time. To avoid repeating requests, a **notification interaction** pattern could be used to report the user the data availability or event occurrence. Figure 2.4c shows the case of data user receiving notifications of an event or status update of another node whenever available.

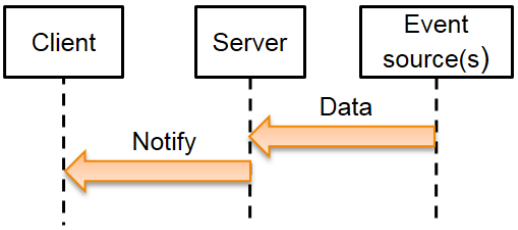
Furthermore, the **instrumental interaction** pattern with actuators, as seen in figure 2.4d, is used to send commands from a decision maker application to one or more actuators. In all these interaction patterns, the communication might be direct



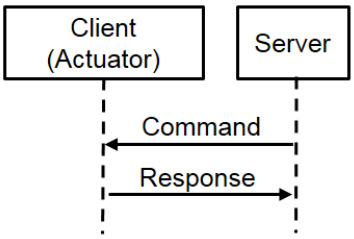
(a) Telemetric Interaction



(b) Inquiry Interaction



(c) Notification Interaction



(d) Instrumental Interaction

Figure 2.4: M2M Interaction Patterns

between two nodes in the same address space, or indirect using an intermediate node (i.e. a gateway) to retarget requests and responses.

Figure 2.5 presents a conceptual classification of M2M traffic classes based on the generated payload size and sampling rate, where the x-axis represents the message size generated by a connected node and the y-axis represents the interval time to send messages from each node. We classify in this conceptual overview four main classes of M2M traffic [40]:

- **C1** - Presents the M2M traffic that is characterized with small messages (few bytes) generated at high rate. Mainly sensors used for monitoring cases are included here.
- **C2** - This class reflects the traffic generated in M2M services that produce small message size with slightly low rate, such as once per second. This is the most expected kind of traffic in M2M systems, e.g., energy smart meter will report the total amount of energy that was used in a household.
- **C3** - Presents traffic of small size message at high rates. This includes the traffic generated in an event base manner or generated by sensors with unreliably connectivity. When the connectivity is lost for some time, some M2M applications have to be adaptable in order to avoid losing the data and compose statistics, e.g., create statistics or store the data locally to send it later. The statistical data volume depends on the level of granularity to be preserved on the statistics and the number and type of sensor data to be considered in the statistics. All the data will have to be transmitted once connection is restored, and their importance might force their immediate transmission when an action should be taken based on the carried information. Even in a case of reliable connectivity, the M2M application might use this approach to have a more energy efficient behaviour by transmitting statistics only.
- **C4** - This class includes traffic generated to be used by M2M nodes that have high level of computation/storage resources, where messages of big size (kilobytes or Megabytes) have to be transmitted to M2M nodes at high rates. Use cases for such class usually have high tolerance to delay where data could be aggregate for hours or days at the M2M node before forwarding it to the server, or it could be firmware updates from the server to a set of connected gateways that will be applied whenever needed.

2.4.3 MTC Traffic Models

Traffic modelling aims to design stochastic processes that match the behaviour of physical quantities of measured data traffic. As discussed in subsection 2.4.1, most of the M2M nodes are expected to have low arrival rate separately, but aggregately could have very high arrival rates. This raises the challenge of defining an accurate traffic model for M2M, whether it is feasible to model the traffic relaying on

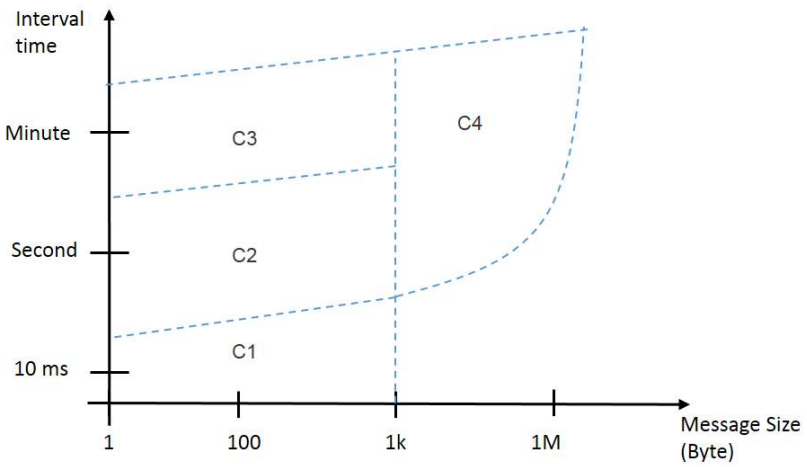


Figure 2.5: Conceptual Classification for M2M Traffic in Terms of Message Size vs Sampling Interval Time

source-traffic model, or it is accurate enough to treat them as one aggregated traffic stream and therefore adapt the aggregated-traffic model. 3GPP has developed two models for MTC traffic generation [56]. The first model is the 3GPP uncorrelated model, which generates perfectly uncorrelated traffic in a specific time interval. The correlation or synchronization between different machines is not taken into account by this model. It assumes every single machine in the system generate traffic independently. Using this model, the expected number of arrivals is based on a normal distribution. The second model is the 3GPP correlated model. This one deals with the correlated or synchronized traffic in a specific time interval. This type of model treats every machine in the system as synchronized. It uses beta distribution to model the aggregated traffic in the simulation period.

In a research paper [57] M2M traffic was classified as an aggregated traffic model, considering that the typical use case includes numerous simple machines assigned to one server. Similar assumptions have been considered in [55, 58, 59] and therefore random Poisson process was adopted to defined the distribution of packet arrivals over a given time period T. Alternatively, authors in [60] have considered different assumptions to develop an analytical traffic model for M2M access networks based on two-class priority queuing system.

2.5 Challenges and Related Technologies

The M2M communication is widely considered as a promising enabler to smart environments. Technically, the M2M communication integrates several complementary technologies, each aiming to provide an essential capability. In Table 2.1, a summary of requirements, challenges and related technologies is highlighted for each M2M functional aspect, listed in Subsection 2.3.3, and the platform architecture

aspect. However, there are many non-technical challenges for rolling-out M2M solutions, among them:

- The conflicts raised by national and international regulations, and their slow evolving process in comparison to the technology evolution.
- The existing business models need to be adapted to support cost effective M2M services. The current charging model based on ARPU (Average Revenue per User) will not be gratified to end users owning or using tens of sensors.
- The exiting roaming policies adapted by access service providers might limit the spread of interworking application between geographically distributed servers, gateways and devices.
- The ubiquitous nature of monitoring and collecting data in M2M could lead to privacy and data security issues, which will bring up concerns about the disclosure of the user's personal information.

Table 2.1: M2M Technologies Aspects and Challenges

Aspect	Requirements	Challenges	Enabling Technologies
Platform Architecture	• Open interfaces • Platforms interoperability • Integrating 3rd party application developers	• Heterogeneity of connected objects • Heterogeneity of access networks and protocols • Converged distributed architecture	• Mediation proxy [45] • Cloud computing [61][62] • Federated architecture [43]
Connectivity	• Multiple protocol support • Mobility support [63] • End-to-end QoS	• Constrained edge platforms and devices • Communication overheads • Traffic heterogeneity • Scalability for addressing and identity [63]	• Short-range wireless technologies • Lightweight and low overhead Protocols [64, 65] • IPv6 • Delay Tolerant Network (DTN) [38]
Device Management	• Ubiquitous and interconnection • Easy-to-deploy • Distributed Searching	• Scalability • Technology heterogeneity • Traffic heterogeneity	• Short distance communication standards • Low-power wireless technologies [66] • Lightweight DM protocols [67]

Application Management	• Compostable algorithms • Large-scale environment • Next generation M2M-based social software	• Service Orchestration • Content delivery	• Service Engineering [68] • API • Ontology reasoning
Data & context management	• User profile management • Device profile management • Context awareness [69] • Searching and data fusion	• Reliability and accuracy of data • Real-time processing • Common sensors ontologies	• Data mining and data fusion [69] • Context Aware Computing [70] • Semantic information modelling [71] • Ontology
Security & privacy	• Confidentiality [72] • Authentication for sensors • Access control • Data Encryption [73]	• Bandwidth efficient authentication mechanisms	• Back-end security solutions (IMSI and IMEI) [54] • Lightweight security methods (e.g., Datagram Transport Layer Security (DTLS) [74])

2.6 Discussion

The M2M communication supports a wide range of potential applications, and therefore gained momentum as an important networking technology. In this chapter the fundamentals of M2M communication were classified, and the challenges of the connectivity aspect were discussed. It was remarked that existing definitions from the literature of the M2M concept are commonly highlighting the limitation of human intervention in this kind of communication. M2M platforms shall enable the seamless flow of data, gathered from connected objects (sensors and actuators) to decision making systems. Such platforms should be able to support new interaction models between connected objects, which are not under human control, produce heterogeneous amount of data and probably have constrained-resources, i.e. they are limited in memory, energy and computation power.

The need to exchange information between different actors in an M2M system motivates the need of an M2M middleware to mediate the information between different stakeholders. Developing a large-scale Smart environment, based on M2M communication, demands interoperability at all communication layers between devices, gateways, and services. However, most existing M2M solutions are not interoperable and have been built in a decoupled vertical fashion, where data gathered by one platform can't be easily reused by other platforms. Generally, M2M networks have the following characteristics:

1. **Dynamically changing topology**: A massive number of devices are foreseen to be existing in the service coverage of each wireless base-station, and concurrent network access attempt from these devices. Some of these connected devices are establishing connections in a frequent manner, operating in an adhoc mode, or requiring mobility support. This will cause frequent changes in the topology.

2. **Different traffic patterns**: The heterogeneity on aggregated data from M2M nodes, in terms of message size and sampling rate, will result on new traffic characteristic dissimilar to conventional H2H traffic.

3. **Heterogeneous QoS requirements**: The wide range of services imposes various levels of QoS requirements, that may require priority-based protocols at different layers.

4. **Heterogeneous device's capabilities**: The magnitude number of connected nodes consists of devices with capabilities, ranging from resource-constrained devices (e.g., simple tags) to resource-rich devices with high computation and storage capabilities.

5. **Autonomous information exchange**: High level of system automation in which the devices and systems can exchange data and make decisions without human intervention.

3 State-of-the-Art

In the previous chapter, a high-level taxonomy of M2M aspects and related technologies was introduced. In this chapter, an elaborated state of the art analysis is presented that covers several research activities on which this dissertation is built

on or relate to the work presented. The analysis of the state-of-the-art starts with the discussion of the standardization work towards an M2M service middleware. We present standardized work from different organizations toward a reference M2M framework. This is followed with a review of a number of existing protocols and access technologies that are commonly adopted in many M2M systems. Finally, an overview of some research projects activities in M2M/IoT context are presented.

3.1 Introduction

The standardization process refers to the developing of technical specifications that aims to maximize compatibility, interoperability, safety, repeatability, and quality. The process shall involve industry, consumers, public authorities and other related parties based on consensus. The need for standardization is highly recognized to remove the technical barriers and support interoperability, scalability and flexibility between connected systems and services in M2M/ IoT environment. In addition, standards help lowering Capital and Operational expenditure (CAPEX and OPEX) for M2M Services, and allow M2M stakeholders to focus on their core business utilizing standardized networking methods without worrying about solving communication challenges on their own. Several contributions to the reliable deployment and standardization of the M2M communication paradigm are coming from the scientific community as well as industry. More than 140 organizations around the world are involved in M2M standardization. The industry has become more active in the standardization process in the M2M domain because of the market demands. Additionally, wireless access standard groups (e.g. IEEE, 3GPP and ETSI) are looking into the impacts to the existing network due to potentially traffic load from M2M devices.

In the following sections, a preview of standardisation activities related to IoT and M2M communication is provided. The preview will include international standard organisations, such as ITU-T, ETSI, oneM2M, OMA, and IEEE. Additionally, an overview of some research projects activities in M2M/IoT context are presented.

3.2 Standardization for Machine-to-Machine (M2M) Service Capabilities

Standardization activities in the M2M field are looking indeed into the direction of horizontal solutions. Thus, they are trying to gather requirements from different vertical domains and provide common reference architectures and protocols that fulfil them. A standardized architecture with a common set of service layer capabilities and open interfaces and APIs should help M2M and IoT service providers to reduce investments, time-to-market, development and on-boarding costs and facilitate management of devices and applications. A lot of activities have been going on recently leading by various standards bodies, Figure 3.1 shows the time-line of Standard activities in this area.

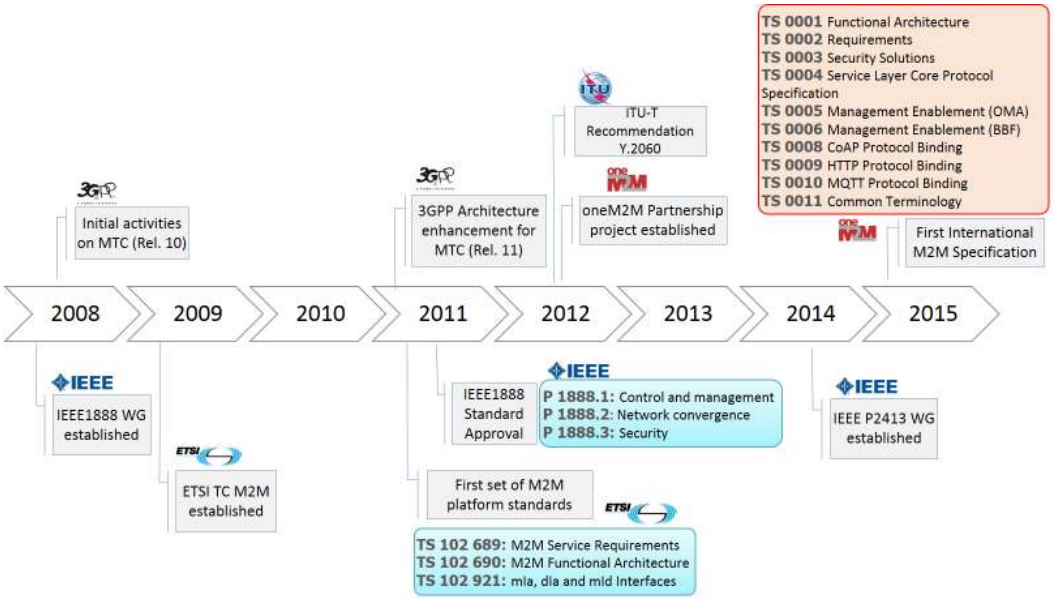


Figure 3.1: M2M Standardization Activities Time-line

3.2.1 International Telecommunication Union (ITU)

ITU-T leads the work of the ITU on standards for NGN and future networks proposing a high level architecture. The standardization efforts in ITU are being addressed under various banners like 'Internet of Things (IoT)' [15], 'Machine Oriented Communication (MOC)' [75], and 'Object-to-Object Communication' [76].

In 2011, the working structure of the IoT-GSI (IoT Global Standards Initiative) [77] was formally established, as the centrality of IoT in the evolution of future network and service infrastructures is widely recognized. Afterward, the ITU-T activities related to IoT have greatly expanded and produced additional Recommendations for general IoT framework, transversal aspects, and various areas of application domains. The ITU-T Recommendation Y.2060 [15] was finalized in June 2012, providing a definition of the IoT that has obtained large acceptance within the IoT community. Additionally, the document specified a Reference Model for IoT consisting of four layers: 1) application layer ; 2) service support and application support layer; 3) network layer; 4) device layer, as well as management capabilities and security capabilities which are associated with the four layers, as depicted in Figure 3.2. The document described some capabilities to be supported in each layer in a high-level perspective.

The Focus Group on M2M Service Layer (FG M2M) [78] was established in 2012 with the key goal to study requirements and specifications for a common M2M Service Layer. It has focused its developments, from the point of view of use cases and derived requirements for the common M2M service layer, on the "e-health" application domain (specifically, on remote patient monitoring and assisted living services).

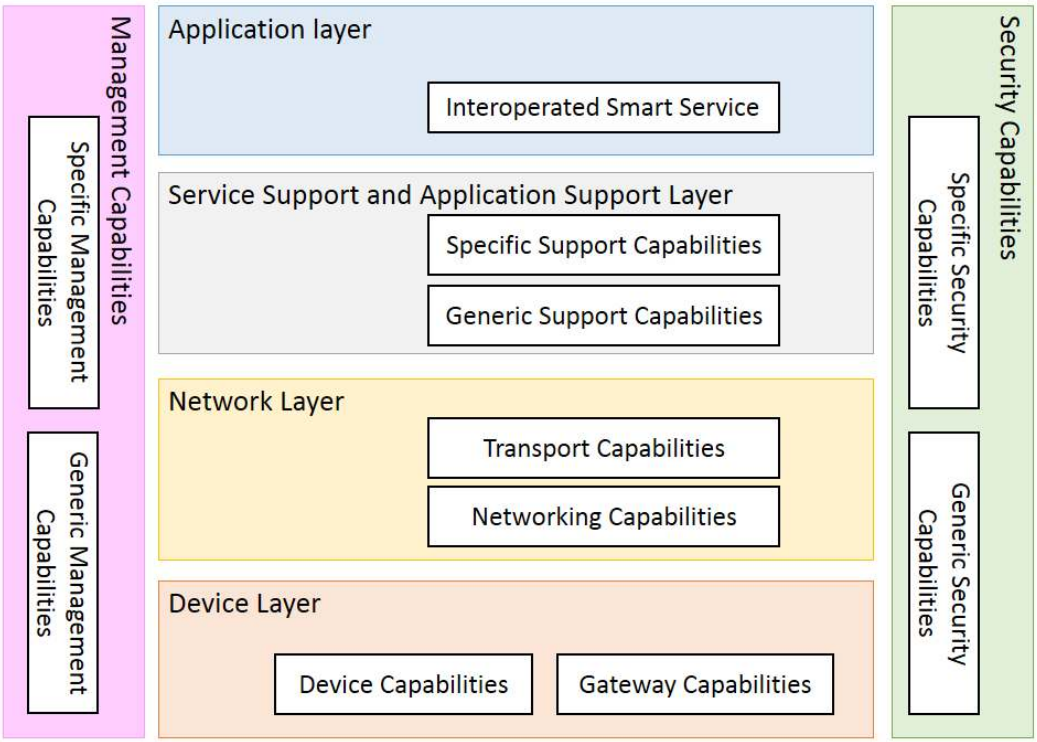


Figure 3.2: ITU-T Reference Model for IoT, based on [15]

The FG M2M, who had targeted the inclusion of vertical market stakeholders not part of the traditional ITU-T membership, such as the World Health Organization (WHO), and the collaboration with M2M and e-health communities and SDOs, has actually liaised with various SDOs, fora and consortia, including for the completion of an e-health standards repository. In the context of the FG M2M service layer work, in line with the IoT Reference Model described in ITU-T Y.2060 [15], the M2M service layer capabilities aim to include those common to the support of different application domains as well as the specific ones required for the support of each application domain. It should be noted, in this perspective, that the M2M communication capabilities are seen as an essential enabler of the IoT.

Recently, the ITU-T Study Group 20 (SG20) on IoT and its applications including Smart Cities and communities (SC&C) was established. The group is responsible for international standards to enable the coordinated development of IoT technologies, including M2M communications and ubiquitous sensor networks.

3.2.2 ETSI M2M Reference Architecture

The European Telecommunications Standards Institute (ETSI) (http://www.etsi.org), produces globally applicable standards for Information and Communication Technology (ICT), including fixed, mobile, radio, converged and Internet technologies.

This non-profit organization is officially recognized by the European Union as a European Standards Organization, granting an international reach.

Aiming at an efficient end-to-end delivery of the M2M services, ETSI has defined a set of requirements [50]. These requirements address main features related to security and communication management as well as the functional requirements for a horizontal middleware oriented towards M2M communication in which the communication with various sensors and actuators is executed in a convergent and consistent manner for multiple applications. ETSI goal is to define a middleware Service Capability Layer (SCL) that interact with M2M nodes over open interfaces named: mIa, dIa and mId. These interfaces offer generic and extendable mechanism for interactions with the SCLs at both device and gateway domain (DSCL/GSCL) and network domain (NSCL). Table 3.1 provides a description of the service capabilities defined by ETSI M2M in [50].

As depicted in Figure 3.3, the ETSI M2M reference architecture consists of three parts:

1. M2M Area Network: That includes heterogeneous endpoint devices, such as sensors and actuators, connected through an access network e.g., ZigBee, M-BUS, or Bluetooth. This part of the network ends with an M2M gateway that hides the complexity of the area network from the rest of the M2M nodes. The gateway provides a set of service capabilities to M2M applications in this domain, including the (Device/Gateway/ Network) Generic Communication (xGC) capability to handle transport and session management functionalities and the (Device/Gateway/ Network) Reachability, Addressing and Repository (xRAR) capability for data storage.

2. M2M Middleware Core: The M2M core implements functionality to facilitate the communication between devices (in the M2M area network) and the network applications. The M2M core provides several features such as device management, reachability, and generic communication mechanisms over the communication network. Additionally the M2M core handles the data exchange between devices and applications. On one hand, it aggregates the data received from the device, and forwards it to applications that show interest of that data by means of subscribing to its resource. On the other hand, it orchestrates the actuation commands or parameter updates received from applications and transferred to devices, depending on the urgency of the communication and on the momentary network conditions, as well as on the parameters of the device.

3. Application Domain: As the M2M middleware allows the connection of heterogeneous devices, an application is needed to execute the logic of different use cases such as energy, automotive, health, transportation etc. The main function of any M2M application is to control the data acquisition from surrounded environment, perform some calculations on them prior to decision making, and finally send commands to act according to that decision. Each

M2M node compatible with ETSI M2M specification include an application i.e. Device application (DA), Gateway application (GA) and Network application (NA).

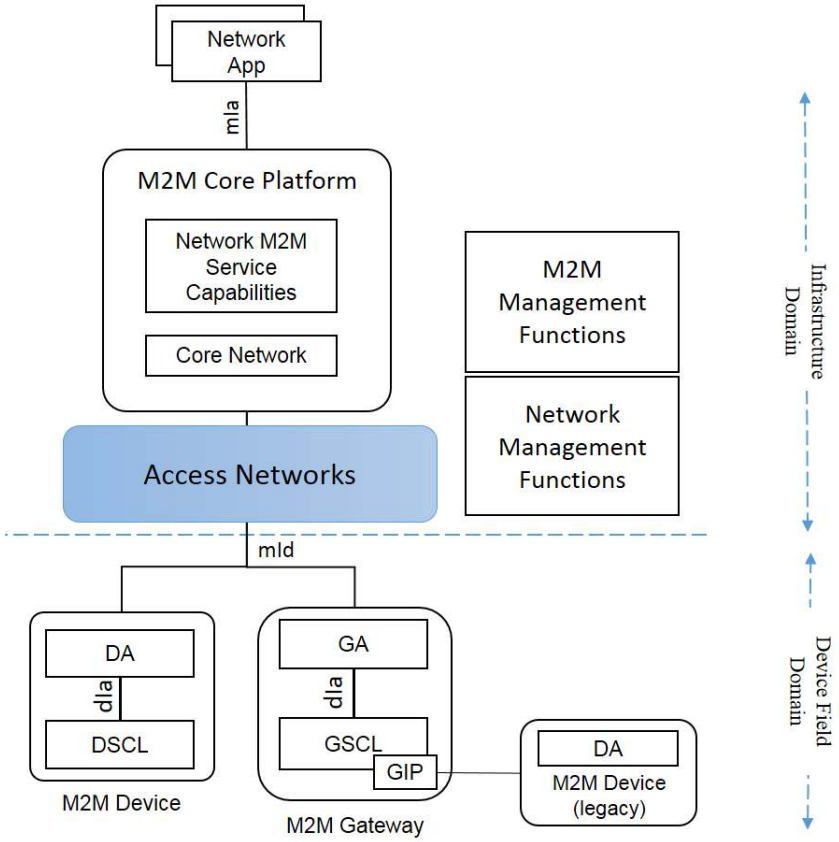


Figure 3.3: ETSI Functional Architecture of M2M Systems, adapted from [50]

3.2.3 OneM2M Partnership Project

In 2012, the oneM2M consortium was established with the aim of consolidating the standardization work in M2M communication [79]. oneM2M is a consortium of seven standards development bodies working in the M2M communication standardization. More than 260 participating partners and members joined oneM2M to participate in the standardization of M2M communication system, including ETSI and OMA. The participating organizations intend to transfer all standardization activities in the scope of M2M service layer to the oneM2M. OneM2M specifies a high-level architecture at both the field and infrastructure domain to support end-to-end M2M services, as illustrated in Figure 3.4. The oneM2M functional architecture comprises of the following entities:

Table 3.1: Description of ETSI M2M Services Capabilities

ETSI Capability	Description	NSCL	GSCL	DSCL
(Device/Gateway/ Network) Application Enablement (xAE)	Handles registration of M2M nodes and allows routing towards different capabilities. Also generates charging records pertaining to the use of capabilities.	✓	✓	✓
(Device/Gateway/ Network) Generic Communication (xGC)	Provides Communication management functionality for SCLs and applications. Also provides transport session establishment and teardown along with security key negotiation.	✓	✓	✓
(Device/Gateway/ Network) Reachability, Addressing and Repository (xRAR)	Store application and xSCL registration information as well as aggregated data and make it available, on request or based on subscriptions, subject to access rights and permissions.	✓	✓	✓
(Device/Gateway/ Network) Communication Selection (xCS)	Provides network selection based on policies, when the M2M device or gateway can be reached through several networks or several bearers. Also, provides alternative Network or Communication Service selection after a communication failure using a first selected Network or Communication Service.	✓	✓	✓
(Device/Gateway/ Network) Remote Entity Management (xREM)	Provides Configuration Management functions, which is the means to provision a set of Management Objects in an M2M Device, an M2M Gateway, a set of M2M Devices or a set of M2M Gateways.	✓	✓	✓
(Device/Gateway/ Network) Security (xSEC)	Supports a set of security functionalities and M2M service bootstrap.	✓	✓	✓
(Device/Gateway/ Network) Interworking Proxy (xIP)	Optional capability to provide interworking between non ETSI compliant devices or gateways and the SCL.	✓	✓	✓
Network Telco Operator Exposure (NTOE)	Interworking and using of Core Network services exposed by the Network Operator.	✓		
(Device/Gateway/ Network) History and Data Retention (xHDR)	Optional capability for storing records pertaining to the usage of the M2M SCs.	✓	✓	✓
(Device/Gateway/ Network) Transaction Management (xTM)	optional capability to manages transactions	✓	✓	✓
(Device/Gateway/ Network) Compensation Broker (xCB)	Optional capability to manages compensation transactions on behalf of applications.	✓	✓	✓

1. Application Entity (AE): responsible of providing end-to-end M2M logic solution, i.e. E-health, Logistic, Smart Energy, etc.

2. Common Services Entity (CSE): comprises a set of Common Service Function (CSF) that are common to the M2M environments and exposed to other entities through four reference points that consist of one or more interfaces. OneM2M specified 12 different CSFs, some of them can be optionally implemented at a given CSE depending on the implementation domain and device, supported networks, etc. An CSE could be implemented on different kind of nodes such as middle node (i.e. M2M gateways) at the field domain, or infrastructure node (i.e. M2M Server Infrastructure) at the infrastructure domain.

3. Underlying Network Services Entity (NSE): to provide services to the CSEs, such as device management, location services and device triggering.

Each M2M node could deploy one or more entities. OneM2M specification describes several types of nodes that could be mapped to physical objects in an M2M system. Table 3.2 symmetrized the oneM2M defined nodes and the consisting entity of each.

Table 3.2: Description of oneM2M Nodes

Node	Physically	AE	CSE	NSE
Application Service Node (ASN)	Capable M2M Device	1 or more	✓	✓
Application Detected Node (ADN)	Constrained M2M Device	1 or more	✗	✓
Middle Node (MN)	M2M Gateway	1 or more	✓	✓
Infrastructure Node (IN)	M2M Server	1 or more	✓	✓

OneM2M is specifying four reference points supported by the CSEs, namely: Mca, Mcn, Mcc and Mcc'. The Mc- nomenclature is based on the mnemonic "M2M Communication", while a, n and c stand to Application Entity (AE), Network Service Entity (NSE) and CSF respectively [80]:

1. Mca reference point: for interaction communication between an AE and CSEs, that enable the AE to use the exposed services from the CSE.

2. Mcn reference point: to allow the CSE to use services provided by the underlying NSEs.

3. Mcc reference point: to enable the interworking between CSEs. Any CSE could use some functionality provided by another CSE in order to provide service to other entities.

4. Mcc' reference point: The Mcc' shall be implemented on CSEs at infrastructure nodes to enable inter-domain communication between CSEs at different service provider domains.

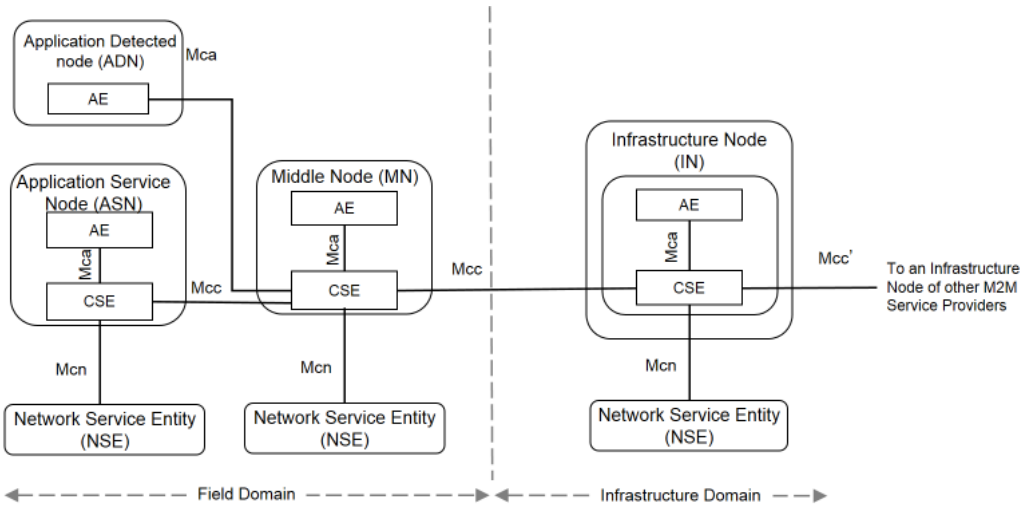


Figure 3.4: OneM2M Functional Architecture of M2M Systems, adapted from [80]

Generally, the Scope of oneM2M work includes:

- Access independent view of end-to-end services.
- Open standard interfaces, APIs and protocols.
- Security, privacy, and charging aspects.
- Reachability and discovery of applications.
- Interoperability, including test and conformance specifications.
- Identification and naming of devices and applications.
- Management aspects (including remote management of entities).

3.2.4 Open Mobile Alliance (OMA)

OMA has approved and released the final version of OMA Next Generation Service Interfaces (NGSI) in May 2012 [81], which focuses on creating a set of open APIs to enable next generation services. The scope of NGSI includes the standardization of six architectural areas, each include a set of functional APIs. Figure 3.5 depicts the relation of NGSI interfaces and functional areas. The functional areas for these interfaces are the following:

Table 3.3: Description of OneM2M Common Service Functions

OneM2M CSF	Description
Application and Service Layer Management (ASM)	Provides functions to manage the AEs and CSEs on the ADNs, ASNs, MNs and INs. This includes capabilities to configure, troubleshoot and upgrade the functions of the CSE, as well as to upgrade the AEs.
Communication Management and Delivery Handling (CMDH)	Responsible of handling the communications with other CSEs, AEs and NSEs. These include buffering communication requests and selecting connection for data deliver.
Data Management and Repository (DMR)	Responsible for providing data storage and mediation functions, such as converting data into a specified format, and storing it for analytic and semantic processing.
Device Management (DMG)	Provides management of device capabilities on MNs (e.g. M2M Gateways), ASNs and ADNs (e.g. M2M Devices), as well as devices that reside within an M2M Area Network.
Discovery (DIS)	Uses the Originator provided filter criteria (e.g. a combination of keywords, identifiers, location and semantic information) in searching for information about applications and services. The result of a discovery request is subject to access control policy allowed by M2M Service Subscription.
Group Management (GMG)	Responsible for handling group related requests.
Location (LOC)	Handles AE location requests to obtain geographical location information of Nodes (e.g. ASN, MN) for location-based services.
Network Service Exposure, Service Execution and Triggering (NSSE)	Manages communications with the Underlying Networks for accessing network service functions over the Mcn reference point.
Registration (REG)	Handles registration requests from an AE/CSE to register with a Registrar CSE in order to allow the use the services offered by the Registrar CSE.
Security (SEC)	Comprises several security functionalities for: Sensitive data handling; Security administration; Security association establishment; Access control including identification, authentication and authorization; and Identity management.
Service Charging and Accounting (SCA)	Provides charging functions for the Service Layer, by supports different charging models which also include online real time credit control.
Subscription and Notification (SUB)	Manages subscriptions to resources, subject to access control policies, and sends corresponding notifications to the address(es) where the resource subscribers want to receive them.

- Data Configuration and Management: Responsible for creating, reading, updating and deleting data of Extensible Markup Language (XML) or non-XML type. It also provides a subscribe/notify mechanism for the managed data.
- Call Control and Configuration: Offers methods for Call setup, handling and event notifications. Call conferencing control is also supported.
- Multimedia List Handling: Management of Lists of media identifiers (e.g. URIs), being used by a streaming functionality.
- Context Management: Management of Context Entities by identifiers, attributes with corresponding values and meta data. It also exposes an interface for access Context Information, following push and pull models.
- Service Registration and Discovery: Is a service dictionary, which supports registration of services and allows to lookup services.
- Identity Control: Allows for the management (creation, modification, deletion) of identities and related identifiers and provides an interface for retrieving identifiers for an identity through another identifier.

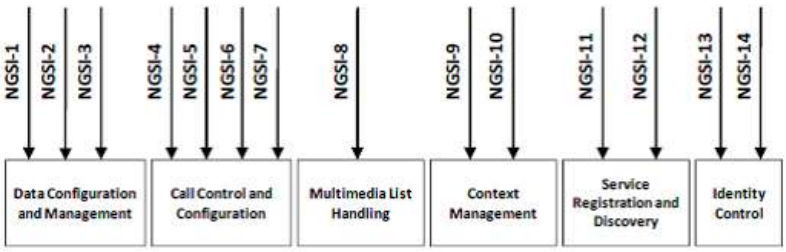


Figure 3.5: OMA NGSI Architectural

Several OMA standards map into the ETSI M2M framework, both standardization bodies work in order to provide associations between ETSI M2M Service Capabilities and OMA Supporting Enablers. M2M Networks connect sensors and actuators as well, thus device management protocols are essential here. Although device management components are present in both ETSI and oneM2M technical architecture, the device management protocol that is supposed to be used by the components is left out of scope. Different options are usable for controlling devices, such as SMS-based protocols from the legacy systems. The OMA is providing platform-independent Device Management (DM) protocol for general devices [82]. The DM protocol defines an interface between the DM Server and the DM Client to manage and configure devices on top of HTTP transport protocol. Recently, OMA introduced the LightweightM2M (LWM2M) DM [67], a device management protocol matching the constraint requirement for M2M domain by using CoAP [64]

as transport protocol. The payload can be encoded as plain text, JavaScript Object Notation (JSON) object or Type-Length-Value (TLV) encapsulated, making the encoding very efficient. Security can be achieved by using DTLS.

3.2.5 Institute of Electrical and Electronics Engineers (IEEE)

The IEEE-Standards Association (IEEE-SA) is an organization within IEEE that develops global standards in a broad range of industries, including: power and energy, biomedical and health care, information technology, telecommunication, nanotechnology, and many more. In its research into IoT, it has identified over 140 standards and projects, a list is available at [83].

The IEEE 1888 Standard for Ubiquitous Green Community Control Network is among the ongoing standardization activities for the IoT within IEEE-SA, it defines a data exchange protocol that generalizes and interconnects M2M components (gateways, storage, application units) over the IPv4/v6-based networks [84]. The IEEE 1888 standard is applied on some Smart Energy projects in Japan and Thailand [85, 86]. The architecture of this standard includes gateways, storage, applications and registry component, which are based on a TCP/IP facility network as shown in Figure 3.6 [87] gateways connect field-bus sensor-actuator networks and TCP/IP-based network. A central storage collects all data sequences from all other components. Applications can be designed to display sensor readings and input actuator commands. And a common registry is a broker for managing all the components. There are two types of communication protocols. Firstly, the component-to-component communication protocol consists of: WRITE protocol to send data towards remote components, FETCH protocol to read data from remote components, and TRAP protocol to notify/update data to remote components. Secondly, the component-to-registry communication protocol comprises of: REGISTRATION protocol to register active components, and LOOKUP protocol to search for components. All IEEE1888 communications use the protocol message structure of Simple Object Access Protocol (SOAP). SOAP is method for exchanging XML based messages over the Internet for providing and consuming web services. SOAP message are transferred forming the SOAP-Envelope.

A recent activity launched in 2014 is the IEEE P2413 [88] intending to consider a very broad range of verticals and stakeholder groups, and develop a standard architectural framework for the IoT. The initial objective of the IEEE P2413 group is to create a standard interoperability architecture and define commonly understood data objects, for information sharing across IoT systems. The standard is targeted by 2016; there will also be cooperation with other standard bodies' efforts in the IoT area, including ETSI, the ISO and oneM2M. IEEE P2413 is currently considering the architecture of IoT as three-tiered, with the layers: Applications, Networking and Data Communication, and Sensing. The goals for the IEEE P2413 group are to [89]:

1. Accelerate the growth of the IoT market by enabling cross-domain interaction

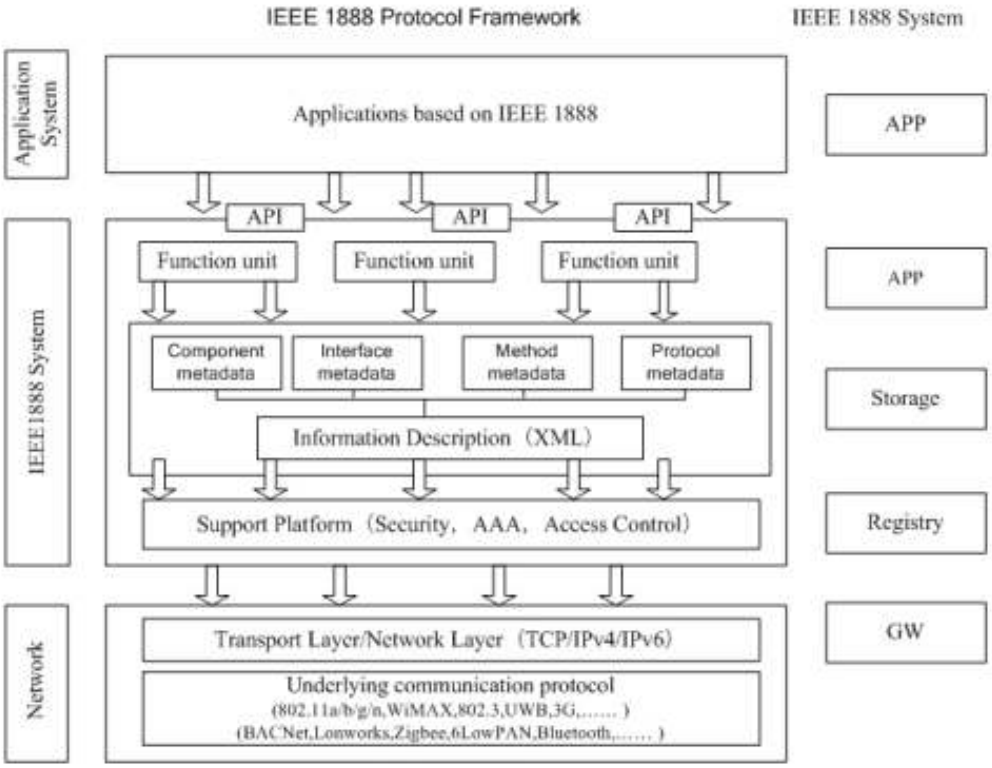


Figure 3.6: The IEEE1888 Architecture [87]

and platform unification through increased system compatibility, interoperability and functional exchangeability.

2. Define an IoT architecture framework that covers the architectural needs of the various IoT application domains.

3. Increase the transparency of system architectures to support system benchmarking, safety and security assessments.

4. Reduce industry fragmentation and create a critical mass of multi-stakeholder activities around the world.

5. Leverage the existing body of work.

3.2.6 Discussion

There is good momentum on M2M standardization efforts, which aim to achieve interoperability and compatibility in M2M systems independently of the vertical market solutions. Several standardization efforts related to M2M and IoT have been carried out and have contributed to the current state of the art of this area.

One direction of these efforts have been concentrating on building a general reference model architecture, another direction has been focusing on specific technologies of the M2M in order to make it of practical value.

The initial work of ITU-T was following the first direction; the Reference Architecture defined on ITU-T Y.2060 [15] has provided a common ground for the field of IoT, that was further extended by other committees such as IERC [90]. The model defines high-level layers and their basic capabilities and relationships with each others.

The other standardization direction, focusing on specific technologies, is mainly presented by the output from ETSI, oneM2M, OMA and IEEE. Each has aimed to develop an application-agnostic M2M framework for the vertical market solutions, with emphasis on specific technologies. It is worth mentioning that, oneM2M has a much global view as it aims to unify the Global M2M Community, by enabling the federation and interoperability of M2M systems across multiple networks and topologies.

In [32], we analyzed the specified capabilities and functionalities by ETSI, oneM2M and OMA, trying to answer the question of how these standard specifications had addressed the requirements towards realizing a reliable and secure architectural framework for M2M services. Table 3.4 summarizes the specified M2M functionalities by ETSI M2M, oneM2M, OMA and IEEE1888.

As presented above, ETSI defines an end-to-end architecture where several devices connected directly or via a gateway to a central backend server in the network side. In this context, the ETSI M2M architecture represents a star topology with

Table 3.4: Specified Functions by M2M Standards

Capability	Functionality	ETSI	oneM2M	OMA NGSI	IEEE 1888
Connectivity	Communication selection	✓	✓	✗	✗
	Session management	✓	✓	✓	✓
Device Management	Location	✗	✓	✓	✗
	Device triggering	✗	✓	✓	✗
	Device management	✗	✓	✓	✗
Application Management	Software Management	✓	✓	✗	✓
	Configuration function	✓	✓	✓	✓
	Registration and Charging	✓	✓	✓	✓
Data Processing	Discovery	✓	✓	✓	✓
	Subscription and Notification	✓	✓	✓	✓
	Resource grouping	✗	✓	✗	✗
	Semantic processing	✗	✓	✗	✗
Security	Authentication	✓	✓	✗	✓
	Encryption	✗	✓	✗	✗
	Integrity verification	✓	✓	✗	✓

a set of distributed Service Capability Layers (xSCL), where the x stands for N (Network), G (Gateway), or D (Device). The oneM2M architecture is almost similar to ETSI M2M architecture, however oneM2M defines an open interface between interconnected CSEs to represent a mesh topology. While IEEE 1888 adapted a bus topology connecting different component i.e. storage, register and gateways.

In relation to the connectivity control aspect, each SCL in the ETSI M2M architecture shall include the xGC capability, which is responsible for the established transport session including encryption and reporting errors features. Similarly, the CMDH CSF from oneM2M handles the communication functionalities with other entities i.e. CSEs, AEs and NSEs. The CMDH CSF uses the Underlying Network equivalent delivery handling functionality based on provisioned policies to decide when to use the communication channel to transmit the data. For managing access of alternative networks, a communication service selection function is described by both ETSI xCS capability and oneM2M NSSE CSF. OneM2M protocol working group specified mapping the standard APIs to more underlying transport protocols to meet the requirements of various use cases. In addition to HTTP, which is the de facto Internet transport protocols, specifications are defined to map CoAP/UDP and MQTT/TCP as well to support integration of constrained devices in the IoT.

Considering the device management aspect, OMA provides standard mechanisms and protocols for device management in wire and wireless areas, which has been mapped by other standards bodies to their specified functionalities. For example, the ETSI M2M committee has specified three OMA Device Management (DM) compliant Management Objects (MO) [91]. The configuration of MOs is provided by the xREM capability. The specification suggests to use an M2M specific data model, which should be based on OMA-DM and TR-069 data models. The model is used to describe a management object resource, which holds the management data and provides a certain type of M2M remote entity management function. oneM2M architecture focuses on the services provided by the underlying network entity to the CSEs over the Mcn reference point, these services include: location management, device management and device triggering. The Location capability from oneM2M specifying three ways of obtaining location information: a location server in the underlying network; a GPS module in an M2M device; or by information inferring location stored in other nodes.

On application management aspect, the ETSI SCL handles resources associated to the system's entities following the RESTful paradigm. Applications at different nodes rely on the SCLs to interchange data between each other, monitor other applications, or control devices. OneM2M supports the Hypermedia as the Engine of Application State (HATEOAS) with REST to enhance service discoverability and extensibility in the future. oneM2M specified the Application and Service Layer Management (ASM) function for handling software configuration, execution, troubleshooting and upgrading at AEs and CSEs by utilizing the Device Management (DMG) functions.

ETSI specifications have focused on the hierarchical representation of M2M resources as well as on standard APIs for accessing them by the CRUD (Create, Re-

trieve, Update and Delete) verbs. The xRAR capability is the cornerstone on ETSI M2M platforms for data handling. It is responsible for data storage and exchange between applications and SCLs. This capability includes also the subscription/notification mechanism, which enables applications to receive event notifications from gateways; also supports information searching based on defined criteria. Similarly, oneM2M specified the Data Management and Repository (DMR) CSF for data storage and mediation functions, the Discovery (DIS) CSF for information searching, and the Group Management (GMG) CSF to enable the M2M System to perform bulk operations on multiple devices, applications or resources that are part of a group. The resource tree at both ETSI and oneM2M specifications have a lot of similarity, however a number of differences could be listed here:

1. oneM2M has defined a set of new resources to handle the additional capabilities and functionalities specified by the CSFs, such as LocationPolicy, StatesCollect, are Request resources.

2. The resource's URL in oneM2M are shorter by omitting the usage of collections resource (i.e., the applications, containers and contentInstances resources).

3. In addition to the parent-child relation between resources, oneM2M specified linking resources in a non-hierarchical method.

4. Additional attributes are defined to application, container and contentInstance resources, for example, the ontologyRef attribute is used to link the resource to a predefined ontology.

Considering the Security aspect, ETSI TC M2M addressed security needs of M2M service providers by specifying the infrastructure protection at the network layer. The ETSI security capability supports M2M service bootstrap and key hierarchy realization for authentication and authorization. oneM2M leverage the security capabilities to provide security services for M2M applications, including: Sensitive data handling, credentials deployments and management, secure connection establishment and management, authorization and access control that supports roles and context attributes, and support of dynamic configurations involving a Centralized Key Distribution.

3.3 Standardization for Transport and Application layer

The current M2M related technologies landscape are highly fragmented. As Figure 3.7 illustrates, the protocol stack for the M2M communication includes various standardized protocols at transport, network and data link layers. In [92], a survey of a standardized protocol stack for IoT is presented, focusing on power-efficient wireless communication. Typically, M2M applications utilize IP-compatible open protocols that are standardized, in order to be widely deployable.

Recently, various protocols have been proposed to the IoT development, aiming to address the requirements of integrating resource-constrained devices and supporting ubiquitous access. In the following subsections we overview some of the protocols for M2M.

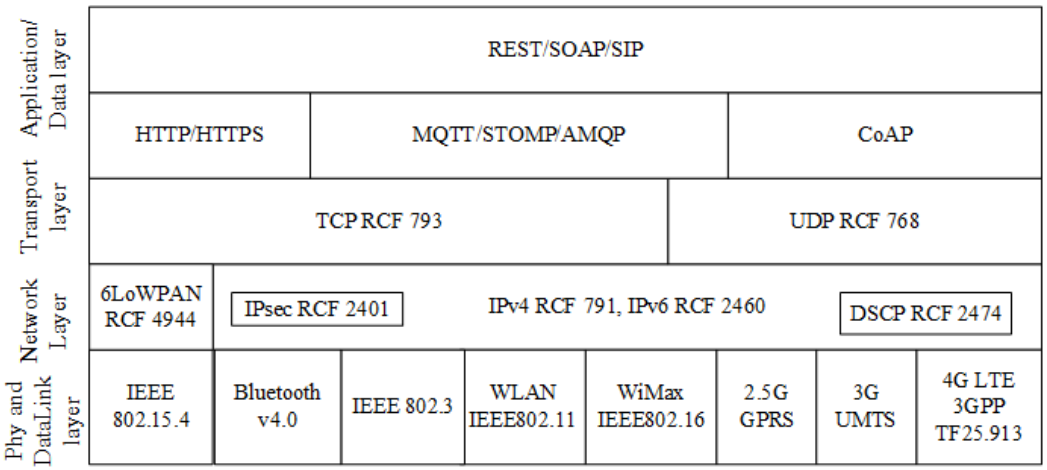


Figure 3.7: Heterogeneity of Protocol Stack in M2M Communication

3.3.1 Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol (HTTP) is an application layer protocol designed within the framework of the IP suite. Version 1.0 of HTTP was published in 1996 as RFC 1945 [93]. The protocol is well-tried and powerful, but it's relatively expensive both in implementation code space and network resource usage. The defined specification presumes an underlying and reliable transport layer protocol, for this TCP/IP is most commonly used. However, it can use unreliable protocols such as the User Datagram Protocol (UDP) over port number 80 [94]. HTTP is request-driven protocol where clients (represented by a user-agent) open a connection to a server and send their request. The server processes this request and returns the resource requested. Besides retrieving information, HTTP also offers methods (called verbs) to Create, Retrieve, Update and Delete information. HTTP is inherently stateless which allows it to scale horizontally.

Although HTTP is widely supported on different kind of computing devices (i.e., smartphones and tablets), it might not be suited for all M2M nodes due to its resource-demanding nature. Typically, a TCP connection is expensive to create, while most HTTP 1.0 or older connections use Transmission Control Protocol (TCP) at least efficiency, which leads to congestion and unwanted overhead. The problem becomes serious when it happens in the scenario of M2M bursty traffic with a large amount of devices interconnected. To improve this, HTTP/1.1 provides connection reuse mechanism by using Keep-Alive in general headers. This mechanism and other improvements have been specified in HTTP/1.1 [95], and thus all connections are

set to persistency by default. This mechanism aims for reducing CPU and memory usage, pipelining, reducing network congestion, reducing latency the frequency of TCP opening handshakes and error reported improvement.

Recently, the HTTP Working Group has published the specification of HTTP/2 in RFC 7540, which aims to enable a more efficient use of network resources and a reduced perception of latency by introducing header field compression and allowing multiple concurrent exchanges on the same connection [96].

3.3.2 Constrained Application Protocol (CoAP)

In 2010, the Internet Engineering Task Force (IETF) Constrained RESTful Environments (CoRE) group was founded specifically to work on the standardization of a framework for resource-oriented applications, allowing realization of RESTful embedded web services in a similar way as traditional web services, but suitable for the most constrained nodes and networks. Their work resulted in the Constrained Application Protocol (CoAP), a specialized RESTful web transfer protocol for use with constrained networks and nodes.

The CoAP was proposed by the IETF CoRE working group, and recently confirmed (RFC 7252) [64], to support constrained devices (i.e., with low computational/memory capabilities) and networks, such as those expected to form the IoT. To facilitate the implementation on similar devices, a number of design choices have been considered including: i) the use of UDP as the transport layer protocol to avoid the overhead of connection oriented protocols; ii) efficient packing of protocol information in a binary base header, which can be as small as 4 bytes. CoAP is a lightweight client/server application protocol, which supports the REST paradigm, through a request/response model using four request types: GET (i.e., retrieve the content of the resource), POST (i.e., create a new resource), PUT (i.e., update the content of an existing resource), DELETE (i.e., remove a resource). Similar to HTTP, CoAP is a stateless protocol that identify resources through URIs, e.g., *coap://coap-server.com/resource/name*. The main conceptual difference between CoAP and HTTP is a message abstraction that determines the type of request or response.

- A confirmable message request is sent when client is supposed to get a response or delivery confirmation. The response can be an acknowledgement message or non-confirmable message or both of them.
- A non-confirmable message request is sent when a client does not expect a request confirmation.
- An acknowledgement message is sent as a response to confirm that a request was delivered. It may contain additional data or be empty.
- A reset message is sent as a response to a confirmable or not confirmable node to notify that a request was received but some data was missed. Usually it happens when the server node was restarted.

3.3.3 Message Queue Telemetry Transport (MQTT)

MQTT [65] is an open protocol specified by IBM and Eurotech, and recently it has been moved into the open source community, and used by the Eclipse foundation in M2M applications. The protocol was designed to be simple and lightweight in order to be suitable for constrained devices. It features a lightweight header size of 2 bytes and reduced clients footprint. The MQTT protocol adapts the Publish/Subscribe (Pub/Sub) module and uses long-lived outgoing TCP connections to a broker node, as illustrated in Figure 3.8. The MQTT architecture includes the publishers, broker server and the subscribers. The broker receives the subscription requests from the clients on the topics they are interested in, at the same time it receives message from publishers and forward them to the subscribers. MQTT ensures reliability by providing the option of three QoS levels:

1. QoS-0 (Fire and forget): A message is sent once and no acknowledgement is required. Thus, the message is delivered according to the best efforts of the underlying network. No response is sent back by the receiver/subscriber and no retry is performed by the sender/publisher.

2. QoS-1 (Delivered at least once): A message is sent at least once and an acknowledgement is required. It ensures the arrived of the message at the receiver/subscriber at least once. A QoS 1 PUBLISH Packet has a Packet Identifier in its variable header and is acknowledged by a PUBACK Packet.

3. QoS-2 (Delivered exactly once): The highest QoS offered by MQTT, a four-way handshake mechanism is used to ensure the message is delivered exactly one time. It comes at a price of higher overhead and traffic in the network.

MQTT supports any type of data (text, binary, JSON, XML, BSON), with a maximum size of 256 MByte. MQTT is also used by Facebook in implementing a

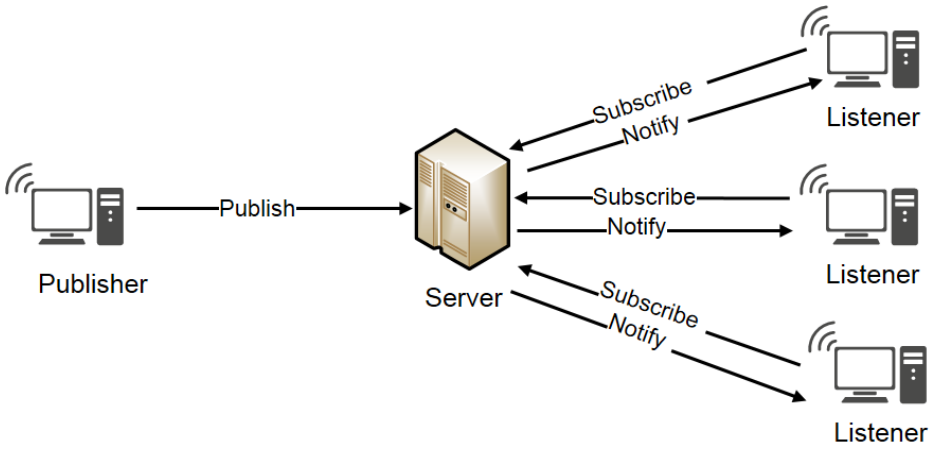


Figure 3.8: The General MQTT Model

fast lightweight asynchronous messaging protocol. Security is handled with the use of the TLS/SSL protocols over TCP, similar to HTTP.

3.3.4 Advanced Message Queuing Protocol (AMQP)

Similar to MQTT, AMQP is a binary open protocol that provides a richer set of messaging scenarios. AMQP comes from the finance community, designed to efficiently support a wide variety of messaging applications and communication patterns. The originators wanted an open way to communicate the increasing over-the-counter trace, risk and clearing market data they transfer, without handling licensing issues of off-the-shelf protocols. The final version 1.0 specification of AMQP was released in 2012 [97].

In AMQP, the messages are self-contained and data content in a message is opaque and immutable [98]. According to the specifications, there is no limits for the message's size, it can either support a message of 1 GByte or just few bytes in size. Several possibilities for message delivering are possible, such as point-to-point, store-and-forward or publish- and-subscribe. For instance, when a message is sent to an AMQP broker, actually it is sent to a queue, and after it is delivered to all subscribed customers to this queue as a push notification, as illustrated in Figure 3.9. AMQP supports different acknowledgment uses cases and transactions across message queues; that allows separation of the different transactional semantics. AMQP ensures reliability with the following message-delivery guarantees:

1. At most once: means that a message is sent once either if it is delivered or not.
2. At least once: means that a message will be definitely delivered one time, possibly more.
3. Exactly once: means that a message will be delivered only one time.

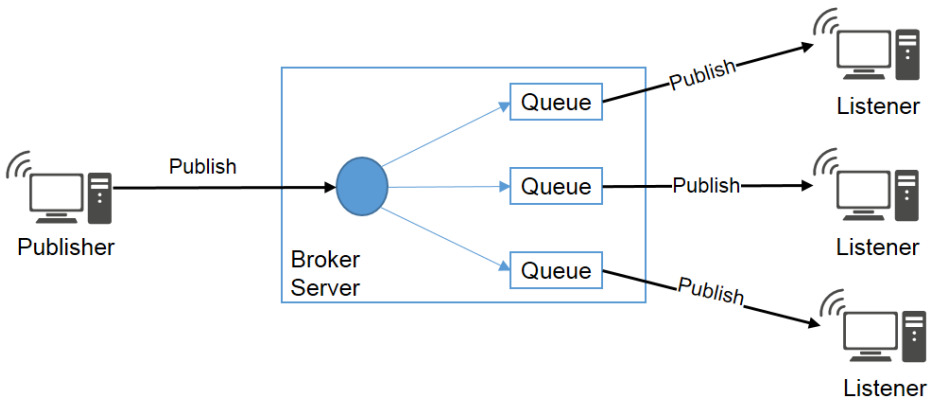


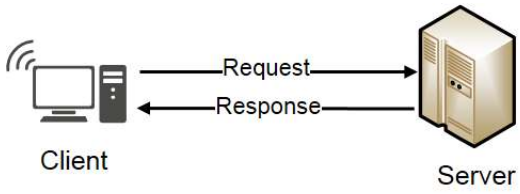
Figure 3.9: AMQP Protocol Model based on AMQP-V1.0

3.3.5 Discussion

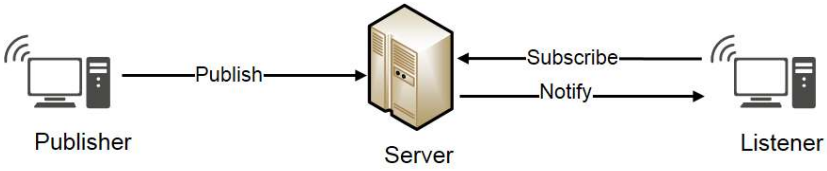
The previously described protocols could be categorized under two main communication paradigms:

- The Request/Response (Req/Res) model, commonly used in distributed system to exchange information through message passing between a sender and a receiver, as illustrated in Figure 3.10a. The Req/Res model adapts the polling mechanism to enable users to retrieve the state of other entities.
- The Publish/Subscribe (Pub/Sub) model, which is based on an event broker to forward updates (notifications) to interested users (subscribers), regarding changes of senders' (publishers') statuses, as illustrated in Figure 3.10b.

The Req/Res model is adopted by common transport protocols like HTTP and CoAP [64], which apply the RESTful architecture. Other protocols adopting the Pub/Sub model include MQTT and AMQP. The authors in [99] analyzed the strengths and weaknesses of both paradigms in supporting communications in ubiquitous systems. The authors conclude that the communication semantics of the developed solution is the main criteria to choose which model to implement. On the one hand, the polling mechanism used in the Req/Res model is considered insufficient to be implemented within systems that have infrequent status changes. On the other hand, the Pub/Sub model lacks the end-to-end delivery reliability due to the existence of intermediary entities (i.e. broker) between publishers and subscribers.



(a) Req/Res Model



(b) Pub/Sub Model

Figure 3.10: Protocol Models

The lack of a protocol that handles different requirements of vertical M2M/IoT applications has resulted in a fragmented market between many protocols. The comparison between different M2M/IoT protocols has been a subject of discussions in recent literature work [100, 101, 94]. Table 3.5 presents a comparison of the M2M transport protocols reviewed in previous subsections.

Table 3.5: M2M Transport Protocols Comparison

Protocol	HTTPv1	HTTPv2	CoAP	MQTT	AMQP
Standards	IETF RFC2616	IETF RFC 7540	IETF RFC7252	Proposed OASIS standard MQTT	OASIS AMQP
Architecture Style	Client/server model RESTful	Client/servers model	Client/server model RESTful	Brokered style	Brokered style
Transport	TCP	TCP	UDP	TCP	TCP
Messaging	Request/Response	Supports multiplexing of request/response	Request/Response	Publish/Subscribe (P2P or Brokered)	Publish/Subscribe
Header	Text-based	Binary (header compression)	4Byte Binary-based	Fixed length of 2Byte	8Byte
Message size	Larger, partly because status detail is text-based	Configurable by server	Small to fit in single IP datagram with 4byte header	Up to 256MB with 2byte header	Unlimited with 8byte header
Dynamic Discovery	No	No	Yes	No	No
Service levels (QoS)	All messages get the same level of service	Priority mechanism of streams	Confirmable or non-confirmable messages	Three quality of service settings	Different 3 QoS levels
Data distribution	One-to-one	One-to-one and one-to-many	One-to-one	One-to-one and one-to-many	One-to-one and one-to-many
Security	Typically based on Secure Sockets Layer (SSL) or TLS	Requires TLS version 1.2 or higher	Typically based on SSL or TLS	Simple Username/Password Authentication, SSL for data encryption	SASL authentication, TLS for data encryption

The HTTPv1 is an ideal Internet transport protocol for requesting data from known sources. However, it is not suitable for resource-constrained devices most likely to be used in M2M communications, due to the high resources consumption of opening and closing TCP connections frequently. Also, it does not provide scalable means for bi-direction communication such as sending notifications, and the textual encoding of HTTP headers obtains unnecessary overhead for parsing. The second major version of the HTTP protocol, known as HTTPv2 or H2, is focusing on performance; i.e. end-user perceived latency, network and server resource usage. One major goal is to allow multiplexing of requests and responses to avoid the head-of-line blocking problem in HTTPv1. A server pushing functionality is also specified to enable pushing resources to clients in a Pub/Sub model, this feature is useful when sending notifications on M2M systems. Additionally, an optimized binary encoding for the header is introduced in order to reduce the needed network bandwidth.

Relatively few studies have been published on evaluating comparison between transport protocols for the IoT. A comparative study of both CoAP and HTTP, published in [102], shows by means of simulation the benefits of CoAP in terms of energy consumption and response time comparing to HTTP. Another study of performance evaluation in terms of latency, memory occupation, and energy consumption for CoAP and HTTP over both TCP and UDP is presented in [94].

The authors of [101] present a compression of MQTT-S and CoAP by means of simulation. The study observes that the maximum achieved Discarded Publication

Ratio (DPR) by CoAP is better than what is achieved by MQTT-S [100]. MQTT-S, currently known as MQTT-SN, aims to extend the MQTT protocol beyond the reach of TCP/IP infrastructure for sensors and actuators.

3.4 Standardization for Wide and Local Area Connectivity

Since many IoT/M2M capable devices require to be connected to M2M platforms via underlying communication networks such as cellular networks, M2M related standards have taken the initiative to support architectural interworking functions between the service layer platform and the underlying communication network.

3.4.1 3GPP Machine Type Communication (MTC)

The 3GPP refers to M2M communication as MTC starting from the Evolved Packet Core (EPC) architecture. 3GPP started standardization activities on MTC in September 2008 as part of 3GPP Rel-10 specifications. The service requirements working group (3GPP SA WG1) had specified a number of use cases and scenarios, and derived a set of service requirements accordingly [14]. In 3GPP Rel-11, some of the proposed MTC features were finalized, such as addressing and device triggering. Most important is the enhancement of the 3GPP architecture to support MTC applications [103], by introducing a Machine-Type Communications (MTC)-InterWorking Function (MTC-IWF) to interact an external MTC Capability Server with the Mobility Management Entity (MME), Home Subscriber Server (HSS) and Master of Science (MSC). The Access Network Discovery and Selection Function (ANDSF) allows application servers (e.g. M2M platform) to trigger the M2M end devices to select certain wireless access technologies according to defined policies [104].

In Rel-12, new enhancements for small data transmission and minimizing overheads are considered in addition to enhancements of device triggering methods by using reference points between MTC-IWF and serving nodes (i.e., SGSN, MME, and MSC). The standard will also intend to optimize the User Equipment/Endpoint (UE) power consumption to prevent battery drain, and enable group-based features that allow multicast communication to a MTC group of devices sharing one or more MTC features. Figure 3.11 shows the 3GPP non-roaming reference model for MTC based on 3GPP specification [105].

For Rel-13, expected in 2016, a new MTC device category with lower complexity and additional power saving techniques will be defined [106].

3.4.2 IEEE 802 LAN/MAN Standards

Due to the nature of connected objects within the IoT, very low power consumptions are required to enable any object to plug into the Internet while being powered by

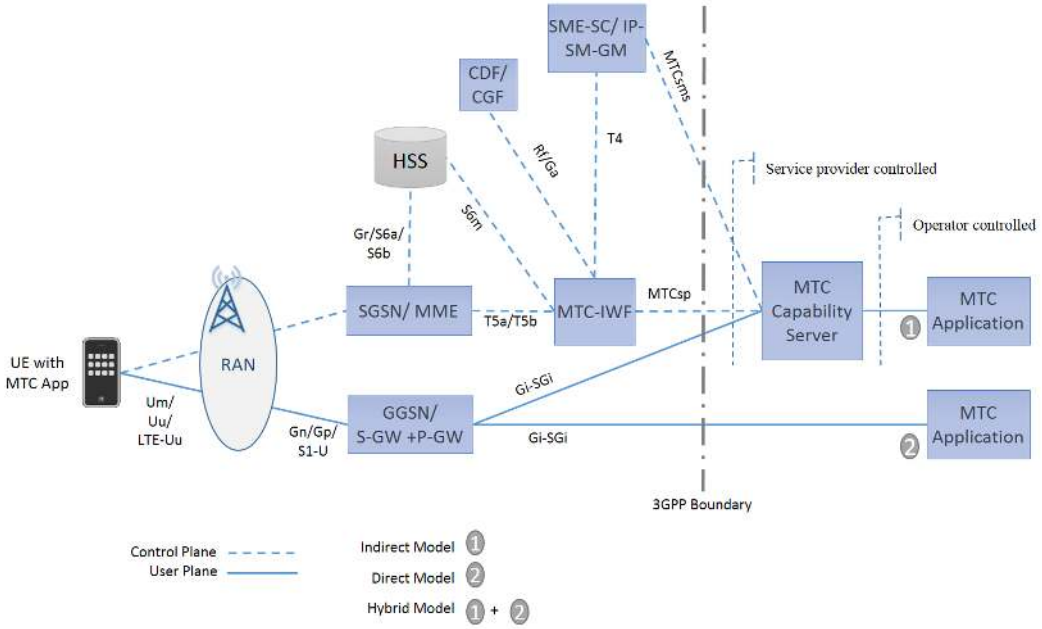


Figure 3.11: Reference Architecture for 3GPP MTC based on 3GPP TR 23.888 Rel.11

batteries or through energy-harvesting. In the past few years, there have been many efforts to enable the extension of Internet technologies to constrained devices.

Several IEEE task groups (TGs) are addressing the impact of M2M communication on the IEEE 802 radio access networks.

- **WiFi IEEE 802.11**: 802.11 supports different communication setups, each of them enabling a different scenario. The IEEE 802.11p defines enhancements to the 802.11 standard to support inter-vehicle communication [107]. Recently, the IEEE 802.11ah wireless LAN standard group targets a wireless communication standard to support use cases that include sensor networks and backhaul communications of sensors data for M2M communications. The target is to enhance the designs of the physical and MAC layers of IEEE 802.11ac so that it operates in free sub 1GHz bands [108]. The lower center frequencies provides longer distances comparing to typical WLAN frequencies around 2.4 GHz and 5 GHz.

- **The IEEE 802.15.4:** defines the Medium Access Control (MAC) and physical (PHY) layers in Low-Rate Wireless Personal Area Network (LR-WPAN). In order to achieve better energy-efficiency, IEEE 802.15.4 can operate in a so called beacon-enabled mode for which a superframe structure is utilized [109]. The ZigBee alliance (http://www.zigbee.org/) has recently developed further network and application layer protocols using small low-power radio devices based on the IEEE 802.15.4. The target applications include Smart Energy,

health care, remote control consumer electronics equipment, etc. The IEEE 802.15.4 provides a reliable communication and could handle a big number on nodes. However, it does not support QoS guarantees.

- **IEEE 802.15.1**: this is the basis for the Bluetooth wireless communication technology. Bluetooth is designed for small and low cost devices to support short range communication with low power consumption. The technology operates with three different classes of devices: Class 1, class 2 and class 3 where the range is about 100 meters, 10 meters and 1 meter respectively. Bluetooth operates in the same 2.4 GHz frequency band as Wireless LAN, but using different signaling methods to prevent interference. The Bluetooth Special Interest Group (SIG) has completed the Bluetooth v4.0 specification, called Bluetooth Smart. It includes Classic Bluetooth, Bluetooth high speed and Bluetooth Low Energy (BLE) protocols. BLE aims at supporting low power sensor devices, and uses a GFSK (gaussian frequency shift keying) modulation to transmit the data [66]. A comparative analysis of BLE and ZigBee/802.15.4 is published in [110], and showed that BLE is more energy efficient in terms of number of bytes transferred per Joule spent.

- **IEEE 802.16**: is a standard technology for wireless wideband access. Among its advantages, the ease of installation is by far the most important aspect. This technology supports either point-to-multipoint or mesh topologies. The IEEE 802.16p TG aims for enhancing the mobile WiMAX base standards IEEE 802.16e and IEEE 802.16m for M2M, identifying a number of requirements for mainly MAC-related functions such as network entry, group and device addressing, etc. [49].

3.4.3 IETF 6LoWPAN

In order to enable low-power devices with limited processing capabilities to participate in the IoT, the IETF 6LoWPAN Working Group was formed to work on the IPv6 protocol extensions required for such networks where the nodes are interconnected by IEEE 802.15.4 radios. The 6LoWPAN WG defined encapsulation and header compression mechanisms that allow IPv6 packets to be sent to and received from over IEEE 802.15.4 based networks [111]. It defines a LoWPAN frame format for IPv6 data packets and a simple header compression scheme, which uses shared context information.

6LoWPAN introduces the adaptation layer between network and data link layers. This allows IPv6 datagrams to meet the requirements of the IEEE 802.15.4. The IPv6 standard defines a Maximum Transmission Unit (MTU) fixed to 1280 bytes, while IEEE 802.15.4 defines it equal to 127 bytes. The length of the IPv6 header (40-bytes) implies a huge overhead that, considering the presence of transport layer header (8 bytes for UDP), MAC header (25 bytes) and link-layer security (21 bytes) would leave only 33 bytes available for application layer payload. The adaptation layer solves these problems by enabling the compression of the IPv6 header and

the fragmentation of packets that exceed the MTU of the MAC layer. In case of fragmentation, a fragmentation header is appended to each fragment. Two distinct headers are used to indicate whether it corresponds to the first fragment or is one of the followings. Besides, those aspects on link layer protocol adaptation for low-power devices and routing protocol, 6LoWPAN defines some standards on network management, neighbour discovery, and mobility, i.e. adapted from Mobile IPv6 [111].

3.4.4 Discussion

The wide diversity of connected objects and the dissimilar QoS constraints for M2M applications lead to a fragmented protocol stack in M2M communication. The necessity of developing and designing an efficient networking concept have been remarked by many standardization bodies. Their prime focus has been to support M2M communication in the existing networks by developing specification that enable basic M2M communication requirements. The majority of M2M standard bodies have proposed a hierarchical M2M architecture based on IP protocol. However, IP may be too complex for small devices such as sensors due to their energy constraints. There are numerous initiatives trying to overcome this issue by developing simplified IP stacks over existing low energy protocol suites. Among them the IETF 6LoWPAN protocol to enable the transmission of IPv6 datagrams over low-power networks based on the IEEE 802.15.4 standard. Also the use of BLE over IPv6 has been recently finalized in RFC7668 [112]. These initiatives are usually accompanied by using CoAP on the application layer. The usage of IPv6 protocol provides a highly scalable address scheme suitable for M2M/IoT deployments. In [113], the performance of 6LoWPAN in Wireless Sensors Network (WSN) is evaluated using a testbed consisting of embedded components, the results show the effect of the packet size on round trip time. Some other problems have been reported such as high rate of packet loss, and ease of interference.

Both 3GPP and IEEE 802.16 (WiMax) have addressed the problem of M2M devices connecting to a base station (BS). The enhanced features specified by 3GPP ensure that LTE can meet M2M requirements of low-cost devices, ubiquitous coverage, and ultra-long battery life [106]. The approach of grouping BS and M2M devices is considered by 3GPP and IEEE 802.16p to tackle the increased number of connected devises to one BS.

Similarly, the IEEE 802.11ah Working Group developed standards for M2M communication utilizing the sub 1GHz band. This standard address the scalability problem and constrained devices. Due to the propagation properties and the simpler needed device components, power consumption could be decreased at these frequencies [108]. It is expected that such features will make the 802.11ah radio technology highly attractive for deployment in rural areas.

Both ZigBee and Z-Wave have been used widely in Smart Home applications. Furthermore, Z-Wave applications can benefit from the flexibility and security of this protocol. Its overall performance has been reported to be superior to ZigBee‘s

performance [114]. However, ZigBee is more affordable and has been included in ETSI specification as one solution for M2M capillary networks in order to meet the Smart Energy market needs. Table 3.6 summarizes the main characteristics of access protocols commonly used in M2M/IoT.

Table 3.6: Comparison of Wireless Technologies for M2M/IoT

Technology	Bluetooth/ Bluetooth Low Energy (BLE)	802.11 (Wi-Fi)	802.15.4 (ZigBee/6LoWPAN)	RFID	Cellular
Security	64/128bit AES CCM	256 bits AES encryption	128 bit, AES	low	confidentiality and integrity algorithms
Latency	100ms/ <3ms (LE)	1.5m	20ms	-	90ms
Max data rate	3Mb/s (enhanced)	22Mb/s (802.11 g)	250Kb/s	Varies	12Mb/s (4G LTE)
Range	10-100 meters	50-100 meters	10-200 meters	<3m	>1000m
Mobility	fixed	nomadic subnet roaming	Yes	Fixed	Seamless global roaming
Power Consumption	Medium/ Low (LE)	High	Low	Low	Medium
Battery life	Days years (LE)	Hours	Years	Years	Days
Frequency Band	2.4GHz	2.4GHz, 3.6GHz and 5GHz	2.4GHz, 868MHz and 915MHz	-	800MHz, 1.8GHz, 2.0GHz and 2.6GHz

3.5 Research and Projects Activities

There are several research projects working towards defining a reference architecture for IoT system deployments, such as IoT-A, and Fi-Ware, while others are aiming to create a global IoT platform for vertical applications, such as OpenIoT, BUTLER, and OM2M. Each of these projects has specific contributions in addressing the challenges of IoT and M2M communication. However, there are still further work required to cover the technological complexity and vertical M2M silos. In this section, a short review of the work and achievements of previously mentioned projects is provided, followed with an analysis of the requirements addressed by each project.

3.5.1 Internet of Things-Architecture (IoT-A) Project

This project, co-funded by the European Commission within the Seventh Framework Programme (2007-2013), focused on designing a comprehensive framework that would facilitate common approach to design the IoT architecture. The IoT-A main objective is to define a generic Architecture Reference Model (ARM) that could be used to derive concrete IoT architectures, through providing a number of means (models, views, perspectives, best practices, etc.) that can be used to derive an IoT architecture. Based on this reference ARM, multiple architectures could be

developed, each focuses on a domain-specific IoT application. The authors in [115] provide a guidance on how to derive concrete architectures from the IoT-A ARM.

Other common objectives of IoT–A are to design protocols and interfaces for end-to-end interoperability between different IoT devices and to design powerful security and privacy mechanisms and scalability to develop IoT device platform components. In General, IoT-A promotes interoperability in the IoT at the communication level as well as at the service level across different platforms established on a common grounding. This could be achieved by providing to IoT system developers a common technical foundation and set of guidelines for deriving a concrete IoT system architecture [116]. The ARM consists of three interconnected parts [116]:

The IoT Reference Model (RM): providing a set of models that are used to define certain aspects and definitions of the architecture to be developed independent of specific technologies and use-cases. The Reference Model consists of several sub-models that set the scope for the IoT design space and that address architectural views and perspectives. The IoT Domain Model is the primary model that describes all the concepts relevant in the IoT architecture,such as Devices, IoT Services and Virtual Entities (VE), to provide the bases of all other models and it also introduces relations between these concepts. Based on the IoT Domain Model, the IoT Information Model, Communication Model and the Trust, Security, and Privacy Model will be developed. The IoT Information Model defines the structure (e.g. relations, attributes) of IoT related information in an IoT system on a conceptual level without discussing how it would be represented. The IoT Functional Model identifies groups of functionalities, of which most are grounded in key concepts of the IoT Domain Model. A number of these Functionality Groups (FG) build on each other, following the relations identified in the IoT Domain Model. The Functionality Groups provide the functionalities for interacting with the instances of these concepts or managing the information related to the concepts, e.g. information about Virtual Entities or descriptions of IoT Services. The functionalities of the FGs that manage information use the IoT Information Model as the basis for structuring their information. A key functionality in any distributed computer system is the communication between the different components. One of the characteristics of IoT systems is often the heterogeneity of communication technologies employed, which often is a direct reflection of the complex needs such systems have to meet. The IoT Communication Model introduces concepts for handling the complexity of communication in heterogeneous IoT environments. Communication also constitutes one FG in the IoT Functional Model. Finally, Trust, Security and Privacy (TSP) are important in typical IoT use-case scenarios. Therefore, the relevant functionalities and their interdependencies and interactions are introduced in the IoT TSP Model. As in the case of communication, security constitutes one FG in the Functional Model.

The IoT Reference Architecture (RA): consisting of a set of Views, each rep-

resents one or more structural aspects of an architecture to illustrate how the architecture addresses one or more concerns. It is important to keep the RA abstract in order to enable different IoT architectures. The IoT RA includes a Functional View, Information View, and Deployment and Operation View. In these views of the IoT Reference Architecture, interactions are not covered since the number of arrangements of the functional components and also their invocation is practically infinite. The Functional view is constructed from the IoT Functional Model and the unified requirements. Figure 3.12 depicts the Functional view diagram and shows the nine functionality groups of the Functional Model. Based on the IoT Information Model, the Information view provides more details about how the relevant information is to be represented in an IoT system. However, concrete representation alternatives are not part of this view. The information view also describes the components that handle the information, the flow of information through the system and the life cycle of information in the system. Additionally, the Deployment and Operation view addresses the realization of actual system by selecting technologies and making them communicate and operate in a comprehensive way.

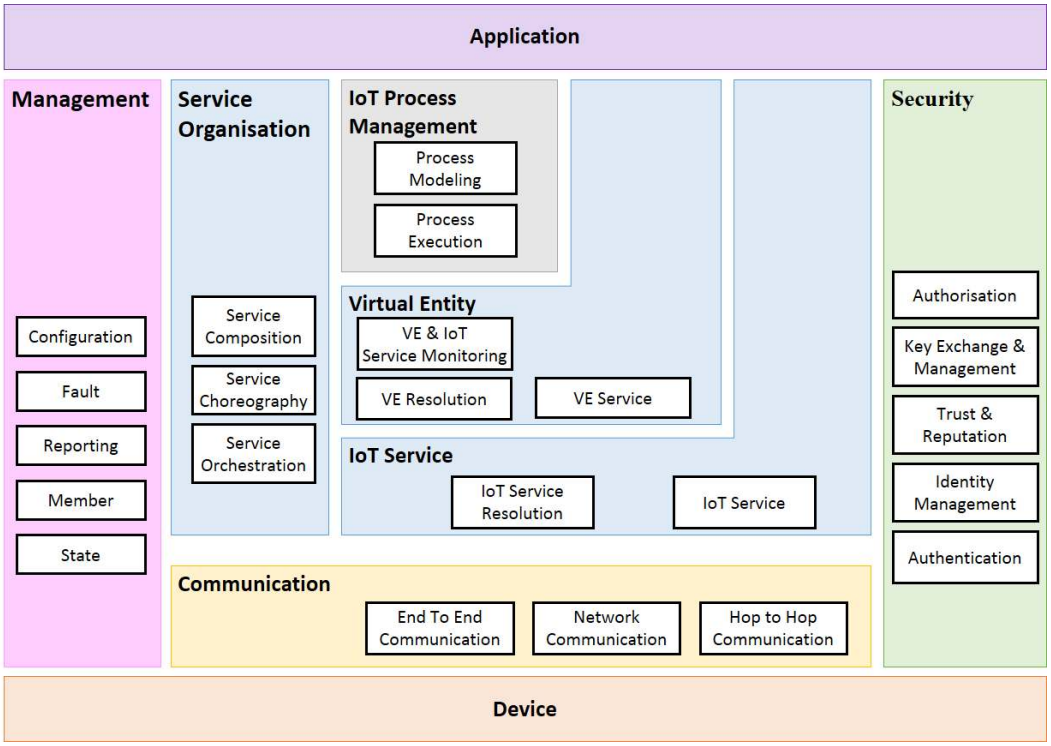


Figure 3.12: IoT-A ARM Functional View based on [116]

The Guidance: explaining the usage of the IoT ARM. It defines the process leading to the generation of the concrete domain-specific IoT architectures based on the RA and RM. The Guidance part contains extensive treatises on how

to use the IoT-A unified requirements on the common contents of an IoT threat analysis; and on how qualitative requirements are translated into design choices concerning the functional view and the information view.

3.5.2 FP7 FI-WARE Project

Fi-Ware is a European FP7 Research Project aiming to foster the emerging Future Internet by creating an open architecture and a reference implementation of a novel service infrastructure. Endorsed by a large industrial community support, the FI-WARE project [12] started to work on designing a core platform for the next generation Internet and as a part of that effort produced architecture for the IoT domain. This platform is open sourced, based upon components referred to as Generic Enablers (GE) which offer reusable and commonly shared functions serving a multiplicity of Usage Areas across various sectors [12].

The Fi-ware generic enablers are classified into seven major groups providing architecture reference model for the specific features addressed within these chapters [117]:

1. Cloud Hosting: computation, storage and network resources, upon which services are provisioned and managed.

2. Data/Context Management: accessing, processing, and analyzing massive volume of data, transforming them into valuable knowledge available to applications.

3. Applications/Services Ecosystem and Delivery Framework: the infrastructure to create, publish, manage and consume FI services across their life cycle, addressing all technical and business aspects.

4. Internet of Things (IoT) Services Enablement: the bridge whereby FI services interface and leverage the ubiquity of heterogeneous, resource-constrained devices in the Internet of Things.

5. Security: the mechanisms which ensure that the delivery and usage of services is trustworthy and meets security and privacy requirements.

6. Advanced middleware, Interface to Networks and Robotics: provides interfaces to run an open and standardised network infrastructure (underlying a network operator control or network virtualization) along with providing access to specific features of highly sophisticated terminals and cloud proxies.

7. Advanced Web-based UI: brings components that will provide a simple, uniform way to create rich networked 2D and 3D applications that run in a browser.

The IoT architecture defined by the FI-WARE project has already taken into account the ETSI M2M specification and has extended it to incorporate OMA NGSI

Context Management activities. Figure 3.13 shows the IoT architecture of FIWARE, which consists of two different domains: the IoT Backend, and the IoT Edge [118]. The GEs shown in Figure 3.13 [118] implement the functionalities distributed across these domains. The IoT BackEnd comprises the set of functions, logical resources and services hosted in a Cloud datacenter. It is connected to the IoT edge elements that represent on-field IoT infrastructure elements needed to connect physical devices to applications.

The BackEnd component consists of three main GEs, namely IoT Broker, IoT Discovery and IoT Device Management. These components provide both REST and NGSI interfaces for interaction with the users as well as appropriate features such as things and resources management using NGSI Context entities, protocol adaptation and connectivity management, and devices composition and discovery functionality.

The IoT Broker GE is responsible for retrieving and aggregating information from the connected devices. While the IoT-Discovery GE is responsible for context source availability management, based on the OMA NGSI-9 Context Management Information Model. The IoT Device Management GE is the central component for most common scenarios. This GE provides the resource-level management of remote assets i.e. devices with sensors and/or actuators, as well as core communication capabilities such as basic IP connectivity and management of disconnected devices.

The gateways at the Edge domain provides similar functionality, but on the local level, i.e. it provides resource management functionality for connected things of the

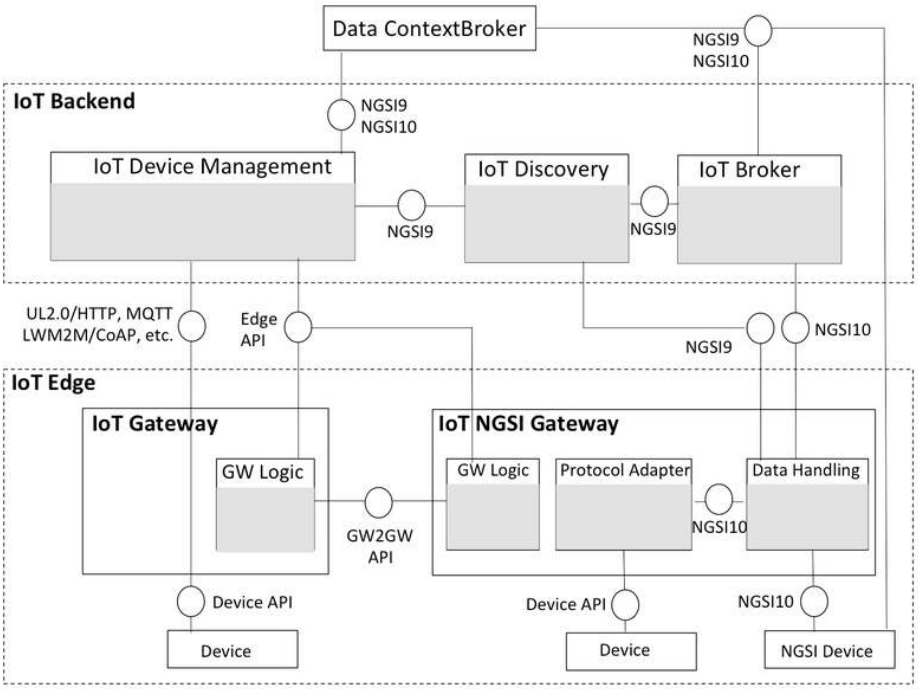


Figure 3.13: FIWARE IoT Architecture [118]

IoT and legacy devices connected to the gateway. An NGSI gateway consists of two GEs, namely Data Handling and Protocol Adapter, as well as the GW Logic component, which handles the APIs of the gateway-to-gateway and the IoT Edge configuration at the gateway level. The Data Handling GE addresses the need for filtering, aggregating and merging real-time data from different sources. And the Protocol Adapter GE deals with the incoming and outgoing traffic and messages between the Gateway and Devices registered, to be served by the gateway towards the Data Handling GE. The Protocol Adapter GE translates device specific protocols into a uniform internal API.

3.5.3 FP7 OpenIoT - Open Source cloud solution for the Internet of Things

The OpenIoT project focused on providing an open source middleware framework enabling the dynamic formulation of self-managed IoT environments and applications [119]. The project adapted the cloud computing infrastructure model to deliver sensing service in an autonomic fashion, including utility based security and privacy schemes.

OpenIoT can act as a blueprint framework that will allow solution providers and users to integrate within IoT cloud systems, and select the most appropriate sensors for a given service while filtering their data. In particular, OpenIoT provides the means for formulating and managing environments comprising IoT resources, which can deliver on-demand utility IoT services such as sensing as a service.

From September 2013, OpenIoT middleware platform was made available to the Open Source community (https://github.com/OpenIotOrg/openiot/) for creating real-time IoT services on demand and enable interoperability between vertical IoT solutions and interconnect data silos. The OpenIoT architecture comprises seven main elements: 1) the Sensor Middleware (SM) that extended the Global Sensor Networks (X-GSN), 2) the Cloud Data Storage enabling storage of data streams on the cloud, 3) the Scheduler that processes requests for on-demand deployment of services, 4) the Service Delivery and Utility Manager that combines data streams as indicated by service's workflow, 5) the Request Definition component which enables on-the-fly specification of service requests to the platform, 6) the Request Presentation component that enables the visualization of the outputs of a service, and 7) the Configuration and Monitoring component which enables visual management and configuration of functionalities over sensors and services.

In order to support the dynamic integration and discovery, the openIoT middleware developed semantic models and annotations for representing Internet-connected objects, along with an IoT ontology based on the W3C Semantic Sensor Networks (SSN) ontology, the SPITFIRE ontology (spt) and the LSM vocabulary (lsm). The OpenIoT ontology represents a universally adopted terminology for the convergence of sensed data with the semantic web. It enhances existing vocabularies for sensors and Internet Connected Objects (ICOs), with additional concepts relevant to IoT/cloud integration such as terms to annotate units of measurement, raw sensor

values and points of interest at some specific levels of granularity [120].

In addition, an integrated development environment is provided by openIoT for deploying and managing IoT applications. OpenIoT IDE comprises a range of visual tools enabling the visual definition of IoT services in a way that obviates the need to master the details of the SPARQL language.

3.5.4 FP7 Butler

BUTLER [13] is a European research project whose main objectives are to develop a horizontal platform features and several IoT applications and services, through integrating existing technologies and develops new technologies. The main mission is to provide context-aware services within the IoT environment. For the achievement of above objectives and support different domains, the BUTLER platform focuses on providing them with communication, location and context awareness abilities, while guaranteeing their security and the privacy of the end users [121].

The project started from the gathering and analysis of the requirements from up to 70 use cases, in order to produced requirements for the platform's specification and valuable information on the potential socio-economic impact of the deployment of an horizontal IoT platform. The BUTLER framework is offering common functionalities on three types of platforms, which are categorized based on their intended use and prime capabilities, namely, Smart Objects (i.e. sensors, actuators, gateways), Smart Mobiles (user's personal device) and Smart Servers (providers of contents and services) [122]. The BUTLER security framework was designed to enable end-to-end security between a data provider and a data consumer, while being compatible with the ETSI-M2M access right feature. Furthermore, the deployed protocols ensure confidentiality, integrity of the messages and authentication of the peers.

In order to address the dynamic data demands, the localization capability was an emerging issue considered in BUTLER. An improved ranging algorithm using super-resolution techniques over phase measurements for distance estimation between devices has been developed within BUTLER. As part of the experimentation of the BUTLER platform, the project has conducted several IoT deployment that include large scale deployments in order to showcase the outcomes of the project to end users and stakeholders outside of the IoT community.

3.5.5 Eclipse OpenM2M (OM2M)

OpenM2M (OM2M) project (http://www.eclipse.org/om2m) is a member of Eclipse IoT Working Group, providing a modular architecture running on top of an OSGi layer based on Eclipse Equinox, using Java programming language and Apache Maven [123]. The software is an open-source implementation of the ETSI M2M standard distributed under the Eclipse Public License (EPL). It aims to facilitate the deployment of vertical M2M applications by providing a horizontal M2M service platform for developing services independently of the underlying network.

The platform consist of a SCL that could be deployed in an M2M network, a gateway, or a device. The OM2M SCL provides the functionalities of mandatory service capabilities, these are: Application enablement(xAE), Generic communication(xGC), Reachability, addressing, and repository(xRAR), Communication selection(xCS), Remote entity management(xREM), SECurity(xSEC) and Interworking proxy(xIP). The OM2M platform provides a RESTful API for XML data exchange only, through unreliable connections. RESTful API provides primitive procedures for machine authentication, resource discovery, application registration, etc.

To ensure extensibility of the system, the SCL is composed of small tightly coupled plugins, each one, offering specific functionalities. A plugin can be remotely installed, started, stopped, updated, and uninstalled without requiring a reboot [123]. On the time of writing this dissertation, the current implementation supports HTTP and CoAP transport protocols, and integration of legacy devices using Zigbee and Phidgets technologies. However, the platform could be extended to perform device firmware updates by reusing existing protocols and interworking proxy.

3.5.6 Discussion

The IoT-A project presented the direction of defining a general Reference Model Architecture for IoT, to be used as a basis of platforms design. The project has created an "Architectural Reference Model" (IoT ARM) as the common ground for the field of IoT. The model defines entities and describes their basic interactions and relationships with each other [115]. It also has resulted in a large amount of background information that includes the uses of IoT architectures in developing general or domain-specific platforms. A number of research projects have taken the IoT-A reference models as its main inspiration, such as BUTLER. However, the resulted architecture of different other projects could be easily mapped to the IoT-A models [115, 124]. It becomes clear that the scope of IoT-A is broad enough to cover the complete IoT domain, providing a fine-grained set of relationships between different types of resources and services.

The Fi-Ware service infrastructure that is developed upon reusable Generic Enablers, i.e. utilities services, is making the development of value-added services easier. A number of research projects have been established as a phase-two use case projects that aim to validate the FI-Ware Generic Enabler in specific use cases, such as FI-STAR [125], XIFI [126] and FRACTALS [127]. Additionally, many projects have used some enablers to speed up the development of their solutions. The compatibility of FI-Ware IoT Generic Enablers with ETSI M2M service layer is quite notable, which facilitates the interoperability of the developed applications with other platforms. However, global interoperability with international standards is still missing.

The way of addressing the heterogeneity of device's capabilities issue is ideally supporting a pool of standardized communication protocols in which the device manufacture can select the appropriate protocol in each case. Unfortunately, this is not the case in some commercial platforms that propose connection via proprietary

gateway, which complicates the mashing up of data and services across multiple platforms.

Providing APIs is crucial for the development innovative services in the FI, and with consideration of the huge heterogeneity nature of networks and devices in the M2M platform, the need of standard interfaces to M2M platforms is more clear [35]. In this direction, Fi-Ware IoT enablers adopted the OMA NGSI interfaces, while some other platforms developed RESTful APIs to make it transparent to developers in order to build applications, without the need to learn deep details of the operator middleware organization. To some extend, some platforms offers libraries binding for different programming languages, such as BUTLER, however, these libraries usually include basic functionalities only. Some (commercial) platforms provide a full Software Development Kit (SDK) in one or more programming languages; this approach could be extended further by developing Domain-Specific Language (DSL), which exposes the functionalities specific to the M2M/IoT to the domain experts. The OpenIoT Virtual Development Kit (OpenIoT-VDK) is an example of this approach.

Generally, heterogeneous applications, devices, technologies and requirements are emerging in M2M/IoT systems. Standardization efforts aims to make it easier for individual stakeholders to partner and interwork with component providers, application developers, solution integrators, data and content owners and with wireless and wireline connectivity providers. At the same time, various research work have been conducted with the aim to develop a horizontal solution that could combine and coordinate M2M devices, mostly for specific-domain IoT services. However, the limited support of cross-platform integration in existing solutions, specially proprietary platforms, is preventing the realization of large-scale IoT, which could result in fragmented IoT vertical silos. Furthermore, renewed business and revenue models should drive these aspects of the IoT to be charged on an consumption-based model, i.e. Everything-as-a-Service (XaaS). Thus becoming available in professional private-public clouds or in the Internet Web as user generated services.

4 Communication Requirements Towards Reliable Smart Service Deployment

This chapter reviews the requirements of reliable Smart services by investigating the usage natural, requirements and traffic patterns of a number of Smart City services. A Smart service is considered reliable when it performs according to its specifications, i.e. it executes the required functionality under stated conditions for a specified period of time.

4.1 Introduction

Nowadays, cities and urban areas are forming complex ecosystems, where ensuring sustainable development and quality of life is an important concern. In such urban environments, people, businesses and public authorities experience specific require-

ments regarding innovative and interoperable services. According to the United Nations, more than half of the World's population lives in urban areas, and it is suggested that thousands of new cities need to be built worldwide by 2050. Less than 30% of the world's population lived in cities in 1950, this number grew to 47% in the year 2000 (2.8 billion people), and it is expected to grow up to 60% by the year 2025. Furthermore, it is expected that most urban growth will occur in less developed countries during the next decades [128]. Experts point out Smart Cities as an emerging market with enormous potential, which is expected to drive the digital economy forward in the coming years.

A report from the International Organization for Standardization (ISO)/IEC Technical committee of Information technology [129] analysed the standardization opportunities of Smart Cities. The report highlighted the networking of collaborative spaces within the city as a key requirement, in order to enable dynamic communities that will spur innovation and growth, and enhance citizen well-being. Due to the ability of IoT systems of performing situated sensing, they have huge potentialities for developing new innovative applications in Smart Cities as well as in many other fields. Smart City is widely considered a hot topic; however, there is no clear definition of the Smart City concept among practitioners and academia. Authors in [4] represented the idea of a Smart City as a "System of systems", where the integrated systems forms a closed loop and are characterized by functions: sensing, information management, analytic and modelling, and influencing outcomes. Each system produces its own information and consumes others' information in a well-defined urban planning. A holistic definition of the term from [3] embraces six characteristics that need to act smartly to achieve a Smart City that is

> *"well performing in a forward-looking way in economy, people, governance, mobility, environment, and living, built on the smart combination of endowments and activities of self-decisive, independent and aware citizens".*

Experts from various specialities provide more concrete definitions, which emphasize the role of their own approach and activities in the field. From the ICT perspective, a Smart City can be defined as

> *"A city connecting the physical infrastructure, the IT infrastructure, the social infrastructure, and the business infrastructure to leverage the collective intelligence of the city"* [18].

This definition could be well adopted in the context of this dissertation. Figure 4.1 [26] shows service domains and related applications, i.e. industrial domain, Smart City domain, and health well-being domain. Therefore, cities are facing challenges to maintain and upgrade the needed infrastructures to establish efficient and reliable Smart City implementation that meet the demands of their citizens.

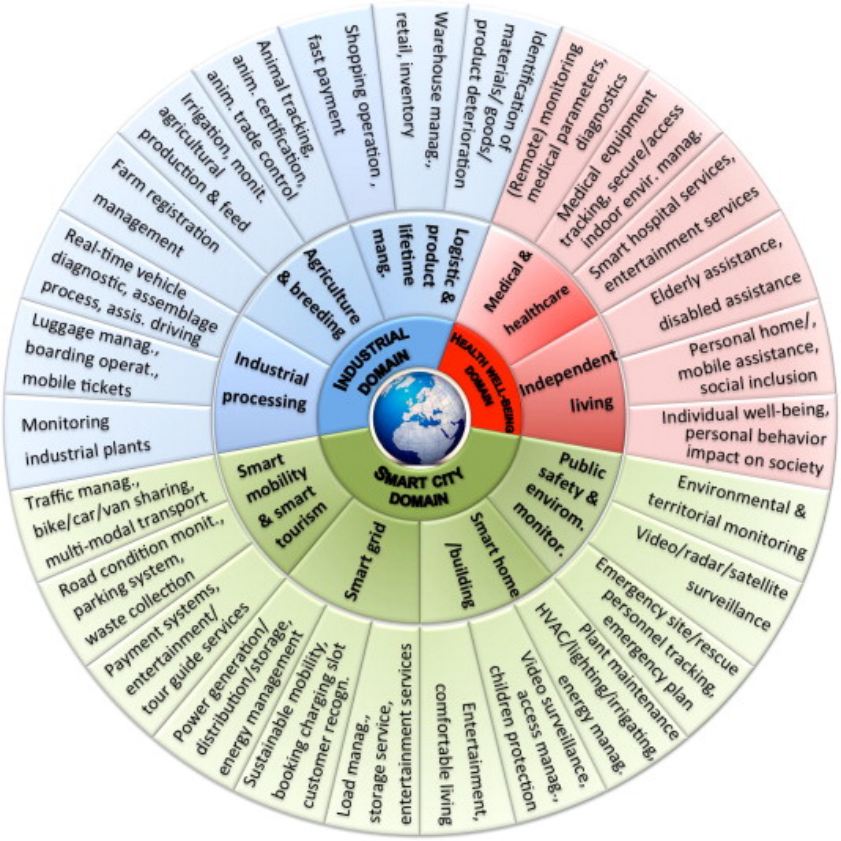


Figure 4.1: IoT Service Domains and Related Applications [26]

4.2 Use-Case-Driven Approach to M2M Requirements

The principles of the IoT and M2M communication are evolving based on existing Internet infrastructure to facilitate the representation of addressable objects and develop innovative services accordingly. There are a lot of domains and environments in which M2M communication is likely to improve the quality of our lives, such as buildings automation, transportation, healthcare, etc. Almost all SDOs started their work in M2M standardization by defining the common requirements of various Smart Services [130, 14, 52, 131].

Generally, a use case definition handles the system as a black-box where interactions, including responses, are perceived from outside the system. Use cases should not be confused with the functionalities, features, or requirements of the system under consideration. A use case may be related to one or more functionalities and/or requirements. A functionality or requirement may be related to one or more use cases [132]. In this section, considered use cases will be described in an architecturally neutral manner that does not assume any particular physical architecture. Additionally, the requirements of selected M2M use cases will be discussed. The surveyed applications have already proved their worth in industry and Smart City

development [133].

It is widely agreed that the end-to-end aspects of communication between M2M devices and servers are critical to many applications [134, 132]. There are two types of characteristics concerning technical systems; functional aspects and non-functional aspects. The first is what types of functions the system performs while the latter is how well it performs them. Figure 4.2 depicts the methodology adopted to extract the functional and non-functional requirements. It consists of three main stages: selection of use cases from Smart City domains, data collection and data analysis.

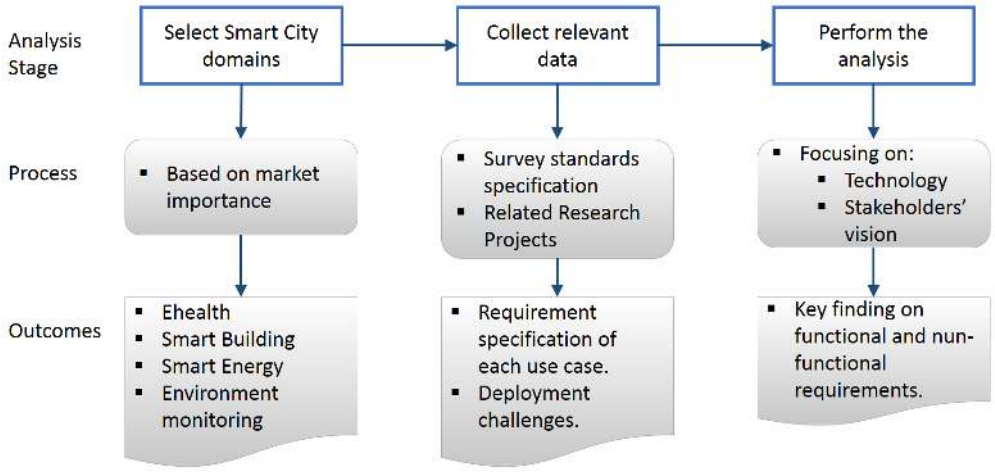


Figure 4.2: Requirements Analysis Framework

4.3 EHealth Use Case

EHealth provides a new method for using health resources, such as information, money, and medicines, with the aim to improve efficient use of these resources using ICT. EHealth systems has been thought to have great promise to improve upon traditional health communication through user-centred design and interactivity, broad social connectivity, deeper understanding of what motivates behavior change and the use of multimodal media that expand people's access to health information and discourse across time, place, and cultures [135]. A typical EHealth system consists of four components [136]:

1. Hardware devices, workstations and peripherals used to perform EHealth activities, such as capturing medical information from patient's body. The advancement of semiconductor industry shrinking lithography continues to reduce chip-set cost and power consumption, and embeds more sensors into devices used in different aspects in our daily life.

2. Users that have different roles and means to use the system, i.e. clinicians and patients.

3. Telecommunications link to connect users of the system and allow the exchange of information. Many types of telecommunication links can be used to transmit medical information; however wireless technologies are more suitable to allow patient's mobility.

4. Policies and protocols: are also needed in EHealth system, to specify users and operates roles and responsibilities, such as how patients pay the service provider and who will take a charge if technical problems exist.

The market potential of EHealth is strong. According to BCC Research report, the global telemedicine market has grown from $9.8 billion in 2010 to $11.6 billion in 2011, and is expected to continue to expand to $27.3 billion in 2016, representing a compound annual growth rate of 18.6% [137]. However, an evaluation study of the benefits of EHealth programs and the cost-effectiveness of EHealth investments is still missing [138]. Main motivations behind EHealth programs include: extending geographic access to health care services; improving diagnosis, treatment and data management; and facilitating patient communications [139]. There are enormous differences in EHealth deployment at international levels, due to the digital gap amongst developed and developing countries. Several barriers could prevent the wider uptake of EHealth in low and middle-income countries such as the lack of the necessary infrastructure to provide reliable electricity and internet access [139].

4.3.1 EHealth Service Classification and Specification

Many concepts and terms that have been propagated during the last decades concerning the use of ICT in healthcare, among them Telemedicine; EHealth and mHealth. Each provides a slightly newer attribute in the range of processing and data exchange between participants [140]. In literature, EHealth services are mainly classified based on specific objectives into the following [141]:

1. **Tele-diagnosis services**: that are generally characterized by asynchronous point-to-point communication (e.g., specialists at a remote site review transmitted patient data and return a diagnosis report).

2. **Tele-consultation**: has a similar objective to the tele-diagnosis service, but based on synchronous viewing and manipulation of medical multimedia data.

3. **Tele-monitoring**: which is the most popular service, refers to the transmission of a patient's vital bio-signals and other related data. Such services are often targeted at treating patients with chronic diseases or for post hospital home care, and may involve multi-parametric monitoring including patient vital signs (e.g., Electrocardiogram (ECG), blood pressure, saturation of peripheral oxygen (SpO2), glucose level, etc.), physical sensors (monitoring patient

activity),and environmental sensors (e.g., air temperature, humidity, and air pressure).

4. **Tele-management**: refer to a combination of advanced tele-monitoring and tele-consultation services, such as those involving computer assisted medical interventions and automatic surgical tools (i.e. tele-surgery), in addition to accessing Electronic Health Record (EHR).

5. **Tele-education**: refers to any health-related education performed at a distance and in non-emergency situations.

In a typical tele-monitoring application, a set of wearable sensors aggregate biomedical measurements into patients' gateway to monitor the status of the user continually. A Smart phone could be used as a gateway to support mobility. Through applying high-level analysis on these measurements it will allow the early discovery of any critical health condition, and actuate different actions according to the detected situation consequently. Examples of such actions include: triggering the medical sensors to increase the sampling rates so as to cope with the possible dangerous situation in a timely manner, displaying warning messages to the patient and his/her relatives with some urgent advices, or sending ambulance request to the nearest healthcare provider with information about his/her location and situation. Furthermore, the aggregated data could be reported to the medical personnel to provide feedback and treatment support to the patient periodically.

A summary of findings related to the QoS requirements for listed EHealth services is given in Table 4.1.

Table 4.1: QoS requirements for different types of EHealth services

Service type	Data Rate	Delay (Max)	Small jitter	Loss	Critical Aspect
Audio (tele-consultation)	4–25 kbps	150–400 ms	No	<3%	Delay sensitive
Video (tele-consultation)	32–384 kbps	150–400 ms	Yes	<1%	Mission critical
Vital signs (tele-monitoring)	1–32 kbps	300ms-1sec	No	zero	Continues transmission and delay sensitive
Access to EHR	NA	-	-	zero	Privacy

Figure 4.3 depicts the interaction flow between M2M nodes on a use case of EHealth tele-monitoring service, which could be applied by a medical centre for monitoring patients remotely. The applications of the client device registers first to the corresponding M2M platform, i.e. the gateway at the field domain and the back-end server at the network domain. The vital signs measured by the wearable

sensors are sent to the gateway storage and assigned appropriate access right to it, in order to prevent unauthorized access to patient's information (i.e. allow only the patient and his/her physician to access). The application controlling the biometric sensors attached to the patient's body should be configured to send aggregated measurements to update the system at the back-end server. The EHealth application used by the Medical personnel could subscribe to measurements of each patient with specifying some filter criteria that present a critical health situation of the patient (e.g., heart beat more than X or less than Y). So when any suspicious measurements in the patient record appear, the platform will send notification to the EHealth server. Consequently the system could actuate different actions according to the detected situation, such as: trigger the biomedical sensors to take more frequent measurements, send messages to the patient with some urgent advices and notify his/her physician, query for nearest healthcare provider to the patient taking his/her location and situation into account. The patient's medical records could be fetched from the platform upon request by the physician, who has granted permission of access.

4.3.2 Challenges of EHealth Solution Development

Generally, medical sensors use Wireless Body Area Network (WBAN) such as Bluetooth or Zigbee, to forward measurements to a gateway device that has a Wide Area Network (WAN) connectivity. Using a mobile capable gateway, such as a Smart mobile devices, PDAs and tablet PCs, gives the users more flexibility to carry on with their daily activities while monitoring their status constantly in real time. ETSI provided some EHealth use cases in [142] describing the requirements and information flow for each case.

The main challenges in developing a Health care communication systems could be summarized in:

1. Heightening the communication interactivity with users to encourage their active involvement in health care activities [135].

2. Interoperability of EHealth systems in order to ensure effectiveness and sustainability of developed solutions over different communication platforms [135, 143]. This rises the need of developing a robust, standard, and efficient ecosystem for EHealth systems covering the main environments.

3. Privacy is a major requirement in EHealth applications, as the user sensed medical data transferring in wireless environment from the sensor to the platform [144].

The existing research work related to cloud-base EHealth care worldwide, addresses advances in the field of medicine such as providing a reliable and cost effective personal health care [125, 145, 146]. IEEE has designed many standards in the EHealth technology to help in creating devices and systems for disease management, fitness tracking, health monitoring and independent living [147]. Among them the

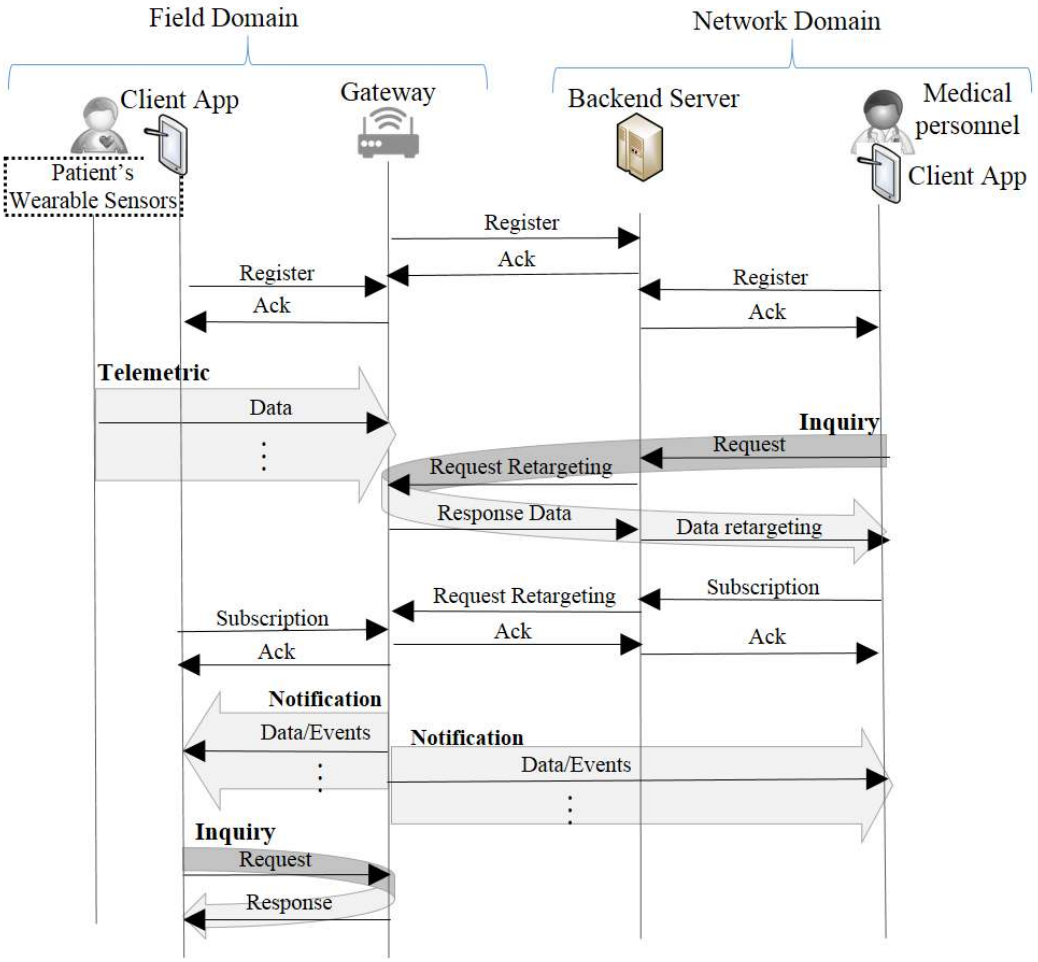


Figure 4.3: EHealth Monitoring Use Case

IEEE 11073 group of standards that address the interoperability of Personal Health Devices (PHDs).

The integrated project Future Internet Social and Technological Alignment Research in Healthcare (FISTAR) [125] have established several trials in the health care domain based on FI technology leveraged on the outcomes of FIWARE FI-PPP Phase 1 [12]. The main aim of FISTAR is to become self-sufficient by the end of the project in 2015 and to continue with a sustainable business model operated by several partners in the health care domain. In order to meet the requirements of a global EHealth industry, FISTAR is using a fundamental reverse cloud approach. Whereas the common cloud approach centralized data in unspecified locations, FISTAR aims to keep sensible patient data close to the hospital and instantiates requires software functionalities within the hospital. The patient's data will be processed in the hospital private data center cloud, without being transported to the public cloud. FISTAR is characterized by use cases, each of them individually describing a specific

scenario, in which FI technologies improve the State of the Art. Mobile telecommunication is a crucial part of this project to establish remote communication to patients, in which efficient connectivity management is required between the patient and hospital.

4.4 Smart Energy Use Case

One of the main targets of the EU 2020 growth strategy is to increase the energy efficiency by 20% and the consumption share of renewable energy [148]. One of the steps to achieve these goals is the European Smart Grid roll-out planned to be completed by 2020. The concept of distributed generations have gained more momentum recently, due to increasing energy demand and the move toward clean energy production. Traditional power grids are used to carry power from a few central generators to a large number of customers. Although this hierarchical structure is highly reliable, it does not allow full utilization of distributed resources. Currently, there is an increasing trend from one-way communications and traditional Automatic Meter Reading (AMR) systems to Smart Grid technology with advances of Advanced Metering Infrastructure (AMI) [149]. Smart Grid aims to make the existing power infrastructure more sustainable and autonomous by integrating advanced communication networks to enable Smart Grids to respond to events that occur anywhere in the grid, such as power generation, transmission, distribution, and consumption by adopting the corresponding strategy.

Generally, the Smart Grid can be defined as an

> *electric system that uses information, two-way, cyber-secure communication technologies, and computational intelligence in an integrated fashion across the entire spectrum of the energy system from the generation to the end points of consumption of the electricity.* [150]

Based on the National Institute of Standards and Technology (NIST) conceptual reference model, the Smart Grid consists of the following domains [151]:

- Customer: At customer domain, the electricity is consumed. This domain is usually partitioned into sub-domains for home, commercial building, and industrial premises.
- Market: This domain includes the operators and participants in electricity markets. In this domain the grid assets are bought and sold.
- Service providers: Organizations that provide services to electrical customers and utilities.

Smart Grid is one of many applications considered to benefit from advance functionalities provided by M2M platforms to enable efficient and optimized operation. In this regards, ETSI TC M2M has approved a document [152] that discusses several detailed use cases relevant to a Smart Grid. A general architecture of a Smart Grid

system is shown in Figure 4.4. As illustrated in the figure, electricity consumers will take part in producing electricity and putting it back into the grid. The two-way communication and utilizing Smart meters will help in balance loads and optimize the power consumption.

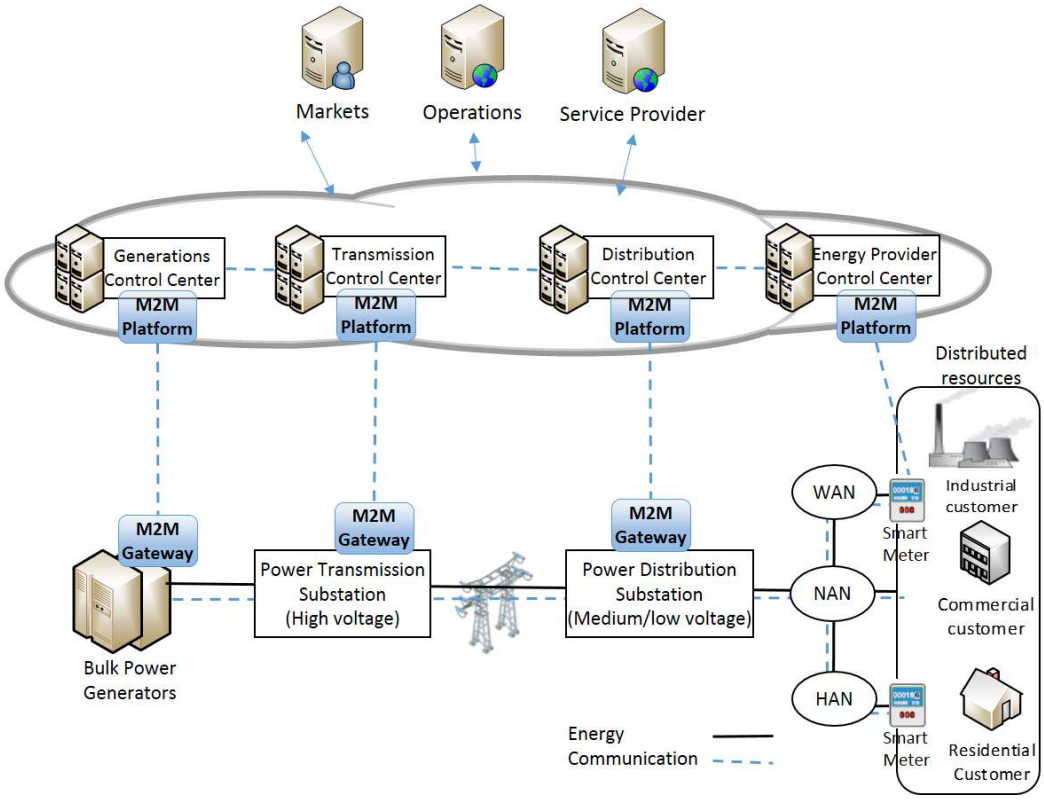


Figure 4.4: Mapping of M2M Platform to Smart Grid System

Figure 4.5 depicts the interaction modules within a substation automation use case, where the distribution system operator have to control their substations in order to detect malfunctions and avoid outages. The M2M back-end server can be placed at the control center, and the control application registers to the M2M server, subscribes to all the sensing data from the substations and also controls the functionality of the substations. At the substations an M2M gateway is located at each station as the central gateway. These are registering to the M2M back-end server and are responsible for the data transmission from the substation to the control center. The gateways and PMUs are connected to the control center via a wide area network connection.

4.4.1 Requirements of Smart Energy Deployment

The development of new management applications on the Smart Grid domain can leverage the technology and capabilities enabled by the infrastructure, in order to

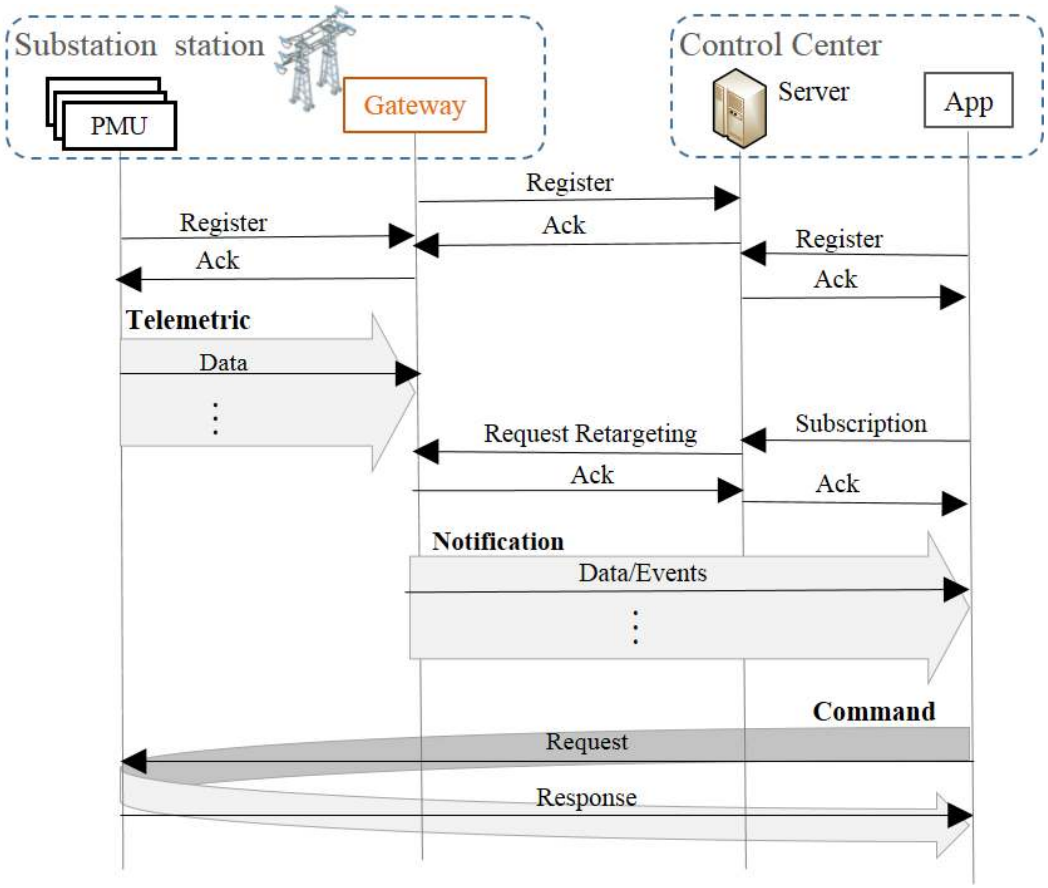


Figure 4.5: Substation Automation Use Case

improve energy efficiency, balance between supplement and demand and reduce operation costs. The new requirements and demands drove the electricity industries and research organizations to study and define roadmaps to interconnect distributed energy technologies. Some good review of existing standards activities are presented in [153] and [154]. The legacy power generation and transmission concept is converting to a massively distributed energy generation landscape by integrating broad number of variable and small renewable energy sources, such as wind and solar panels at consumer side. Thus, the term "prosumer" was introduced to define a consumer with generation capability [155].

Form the technical perspective, Smart Grid traffic could be categorized into five major categories [156]:

1. Protection traffic: Smart Grid systems control a critical resource in Smart Cities, and therefore it's essential to provide a smarter protection system that can support failure protection mechanisms, address cyber security issues and preserve privacy at the substation and in distribution systems.

2. Control traffic: produced from the field devices (sensors/actors) offering ser-

vices to substations, high and low voltage distribution networks.

3. Monitoring traffic: such as the Phasor Measurement Units (PMUs) over a large distributed area covering the electricity distribution network.

4. Metering and billing traffic: mostly servicing residential and industrial premises through Smart Meters.

5. Demand management traffic: serving different applications such as electric vehicles charging, energy storage systems, and scheduling renewable energy sources.

Designing a communication system architecture that meets the complex requirements and challenges of Smart Grid is a key factor to the successful implementation. The communication network will have to evolve to cope with both legacy and next generation applications in order to provide: i) secure communication between interworked energy subsystems, ii) sufficient QoS in terms of bandwidth and latency levels to retrieve, manage, store and integrate the large amounts of data that Smart Meters and devices will produce, and iii) reliable technology enabling the interoperability of needed standards and protocols.

The authors in [149, 157] compare wired and wireless communication technologies in terms of data rates and coverage ranges and identifies their suitability to enable some selected Smart Grid applications. For the Smart Energy communication network, latency is a critical technical requirement as some applications have a latency constraint of less than 20ms [157]. Tables 4.2 lists the communication requirements of some Smart Grid applications [149, 154].

Table 4.2: Requirements for Different Smart Grid Applications

Application	Required Bandwidth	Delay	Traffic patten	Critical Aspect
Substation Automation	1Mbps (per substation)	15-200ms	Event based/ multicast	Mission critical
AMI/AMR	10-100kbps per node, 500kbps for backhaul	Delay tolerant	Multicast/ broadcast	Large number of devices
Distribution Automation	28-128kbps	20-200ms	Periodic/ event based	Large number of devices
Wide Area Monitoring	600-1500kbps	15-200ms	Periodic	Continuous transmission
Distributed Energy Resource / Electric Vehicle	9.6-56kbps	300ms-2sec	Multicast/ broadcast	Mobility (e.g., EV roaming)

4.4.2 Challenges of Smart Energy

A major challenge in the deployment of Smart Grids, is to engage customers to participate in the dynamic energy market. By knowing their power consumption, customers are able to decide whether to use energy or even to respond to price changes and consequently to decrease their electricity bill and greenhouse gas emissions. However, survey by software giant Oracle Corp. expressed that although 95% of electricity customers would like to have detailed data about when and how they use power, only 20% are willing to pay for real-time information [158]. Moving forward, the EU needs to leverage their Smart Grid information and actively involve the consumers. Technical challenges facing the Smart Grid communication infrastructure include:

1. The Complexity of the communication infrastructure as it needs to support multi-physics approach, dynamic and configurable models. As a matter of fact, the power system is a tightly coupled non-linear system, therefore the control system as well as the communication infrastructure must be designed to manage uncertainty and inconsistencies to be resilient or gracefully degrade when necessary [159].

2. The transmission of short data bursts from large number of devices is challenged on the existing communication networks, which have been designed to support moderate number of nodes per base-station but with high data rates. Some M2M communication standardization committees have already considered this critical issue, e.g. IEEE 802.16p and LTE-advanced (LTE-A) [160]. However, it is still an open research issue that needs further investigation [156].

3. The lack of security in the Internet is a major concern of all M2M/IoT applications. With consideration to the scale of damage that could be caused by cyber attacks in Smart Grid system, the idea of using a separate network for Smart Grid communication was proposed in some academic work such as [161]. However, standardization groups from International Electrotechnical Commission (IEC) and NIST are working towards generic polices and procedures for a common control security system [159].

4.5 Smart Building Use Case

Automating building management is another key M2M application aiming to implement more comfort, convenience and secure environment in both private residence and commercial buildings. A Smart Building system consists of devices that monitor and control technical systems in the building automatically, using two way communication. Furthermore, different physical and functional characteristics of a building covering geometry, spatial relationships, light analysis, geographic information, etc., could be collected and represented within a Building Information Modelling (BIM) as a data repository of the building. The need for robust characterization of energy

use in buildings has gained attention mainly due to the need to reduce energy and greenhouse gas, which is in line with the EU 2020 objectives [148].

Buildings are responsible for more than 40% of the global energy use [162]. Based on Siemens experience [163], the transparency of the energy flows and the actions dependent on them should enable up to 20% of energy cost to be saved. With integrating proper sensors and actuators, different objects and activities inside the building could be monitored and controlled. The objectives of such systems includes: improving energy utilization efficiency, reduce managements and energy costs, and enhance comfort and security levels for inhabitants. For example, lights and air conditions could be switched on/off based on room occupation, smoke sensors can be used to detect fire, etc. The data aggregated from various sensors to local M2M gateway, will help in monitoring activates inside building and trigger corresponding actions automatically, such as:

- Playing a dedicated audio alarm to alert the inhabitants in case of danger.
- Sending nomination to the emergency authorities including the event type and location.
- Acting upon pre-set actions like shutting down the gas and electric appliances, turning on electric light with accumulators.
- Following recommendations from the owner or the emergency authorities.

4.5.1 Requirements of Smart Building applications

Regarding the performance criteria of data throughput and latency, Smart Building applications have relaxed requirements. Since the control of HVAC (heating, ventilation and air conditioning) has to deal with high system inertia anyway [164]. However, the cost prospective is highly important, especially for private residence, to deliverer the requires performance at low system cost. Thousands of nodes might be installed to provide automation for a building, so every single node has to be as cheap as possible to make a sensible investment.

4.5.2 Challenges of Smart Building Deployment

The challenges of Smart Building deployment are driven mainly from the heterogeneity in connected devices and networking technologies. Wireless networking technologies are very popular in Smart building applications, as they offer distinctive advantages in terms of installation costs and flexibility in placing sensors where cabling is not appropriate for aesthetic, conservatory or safety reasons. However, the deployed technology must be tailored to the specific requirements of connected sensors and actuators to deliver these benefits at an attractive price to performance ratio. A lot of developments have emerged in this respect recently.

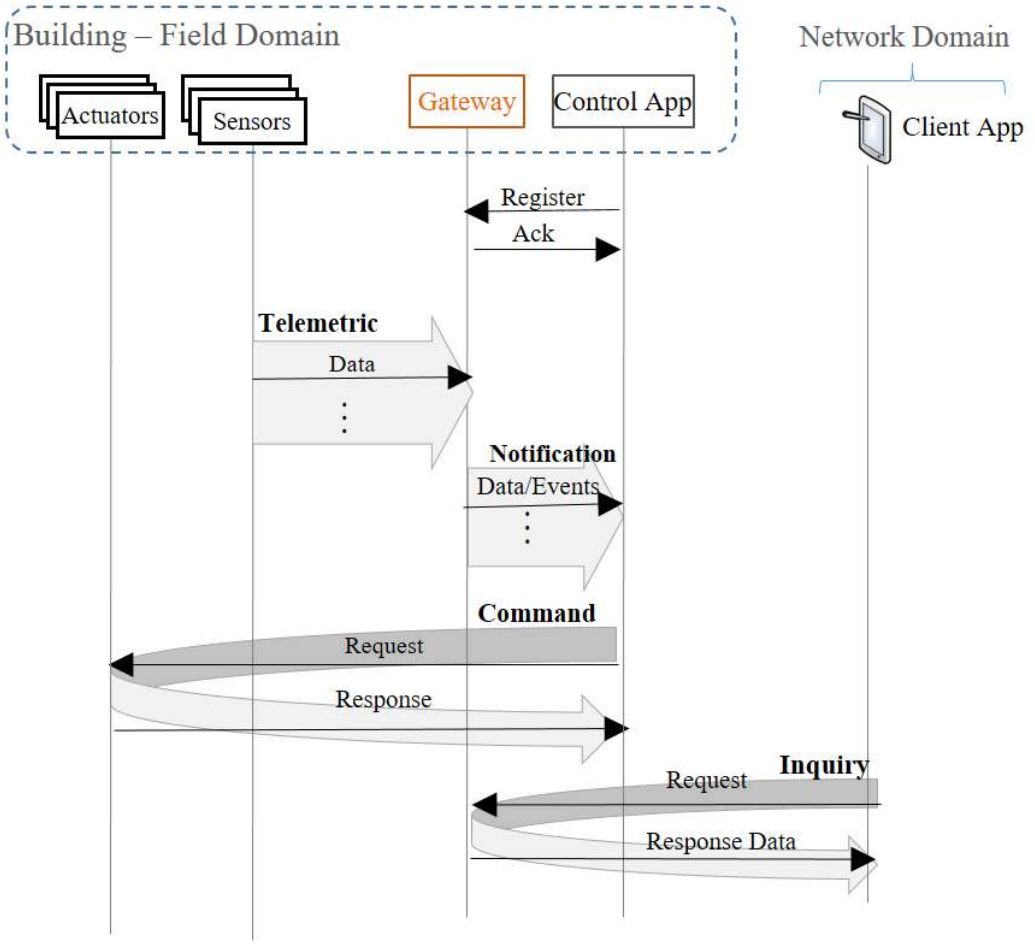


Figure 4.6: Smart Building Use Case

A number of technologies that fulfill the specific requirements of this class of M2M service have reached commercial status, including IP and non-IP based, but none of them is clearly in the lead [164]. Despite the fact that IP-based protocols requires large memory footprint, the IP-base approach for home automation is gaining more momentum in large-scale deployments in the context of IoT. This is due to the interoperability and auto-configuration supported features of IPv6 protocol [165]. A survey of wide range of Smart Grid standards, proposed for home and building automations systems, is presented in [166]. Among the existing standards for home communications, it is highly likely that those standards, which combine both wireless standards and the PLC, i.e. IEEE 802.11, IEEE 802.15.4 and IEEE P1901, will be widely deployed in the near future.

From the data interoperability perspective, more challenges are rising as the information exchange between architect, builder and maintenance organisations is still in early phase. Although the popularity of BIM is growing, more work is needed to adapt it to various applications [167].

4.6 Environment Monitoring Use Case

Although ICT has many positive effects on the environment by making industry and lifestyles more efficient, it has a negative effects as well caused by ICT devices and equipment, such as the consumption of energy and natural resources and generation of waste. Towards constructing a Smart City from an environmental viewpoint, the negative factors of ICT shall be minimized as much as possible. To this end, both the positive and negative environmental aspects of ICT shall be quantified to enable further actions of improvement. The ITU-T recommendation L.1400 provides general principles of methodologies for assessing the environmental impact of information and communication technologies [168].

Monitoring activities are used to detect changes in the environment for several purpose. It is expected that the majority of the world's population growth will be concentrated in urban areas in the next decades [169], therefore innovative management and monitoring systems are required to enhance the citizen's quality of life. Environmental data (temperature, humidity, air pollution, noise, etc.) are collected, stored and analyzed by Smart City Platforms and rendered to other applications. Few examples are presented in [29].

4.6.1 Requirements of Monitoring Services

Monitoring Services are mostly non-real time services. In some cases, data monitored from sensors might be not critical, and its transmission to a service platform can be simply delayed or even transferred few times a day without endangering the consistency and sense of the information. In these cases, one can think on opportunistic communication solutions to gather the data from the sensors when connectivity is available. Some examples of services useful for citizens with mild or no real-time requirements would be, for example, trash collection optimization, street furniture maintenance or environmental monitoring.

On the one hand, some places where devices are located there is no reliable network infrastructure available. Data can only be gathered from sensors by a moving gateway installed on a vehicle and sent through a M2M connection to the service platform. On the other hand, the sensors with restricted power capabilities shall be periodically switched between active/sleep modes to preserve the battery life. The collected data could be further analysed by Smart City Platforms and rendered to other applications; few examples are presented in [29].

4.6.2 Challenges

The main challenge in deployment of such service is the lack of a fixed/mobile infrastructure to gather data on some areas. Deploying an on-purpose infrastructure to collect and forward the information from the sensors through the city might be costly or infeasible. In these cases, providing ways for opportunistic networking would be the alternative.

4.7 Requirements Identification and Analysis

This section summarizes high-level system requirements that have been derived out from the analysis of the use cases presented in previous sections. Generally, current communication platforms are optimized for H2H communications, and require essential improvements to support the following properties of M2M applications:

- A new trend of traffic patterns and interactions that include transmitting burst short data from a great number of devices.
- The need to support multi-channels of communication.
- The ability of decoupling the diverse service domains from underlying access technologies.
- Propagation of large amounts of data and resources that need efficient management.
- The heterogeneous nature of connected objects and their capabilities.

Two major differences between M2M and H2H are worth mentioning in this regard. Firstly, the uplink to downlink ratio is much higher for M2M traffic compared to traditional H2H/H2M traffic. The downlink traffic of M2M application are generated much less frequently as it consist mainly of device configuration commands and requests. Still there are some examples of downlink traffic, such as firmware upgrade, change of charging tariff, etc. In the H2H communication the downlink traffic occupies the most bandwidth, which is not the case in M2M traffic. Secondly, the machines transmit on regular basis or on random intervals and their data consists of packets with short payload lengths. In contrast, H2H usually involves continuous flow of information.

4.7.1 Functional Requirements

The following functional requirements have been derived from the use cases and M2M applications presented previously. Functional requirements specify essential functions of a system and its components. The requirements are listed here in sort of their relativity to the dissertation objectives. In Table 4.3, a detailed functional requirements list is analyzed for the use cases presented previously.

Internet-based Connectivity: Providing ubiquitous computing and communications is the main objective of the IoT system, therefore supporting multiple protocols and sensor technologies is essential. The end-to-end IP architecture is widely accepted as the best alternative available to support the design of scalable and efficient networks comprised of a large numbers of communicating devices. IP enables interoperability at the network layer, but does not define a common application-layer standard. Consequently, it appears as the optimal base for use in a wide variety of applications ranging across several industries [90].

Supporting of Multiple-Protocols: Connections with multiple bearers could be established by an M2M node in order to supply different QoS flows or interoperability with different servers. Supporting generic communication technologies is important for ensuring multiple protocol binding such as HTTP and CoAP.

Adaptability: The system should support dynamically configuration on M2M nodes to adapt with application's traffic change.

Data Reporting: An M2M platform is not only connecting electronic devices by using communication networks, but also facilitating the data exchange between them in order to enable systems with Smart capability to realize information flow between connected objects. Different patterns of reporting should be supported, i.e. periodical, event-base and on-demand.

Group Communication Support: M2M devices could send and/or receive frequently or infrequently small amounts of data, some of them may be grouped into groups that is subject to the operation. Multicast and broadcast communication shall be supported.

Support of Device Management: New technologies are needed to facilitate the interaction between a decision making server and actuator clients to replace the SMS-based protocols, which are still used for controlling devices over legacy systems. A number of platform-independent device management protocols have been specified to be used in managing and configuring devices over standardized interfaces.

Application Management: Offering innovative services to control connected devices in an interoperable manner, raises various challenges in designing open and standardized service enablers for M2M. An abstraction layer for application development shall be supported. Today service providers are building an eco-system with 3rd party partners to offer new innovative services.

Mobility Support: Connecting a broad range of physical objects, including cars, robots, etc., requires mobility support in various levels within the M2M systems. This mobility should be accommodated at specific communication layers like MAC, network and session layers in order to ensure session continuity for mobile M2M nodes. In this regard, the user should be able to retrieve the desired sensing information or send command to specific actuator irrespective of the mobility of these objects.

Authentication and Authorization: M2M objects within the system should be authenticated using existing security mechanisms. Each request originated from an M2M node should be verified before resources are granted.

Table 4.3: Applicability of Functional Requirements to Surveyed Use Cases

Requirement	EHealth	Smart Energy	Smart Building	Environment Monitoring
IP-based	Required	Required	Required	Required
Support of multiple bearer	Required	Not required	Optional	Optional
Adaptability (Self-Configuration)	Required	Required	Optional	Optional
Data Reporting Model	Periodical, event-base and on-demand	Periodical, event-base and on-demand	Periodical and on-demand	Periodical
Group communication	Unicast	Multicast	Multicast or broadcast	Broadcast
Device Management Mechanism	Optional	Required	Optional	Optional
Application Management	Required	Required	Required	Required
Mobility support	Vertical and horizontal	Not required	Not required	Not required
Authentication and Authorization	Required	Required	Required	Not required

4.7.2 Non-functional Requirements

The non-functional requirements elaborate a performance characteristic of the system. Following are the non-functional requirements that have been derived from use cases presented above

Backward and Forward Compatibility: Compatibility with existing infrastructure is vital to IoT deployment. The system should support existing deployments as well as future standardized extensions.

Extensibility: Allowing the extension of the system by adding new components and functionalities with minimal maintenance.

Interoperability: M2M systems should allow the interoperability between various domains in Smart environments. This is a very important issue, as the different M2M nodes, i.e. gateways, core server and devices might come from different vendors, or use heterogeneous technologies, and they all have to be integrated in the global IoT.

Scalability: Scalability refers to the ability of the solution to support growing amount of work and the increasing demands for resources and services, such as numbers of connected entities, or transactions among system entities. Due to the rapid increase in the number of M2M connected objects coupled with heterogeneity in sensor networks, scalability is a main technical requirement for enabling ubiquitous access in large-scale deployment of M2M platforms.

Reliability: Several protocols have been integrated to the protocol stack to provide a reliable bi-directional communication over the best-effort transport medium that the Internet basically provide. These protocols incorporate error detection, retransmissions and flow control. For the IoT to merge seamlessly into the Internet, it needs to offer high reliability from end-to-end. This is very important as several applications having restrict requirements on the end-to-end delay of transmitting a packet.

Power Efficiency: A great amount of M2M objects powered by batteries are integrated in everyday life service. Consequently, finding the needed energy to power the processing and communication is a major challenge. Therefore, the utilized protocol stack should exhibit a low average power consumption.

Security and Privacy: M2M systems are extremely vulnerable to attacks as they are used in many sensitive sectors in home and industry. Furthermore, a big amount of the integrated components are characterize by low power and computation capabilities, and thus cannot implement complex security mechanisms. Authors in [54] categorize the main security vulnerabilities in M2M systems. Generally, mechanisms of secure connection establishment and sensitive data handling are required.

Considering the use cases discussed early in this Chapter, each one of the non-functional requirements listed above have different levels of importance. For example, interoperability is highly requested within the Smart City context to enhance the consolidation of vertical domains. Privacy is rather dispensable for environment monitoring applications, as the collected data are meant to be shared widely, however, it is important to support security mechanisms to protect the system topology from attacks. In Table 4.4, a summarized description of functional and non-functional requirements is listed.

Table 4.4: Summary of Functional and Non-Functional Requirements

	Requirement	Description
Functional	IP-based Connectivity	An end-to-end IP architecture is required to provide a scalable and efficient M2M system comprised of large numbers of communicating devices.
	Supporting of Multiple-Protocols	The system shall support multiple protocols and sensor technologies to enable the integration of wide range of devices.
	Adaptability	The capability of well-adaptation to different services and topologies.
	Data Reporting	The system shall allow connected nodes to perform data reporting in different patterns (e.g., periodical, on-demand or event-base).
	Group Communication	The system shall support the ability to multicast and broadcast data notifications or management commands.
	Support of Device Management	The system shall support at least one platform-independent DM protocol to enable managing and configuring devices over standardized interfaces.
	Application Management	Supporting an abstraction interface to effectively manage application's deployment.
	Mobility Support	The capability of allowing continuous service for a node to change its location (horizontal mobility) as well as a node to change the used network technology (vertical mobility).
	Authentication and Authorization	Mechanisms for authenticating and authorizing M2M devices/applications must be provided by the system.
	Accounting and Charging	Accounting and charging should be depend on an application, rather than on time duration or data usage.
Non-Functional	Backward/Forward Compatibility	The system should support existing deployments as well as future standardized extensions
	Extensibility	The easiness of extending system's capabilities with minimal maintenance.
	Interoperability	Supporting a certain level of interworking between various domains in the Smart City context.
	Scalability	The ability of the system to handle the increase in the number of requests or connected devices.
	Reliability	The ability of the system to fulfil applications' requirements.
	Power Efficiency	Mechanisms to optimize the energy consumption shall be deployed.
	Security and Privacy	Mechanisms for securing the connection and M2M devices/applications must be provided by the system.

5 Design and Specification of AdM2M

5.1 Introduction

This chapter presents the design and specification of the AdM2M framework introduced in this dissertation. Based on the fundamental M2M communication concepts and on the understanding of functional and non-functional requirements of M2M applications, presented in previous chapters, a new framework is designed enabling the adaptability of M2M nodes to applications characteristic and requirements.

As remarked in previous chapters, the M2M communication is foreseen to involve a large number of objects/entities with variant capabilities and functionalities. Moreover, each device might be engaged in one or more application domain. Recently, various protocols have been proposed to the M2M development aiming to address specific issues of M2M communication, such as the requirements of integrating resource-constrained devices and supporting ubiquitous sensing service. However, there is no sophisticated guidelines for the protocol stack deployment for M2M applications in the literature. Furthermore, most of existing M2M implementations and solutions lack the interoperability feature as they have been built in a highly vertical fashion using proprietary technologies. Thus, the integration of any-Thing into a large-scale system is difficult to implement and as a result, they form multiple Intra-net of Things [11] instead of a global Internet of Things (IoT).

In Chapter 3, the current state-of-the-art in M2M service capability platforms

is discussed to answer the question of how standardization efforts can address the requirements towards realizing a reliable and secure architecture for M2M services. The oneM2M specifications [80] is considered as the basis of the AdM2M solution, as it is the international standards for M2M communications and IoT that provides a global service layer standard.

In Chapter 4, the requirements driven from the service's specification and deployment's challenges was presented. From an operational point of view, each use domain has its special work model, stakeholders and technical requirements that impact the design of the end solution. However, within a typical Smart city system, interoperability of integrated systems and components is mandatory. Therefore, an application-agnostic M2M middleware that implements the core service layer is required to enable the large scale Smart City deployment.

5.2 Design Considerations

This section provides the fundamental considerations taken into account on the scope of this work. These considerations place particular emphasis on ubiquitous sensing coordination.

With reference to the characteristics of M2M systems discussed in Section 2.6, the following key assumption on the involved M2M environment are made:

1. The connected M2M nodes will include both resource-constrained and resource-rich devices. Consequently, their computation capabilities and storage size are disparate.

2. The connected M2M nodes are autonomously involved in the system. Some might be located in hard to access locations, others are supposed to work with limited human intervention.

3. An M2M device/gateway could integrate several sensors and/or run more than one application at the same time.

4. An M2M application might be registered to more than one platform server.

Figure 5.1 illustrates the basic parts of the design principles. The main objective of the AdM2M framework is to support the adaptability of M2M nodes to application's requirements, and different interactions patterns. The AdM2M utilizes several resources that are provided by the modules and functional entities of the hosting M2M Core middleware. These resources include data handling, application enablements, device management, and authentication and authorization mechanisms. The underling basis of the framework and core resources shall be inline with M2M standard architecture. The architecture of oneM2M is adopted as the basis of the AdM2M framework. It is anticipated that the oneM2M standard will become globally prevalent due to its international member partners.

Further features are introduced by AdM2M to the system that addresses the need to handle heterogeneous traffic patterns produced by different entities within

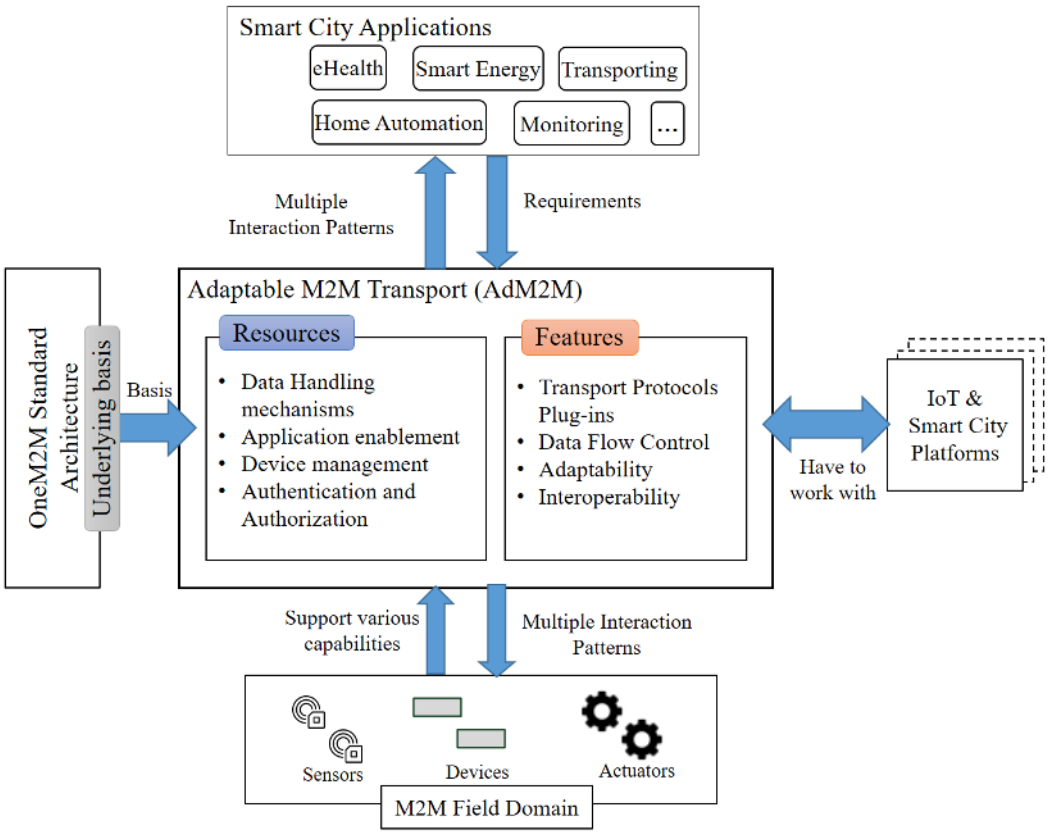


Figure 5.1: Design Principles for The Adaptable M2M Transport (AdM2M) Framework

an M2M system. The dissimilarity in resources and capabilities of the connected objects as well as applications' objectives has its clear impact on the overall system. Thus, it influence the design process. The extendability of the system is supported by adding new protocol libraries and binding functionality plug-ins.

By examining the communication requirements driven from the service's specification and deployment's challenges, discussed in Chapter 4, and the variety in M2M objects capabilities, the following major aspects are considered when designing the AdM2M framework:

Internet-based Connectivity: The IP-based Internet is the largest network worldwide, and there are great efforts from standardization bodies and research groups to connect WSNs into the Internet.

Standard Conformance: As previously discussed in Chapter 3, there are a number of bodies involved in the standardization of the M2M architecture and technologies. The standard conformance of the proposed solution is an important requirement in support the interoperation between different vendors. To

this end, the architecture of oneM2M is adopted as the basis of the AdM2M framework to confirm a degree of compliance with widely accepted industry standards.

Adaptability: The main approach of AdM2M is to enhance the ability of M2M nodes to adapt to operation circumstances.

Scalability: The M2M architecture will see a regular increment of connected objects and deployment of new services, resulting in a constant change in the system environment. Any extension in the M2M architecture should consider scalability issues to support the growth in objects.

Interoperability: The AdM2M shall fulfill the Interoperability requirement, to address the heterogeneity of M2M nodes.

Extensibility: The future growth of M2M communication protocols and technologies is taken into account, in order to enable supporting more functionalities such as interoperability issues, operational problems, or security vulnerabilities.

Reliability: Generally, the reliability is a core aspect of communication systems that could be tackled at multiple levels. The system needs to be able to cope with different environments and topologies.

5.3 High-level Architecture

The High-Level Architecture of M2M system discussed in Section 1.3 and in Section 2.3 is adapted to define the detailed architecture of the proposed framework. M2M middleware platforms are required as a key enabler of Smart City implementations, as it facilitates the collaboration of various stakeholders, increases the efficiency of administrative services, and assists in developing environment friendly applications. Consequently, it is expected that variant applications will be deployed on M2M entities and generate heterogeneous traffic patterns. In this section, we specify an adaptable framework to handle heterogeneous traffic patterns within an M2M system and support a reliable data delivery at the same time. Figure 5.2 depicts the High-level architecture of AdM2M framework within an M2M service middleware. By definition, the AdM2M Framework is considered to be located in a Middle Node (MN) e.g., a gateway, or in an Infrastructure Node (IN) e.g., backend server, within an ETSI/oneM2M compatible system to enhance the communication facility of the ETSI SCL/oneM2M CSE.

Today service providers are opening their core infrastructure to 3rd party developer through open APIs, giving the opportunity to connect thousands of developers with millions of users [170]. In result, they gain new revenues from mediating the application delivery, and reduce the time of developing new services. The same concept is placeable in M2M platforms with a set of application enablement APIs

to support the development of application by 3rd party. In the context of the roles of ETSI mIa interface and oneM2M mca interface, the designed APIs have been divided into three categories: Network, Device, and Data APIs [31]. These APIs intend to provide a communication channel for applications at higher layers with the Infrastructure Node (IN) or the Middle Node (MN) in the M2M system.

The prime functionalities of an infrastructure/middle M2M node, as specified by the oneM2M CSEs, are provided by the core middleware. Section 6.2.2 describes the implemented capabilities within the prototype system. These functionalities are inline with the oneM2M common service functions listed in Table 3.3.

An internal bus supports the inter-communication between the component by sharing an inner-API based on events, where an internal subscribe/notify mechanism enables internal components and plug-ins to be notified about updated status. The AdM2M framework provides different transport protocol stacks dynamically, with using HTTP and CoAP as plug-ins to support the communication with M2M devices using the proper stack in each use case. The framework is extendable to add more protocol libraries and plug-ins.

The Req/Res paradigm is widely used for information exchange in the Internet. However, more protocols are emerging that adapts the Pub/Sub paradigm due to the advantage of supporting asynchronous and one-to-many messages distribution. In subsection 3.3, a number of protocols commonly used in M2M systems are reviewed. In most telecommunication networks, transactions happen between several components or applications in a session based communication manner, as all these components are always connected to the network and in on-state. The case is different in M2M networks, which connect huge number of devices with limited capabilities (storages) and are more likely to face energy shortages. Such M2M devices might not be able to stay connected all the time, and thus it can't interact immediately to all transaction with other component in the network, this situation will cause cancellation of the transaction and effects in all parties. The REST based architecture provides a good solution to this problem in M2M networks, by handling transactions in a resource-based communication. REST is based on the concept of resources addressed by a URI, thus it provides to M2M devices a place to store their states and data, which is always on and connectable [31]. The main idea is to let transactions occur in a subscription/notification manner. Applications willing to start a transaction (e.g. getting data from remote device) subscribe in the SCL in ETSI M2M systems or CSE in oneM2M system to a specific resource, and when the data is ready by the SCL/CSE, it sends notification to the component that started the transaction i.e. had previously subscribed to that resource. Figure 5.3 presents part of the M2M resource tree, which is considered in our platform. Each entity in the M2M system (i.e. gateway, application, or device) is presented by a uniquely addressable resource in the hierarchical tree. For example, each gateway is presented by one *<scl>* resource, while devices are presented by the *attacheDevice* resource resides under the *scl* resource presenting the hosting gateway. Applications are presented by the *<application>* resource that includes *containers* resources to categories different content instances under it. The data content is presented by the

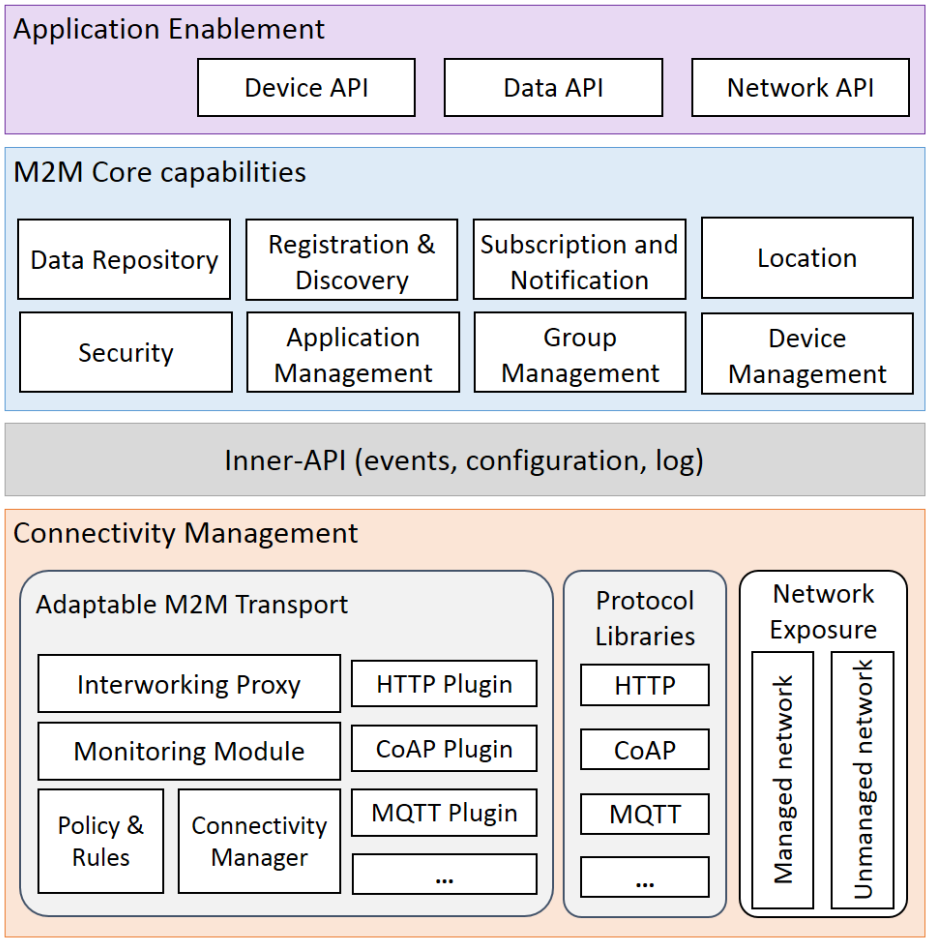


Figure 5.2: The Adaptable M2M Transport Framework Architecture

contentInstance resource. Finally, a subscription to any of these resources is presented by one *subscription* resource that resides under the corresponding resource.

Furthermore, the Pub/Sub architecture is presenting a dynamic and loose coupling network paradigm to deliver sensor driven processed data to subscribers, facilitating exchange between sensor networks and cloud-based networks. The Pub/Sub architecture is also implementing as an asynchronous and multi-point communication mechanism. It makes entirely decoupling communication among publishers and subscribers on time, space and control flow [171]. The content-based Pub/Sub model to deliver sensor driven processed data to subscribers, facilitates exchange between sensor networks and IoT platforms. It not only meets outstandingly requirements of loose communication in large scale distributed system, but also supplies a need of sharing efficiently data in multi-distributed system in the same network.

Figure 5.4 depicts the mapping of the data and requests routing in an ideal M2M system for the general interaction modules shown in Figure 2.4. Usually, sensors are

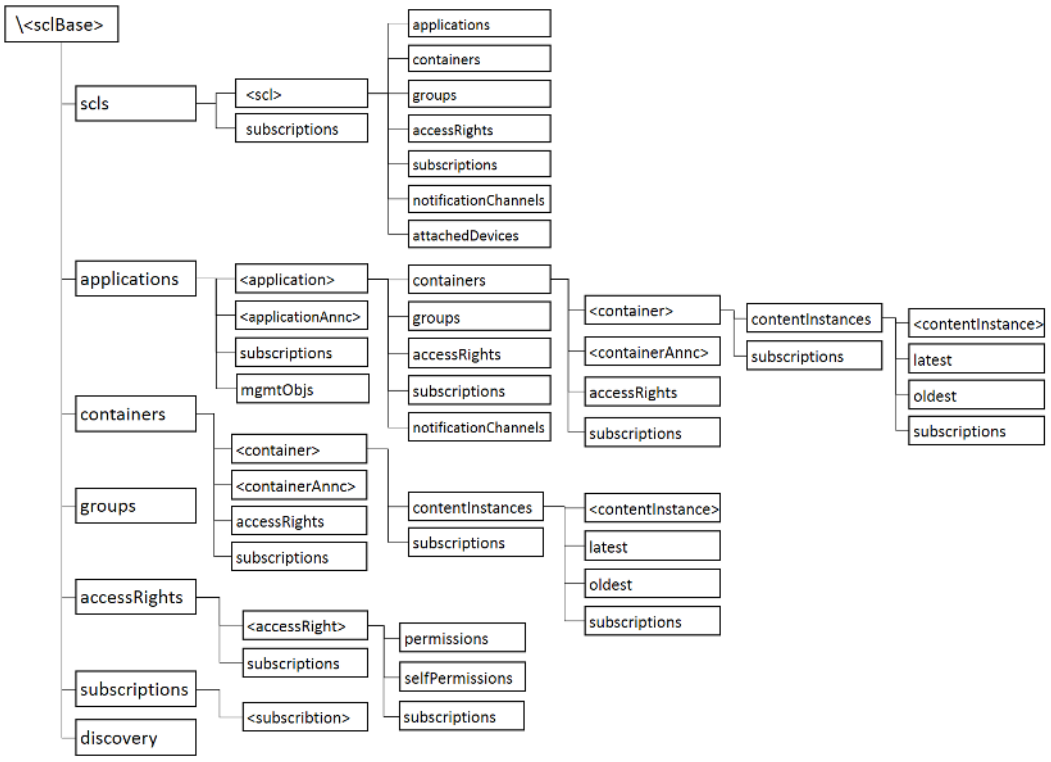


Figure 5.3: M2M Resource Tree based on ETSI/oneM2M specifications

pushing their data in a telemetry interaction model toward the front-end gateway via a data adapter, a response is returned to acknowledge the delivery of the data to the adapter. Additionally, the re-targeting step is performed to announce the availability of updated data to the Back-end platform, thus Network Applications (NA) connected to the Internet could access this data. There are two interaction modules to be used by NAs to receive the data. First option is using the inquire module, where the NA send a request to receive the data of a specific sensor/device and receive in response the requested data. The second module is based on the notification mechanisms, where the NA send a request to subscribe for the data and get notified later whenever the data is available. The later method is more suitable for retrieving data that are generated in event-bases. With both models, the NA could specify a set of filter criteria within the request in order to narrow the notification messages to the needed amount and optimize the communication channel usage.

Due to the fact that M2M workflow is based on data acquisition, decision making and controlling (Figure 2.1), a command sending interaction module is needed to send commands from the server to connected devices. Commands are issued from a decision making service that could be deployed on an NA, which executes a logic for data analysis using the data aggregated from sensors and concludes the action(s) needed to optimize and improve the provided service. The command request will be

then forwarded to the devices via the hosting gateway. Additionally, the command could be issued by the hosting gateway itself based on a previously defined policy.

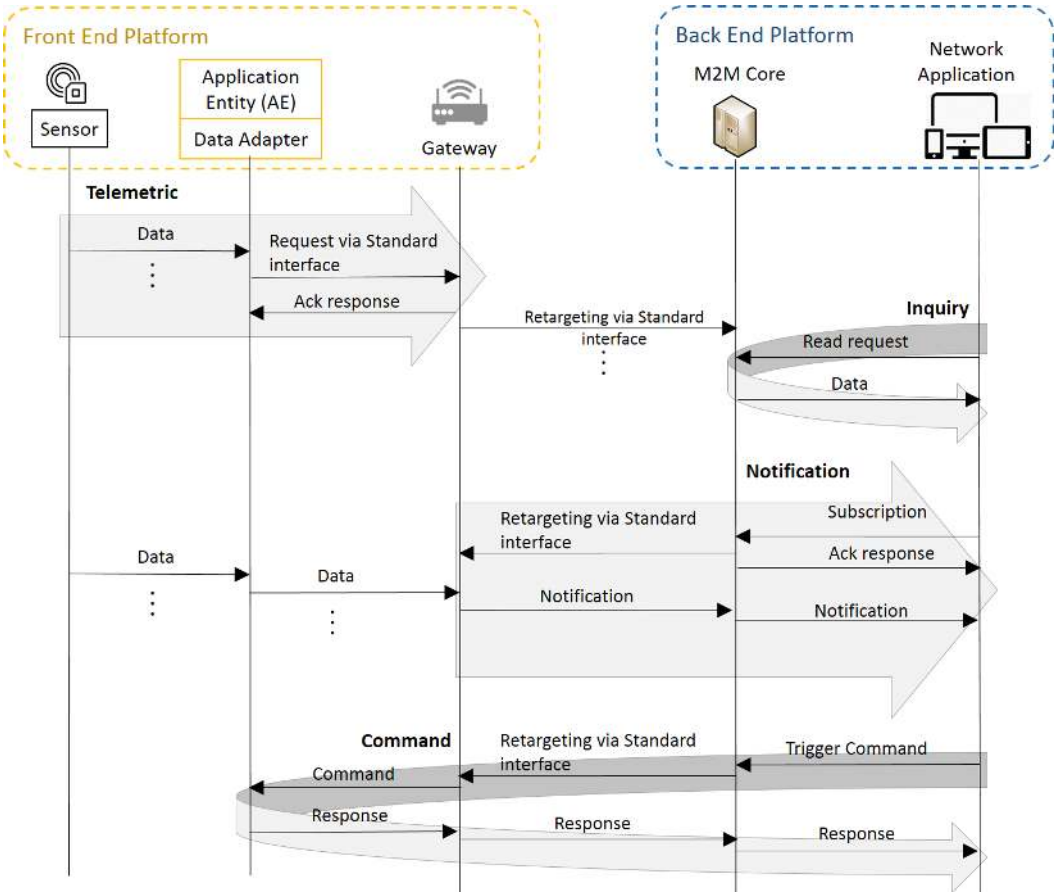


Figure 5.4: Overview of Request Routing for M2M Interactions

5.4 Functional Entities

5.4.1 Communication Selection Module

Usually M2M devices, such as smart phones or portable devices, have multiple connectivity capabilities spanning over new access networks and also allow various application installations. Applications running on an M2M node (device or gateway) might be aware of the current allocated IP connection on the node, but they are usually decoupled from the connectivity and mobility management components for security reasons. As application requirements might collide in terms of QoS requirements and data interactions, e.g., some require frequent real-time data exchange while others perform event-based data transmission, it is useful to enable multiple connectivity options to meet different needs. Nevertheless, the dynamic

adaptability of configuration settings in terms of transport management is required, in order to minimize the human intervention in an M2M solution and increase service reliability. Two approaches of dynamic adaptability on M2M nodes could be developed. One is application-oriented; the M2M node applies provisioning policy based on the Application ID. The second is traffic-oriented; the M2M node adapts to current application's traffic and IP connectivity parameters. The second approach is more suitable for M2M systems, due to the fact that some applications might require the support of multiple interaction models (Table 4.3).

In the case of an M2M gateway serving multiple sensors/devices, it should adapt to the heterogeneous traffic generated from both upstream and downstream, and engage into strategies to optimize the usage of available communication channels and energy. One option is to configure provision policies based on the application identity (i.e., originator of the request), to utilize the transport protocol that performs better with the expected traffic pattern of the application. The concept is based on a connectivity management module used for handling heterogeneous Access Networks, as depicted in Figure 5.5 [39]. The AdM2M framework consists of a Monitoring Module that constantly collects a set of parameters related to the data stream generated by registered ASN, i.e. rate and data size. The data are presented in constanceInstance resources created in response to application's POST requests, as specified by oneM2M specifications. A Policy and Rules component is necessary to store the provisioned or learned policies and rules, while the Connectivity Manager operations enable and disable interfaces and plug-ins depending on the operating system (Android or Linux) APIs.

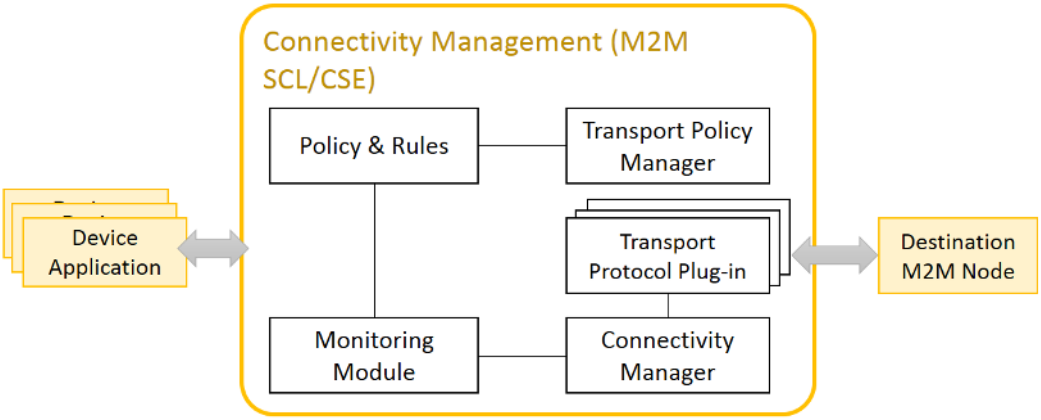


Figure 5.5: Transport Policy Concept

Based on the application ID (originator of the request), the M2M gateway either has provisioned policies stating the protocol to be used or learn what is the most appropriate one and which parameters to use. For a normal HTTP transport the connection time can be learned in order to meet the traffic pattern of the application, such as sending multiple chunks of data at a specific hour and for another time only

a few. For example, in case of air quality monitoring use case, at night the interest on pollution monitoring is decreased and the application might be instructed to aggregate the data and send it less often than during the day.

When an ASN subscribes to receive notifications related to a new content or updated information on the front-end server or back-end server, it could specify a set of criteria to receive notifications in specific conditions only. Furthermore, OneM2M define more attributes to enable the ASN to control the rate of notifications to receive, however no specification is provided on how to implement this. These attributes, the rateLimit and batchNotify, are used by the AdM2M to select a proper connectivity channel that matches the capability of application nodes and the size of notification payload [46]. The rateLimit attribute is used by the subscriber application to limit the notification rate, while batchNotify attribute is used to enable the batching of notifications. The HTTP protocol will be selected for sending batched notification of big sized. For small high rate data packets, CoAP will be selected. The thresholds of the rates and packet size will be continually updated by the Monitoring Model. Figure 5.6 shows the process diagram of the notification mechanism.

Requests transmitted from the M2M field domain (ASNs or MNs) to the back-end network domain can use one of various channels defined by the Access Network providers. The retargeting mechanism is used to forward data from an M2M Gateway Application (GA) to other M2M nodes. An evaluation process is executed to provide a decision on selecting which protocol to use in retargeting requests or forwarding notifications. The AdM2M implements an algorithm to build rules defining the transmission capability of the served devices in order to optimize the message exchange. This is controlled by either the originating or destination application's parameters used as a key of the learned rule: identity, IP or a Fully Qualified Domain Name (FQDN) address. Using the IP address would be recommended when related to the served nodes.

M2M nodes will be characterized with variant sensing, actuating, processing and communication capabilities. In term of the communication capabilities, some devices will have the ability to connect directly to the Internet via cellular networks 2G/3G, LTE and 5G, while others will connect through a MN, e.g., a gateway, forming a Local Area Network (LAN). In the latter case, radio technologies such as ZigBee, WiFi, or Bluetooth will be used to access the MN that might use cellular network to connect to the Internet. Regardless of the access technology in use by the M2M nodes, a reliable path between data-producer nodes and time-consuming nodes should be realized.

The messaging technologies discussed in Section 3.3 can be used by M2M nodes in different domains to interconnect them in a large-scale IoT via a range of wired and wireless communication technologies. As depicted in Figure 5.7 each node could include the deployment of one or more of these technologies depending on the node's capability, use-case requirements and operating environment. In terms of reliable delivery of data and events notifications, three levels of communications could be analyzed:

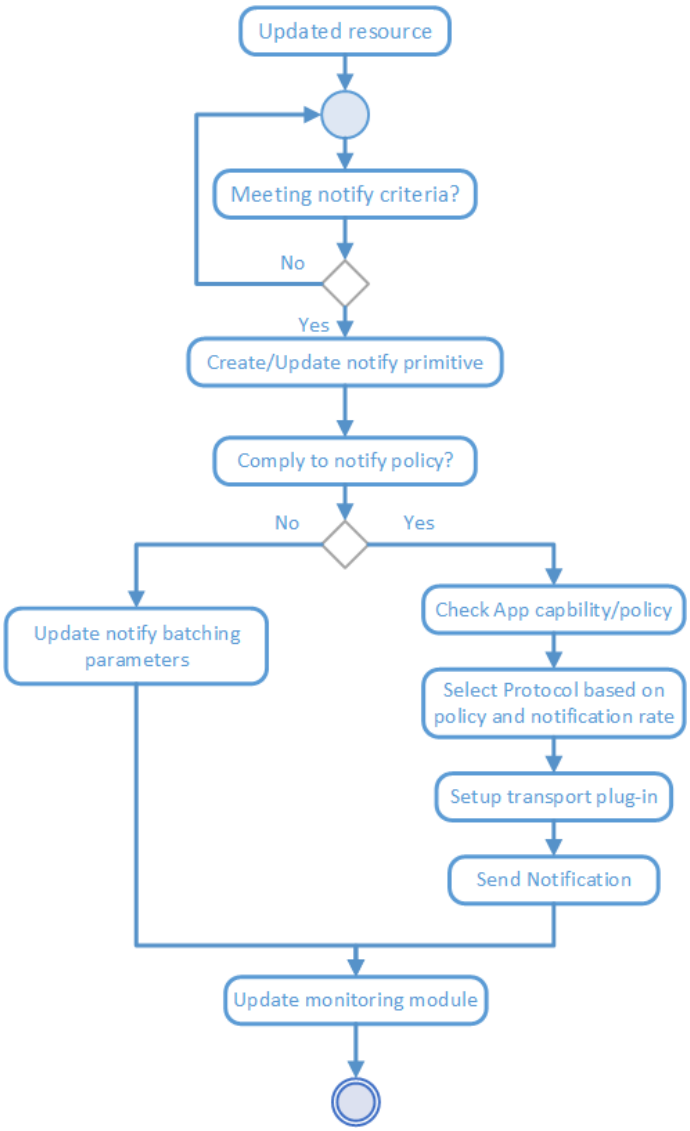


Figure 5.6: The Process Diagram of Notification Mechanism

- Inter-Device communication: where messages exchange between device nodes connected over LAN, e.g., between a device/sensor and the gateway.
- Intra-domain communication: message exchanges between two nodes connected via a gateway.
- Inter-domain communication: message exchanges between gateway and the backend server over the Internet.
- End-to-end communication: messages exchanges between data producers and data-consumer in an IoT use case.

Each messaging protocol addressed by this dissertation is suited for addressing one, or all of the connectivity levels identified above and illustrated in Figure 5.7.

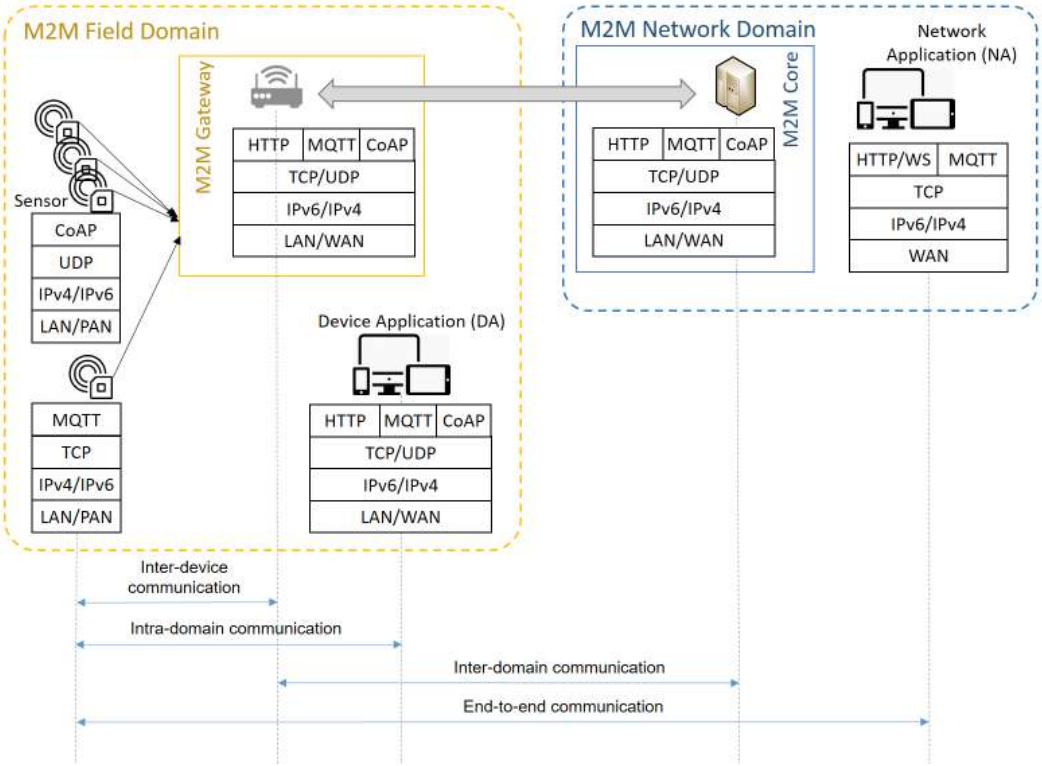


Figure 5.7: Protocol Stack within The System Architecture

5.4.2 Multiple Data Flow

The state of the art procedure of data collection and transmission involves reading datum from the sensor, processing it to a suitable format and creating a delivery request holding the datum to be sent towards the middle/infrastructure node. The final size of the datum unit depends on data encoding, encryption, local communication path selection policies. Moreover, it could be negotiated as in the case of the blockwise transfer in CoAP [64]. This negotiation procedure is repeated every time a measurement is taking place, thus leading to an inefficient data transmission as the application is not aware of the optimum data size to be used for transmission. Although the CMDH function, briefly described in Table 3.3, can use policies to select a communication path, it is not aware of the data sampling rate or its priority to be sent. Making the CMDH aware of all the possible data formats would make the transmission complexity impossible to manage.

In [40], we propose a new class of resources named BufferResource that can store each datum from each registered application temporarily. The BufferResource

is designed to accept chunks of data with different standardized priority class and transmission delay tolerance or other criteria like synchronization between chunks (useful in real time flow synchronization). The data chunks will be organized into child resources of the BufferResource. The child resources will have attributes containing the delay tolerance, the priority, and the destination Resource to which the data should be sent. The concept is depicted in Figure 5.8.

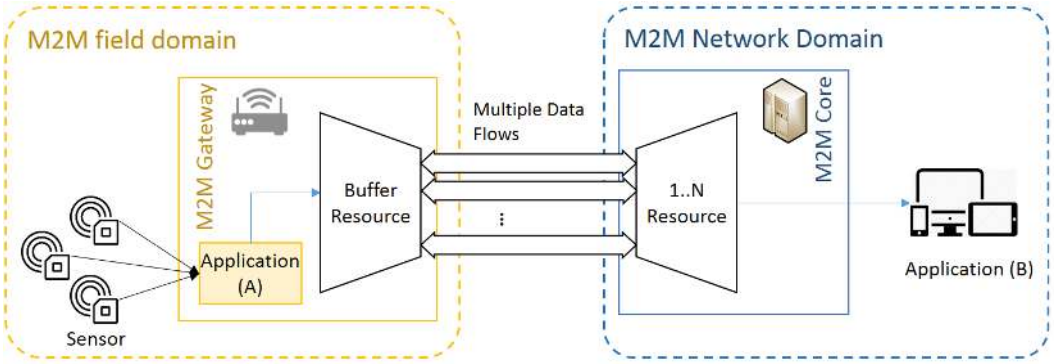


Figure 5.8: Distribution of BufferResource in Multiple Data Flows

In order to enable long term information exchange between the communication modules of the M2M Gateway and M2M Server endpoints, the application starts a transmission session in the communication module. The application can indicate at session establishment the initial transmission criteria of the data to be sent. During session establishment, the application will refer to the Buffer Resource and the communication module will monitor it, being aware of the whole available data to be sent. The communication module will organize the data into multiple data flows according to their priority. The flows will only become active if instructed from the application by indicating transmission criteria characterizing the data to be sent. In case the user or the back end application would request more details, the front-end application will indicate to use another minimum priority of the data. This might trigger the communication module to use another Access Network characterized as delay tolerant or with lower expense to send the data with lower priority in separate flows. If new enabled flows contain considerable amount of data and the current communication path would become congested or too costly to accommodate the new requested data, then another communication path might be used. This ensures that data with higher priority can reach the target on time and the lower priority data containing details would still reach the target with some delay.

5.4.3 Platforms Interworking Proxy

Several M2M solutions have been developed to serve a specific business application, which resulted in vertical silos of proprietary technical solutions. This has motivated the standardization work toward large scale M2M architecture to meet the expec-

tations of new business and revenue opportunities while reducing maintenance and resource costs. However, with the many initiation ongoing world-wide, interworking proxies become essential to allow the seamless interoperability between in-compliant devices and technologies.

IEEE defines interoperability as: *"The ability of two or more systems or components to exchange information and to use the information that has been exchanged."* [17]. The Heterogeneity of integrated devices and application is one of the main characteristic of M2M communication systems, taking in consideration the huge amount of connected devices to the Internet. Therefore, it is likely that interoperability enablers will be needed to overcome the challenge of exchanging information between all connected devices, as well as to design a system that efficiently addresses the specific needs of the target environment.

Different levels of interoperability have been defined in literature. A substantial classification is given by the Levels of Conceptual Interoperability Model (LCIM) [172], which specified 7 levels, starting from level 0 (No Interoperability) to level 6 (Conceptual Interoperability). As shown in Figure 5.9, the provided capabilities increases at each level. Levels 1 and 2 address the interoperability based on connectivity oriented concepts, levels 3 and 4 enables the data-centric interoperability, while levels 5 and 6 reach the conceptual interoperability to support composable services.

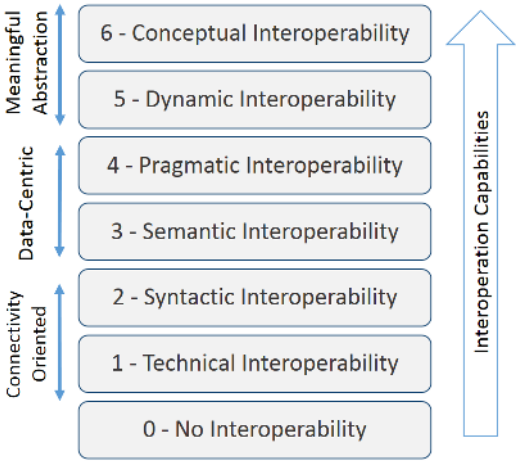


Figure 5.9: Levels of Conceptual Interoperability Model (LCIM), adapted from [172]

In the work presented here, we aim to achieve the level of Syntactic Interoperability (level 2) between standard-based M2M platforms. M2M platforms are designed to enable multiple applications to interact with single or multiple devices, therefore, any shared data or information could be used in various manners depending on the application's object and logic. Consequently, the semantic interoperability is the highest level to be reached between M2M platforms. Thus, the interoperated systems could share the meaning of the data, which will promote deployment of the Smart city framework. However, the usage of these data could vary depending on

the application itself.

The interworking proxy works as a mediation gateway that performs two main functions. First it maps the operation and protocol requests between interconnected platforms, and the second mediates the data model used by the interconnected platforms, as shown in Figure 5.10.

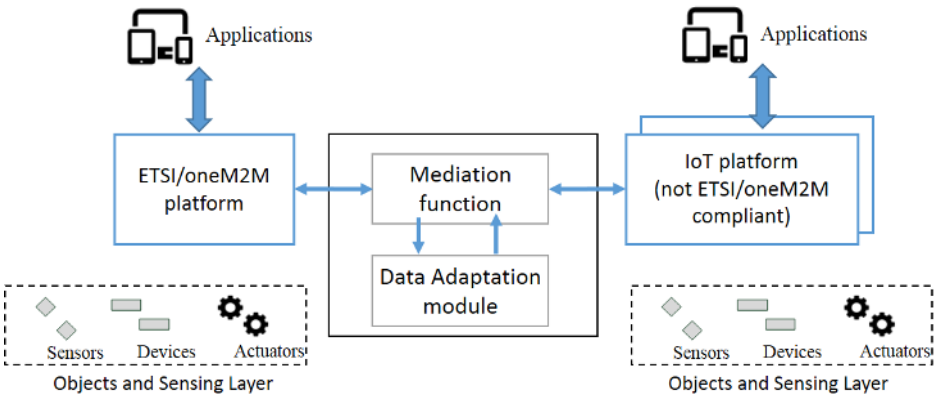


Figure 5.10: Interworking M2M platforms to Enable Large Scale IoT Implementations

By reference to the classical Network Interworking Proxy (NIP), defined by ESTI specification [50], the AdM2M Framework includes the design of a NIP, that supports the seamless integration of M2M platforms within the Universities for Future Internet (UNIFI) testbed [45, 33]. The NIP shall provide interoperability between non-ETSI compliant devices or gateways and an ETSI compliant M2M server (NSCL). The implementation of the NIP was developed as part of the UNIFI collaboration activities between TUB and Chulalongkorn University-Thailand. As an example of a non-ETSI compliant M2M system, the IEEE1888 M2M standards introduced in Section3.2.5 were considered. The same design could be easily extended to be integrated within an OneM2M INs as well as to support the interoperability with different M2M platforms.

The UNIFI project aims to address design, requirements and challenges of implementing unified Smart City Communication platforms. The project's approach is based on building sustainable teaching and research infrastructures in the areas of Future Internet and Smart Cities through global collaboration among academic institutions. UNIFI partners from Chile, Vietnam, South Africa and Thailand have developed different research activities in collaboration with TUB in Germany. The federation of seamless Labs that support Smart City prototyping is another important goal of UNIFI [173]. The existence of distributed and seamless architectures for communicational platforms that support Smart Cities prototypes at the Universities joining the UNIFI project, motivates the work to design and implement the platform Interworking proxy.

6 AdM2M Framework Implementation

6.1 Introduction

Following the AdM2M framework specification described in Chapter 5, this chapter describes how the specification was implemented in a prototype solution as part of the OpenMTC platform and related research project.

First the background implementation of the hosted M2M communication platform is presented and the overall structure is discussed. Afterwards, the main implementation parts are explained.

6.2 Implementation Background (OpenMTC Platform)

The OpenMTC platform [41] is a prototype implementation of ETSI/oneM2M standard specification for M2M service middleware. It has been designed to act as a horizontal convergence layer supporting multiple vertical application domains, such as transport, utilities, automotive, eHealth, etc., which may be deployed independently or as part of a common platform [42]. OpenMTC features are aligned with ETSI M2M Rel. 1 specifications [50, 174] and oneM2M Rel. 1 specifications [80], providing an implementation of ETSI SCLs/oneM2M CSEs at both Front-end gateway (i.e. GSCL or MN-CSE) and Backend server (i.e. NSCL or IN-CSE) of an

M2M architecture.

As illustrated in Figure 6.1, the OpenMTC platform includes a connectivity management layer that controls the interaction between the front-end and back-end over unmanaged access, as well as over managed Access Networks. This layer is responsible for the communication establishment and messages relaying with other M2M nodes (i.e. devices or gateways). It could be also extended to include more sophisticated functionality to enhance the communication capabilities.

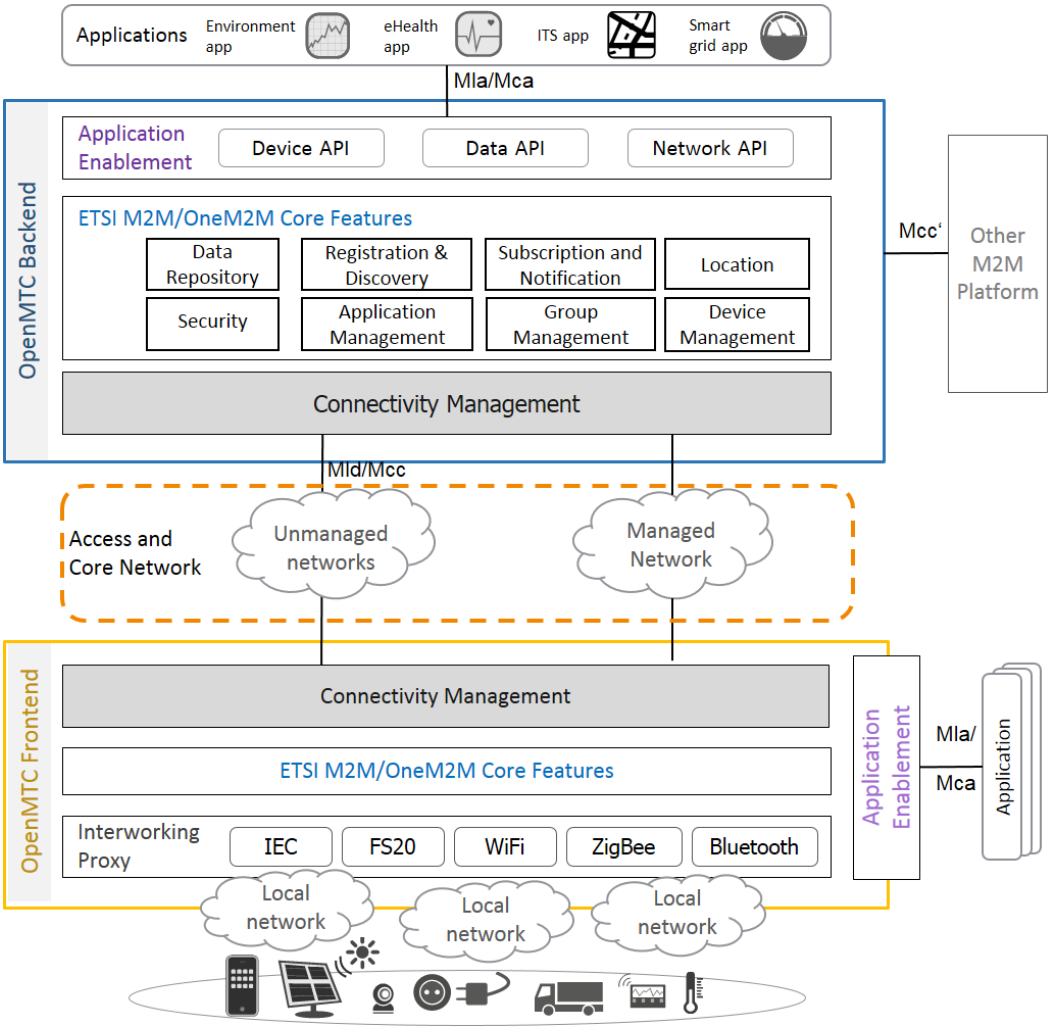


Figure 6.1: The OpenMTC Platform Architecture

Keeping in mind the diversity in computing capabilities of connected devices, OpenMTC components are implemented with the support of various hardware platforms, such as Android platform for mobile devices and Arduino platform for constrained devices. On the one hand, Android smartphones could be utilized as M2M gateways for sensors and devices connected to the Personal Area Network (PAN), such as eHealth sensors. On the other hand, Arduino provides a light platform for

power constrained devices installed for long range monitoring and controlling usage, such as home automation to remote control / remote measure. OpenMTC can be deployed in such devices and act as DSCL supporting their special requirements and capabilities. While OpenMTC can be deployed on multiple Unix based systems, most of this implementation were realized using Ubuntu GNU/Linux distribution (12.04 LTS). Different development packages are available with good support for programming. It offers important configurations possibilities on every operating system aspect. The LTS version of Ubuntu has been stable for the past years, fulfilling the requirement of a stable system.

6.2.1 Platform Specific Packages

Implementing the OpenMTC platform and the set of functionalities defined in the previous chapter was done using three programming languages: Python, Java and C. While the majority of the code is written in Python, some functionalities and libraries are accomplished with the remaining two. Python is a widely used general-purpose high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in other languages such as C++ or Java. Java is an object-oriented programming language, that promised the "Write Once, Run Anywhere" (WORA) concept by providing no-cost run-times on popular platforms. The OpenMTC repository structure is listed below. It is split in several folders to share some code base and also have a special code for the specific component and applications.

```
/
├── apps
├── common
│   ├── openmtc
│   │   ├── lib
│   │   └── src
│   ├── openmtc-etsi
│   │   ├── doc
│   │   └── src
│   └── openmtc-onem2m
│       ├── doc
│       └── src
├── doc
├── dscl-emulator
│   ├── device-capability-scripts
│   └── generated-config
├── futile
├── legacy
├── openmtc-android
├── openmtc-app
└── openmtc-gevent
```

```
openmtc-java
openmtc-GIP
    OpenMtcSdk
    OpenMtcSdkAndroid
openmtc-js
openmtc-openxsp
projects
    openmtc-fistar
    openmtc-trescimo
serializers
    openmtc-etsi-bson
    openmtc-etsi-msgpack
    openmtc-onem2m-xml
server
    openmtc-cse
    openmtc-ngsi
    openmtc-scl
    openmtc-server
    openmtc-unified
tests
util
```

The ***common*** package contains libraries used commonly by the platform such as CoAP and LWM2M protocol libraries. Also, the XML Schema Definition (XSD) files given with the standard specifications for both ETSI M2M and oneM2M are placed here. These files define the object's structure as well as the various types of Request-Indication objects and Response-Confirmation objects. All these objects adhere to the functional architecture specified in [174, 175].

Two underlying platforms are used to manage events in the openMTC system: ***GEvent***, and ***OpenXSP***. Both are event based libraries providing asynchronous I/O API that can scale its number of execution units according to the processing load. However, the former is often used in the Front-End for its speed and locality as it is targeted for constrained devices, while the latter is an event platform that provides a solution adapted to cloud deployment.

The ***server*** package contains the core functionality components as defined by the standards to implement the back-end and front-end servers. The libraries and structure defined in previous packages are used by this implementation.

For specific implementation related to Android OS, the ***Openmtc-android*** includes configuration files specific to mobile device's deployment of the platform; this includes the python-for-android library (https://code.google.com/p/python-for-android/), which includes a set of tools allowing python programs to run on mobile devices. In addition, a Graphical User Interface (GUI) is developed to present the aggregated data on a web interface based on users preferences, i.e. from one or more gateways/sensors, as shown in Figure 6.2.

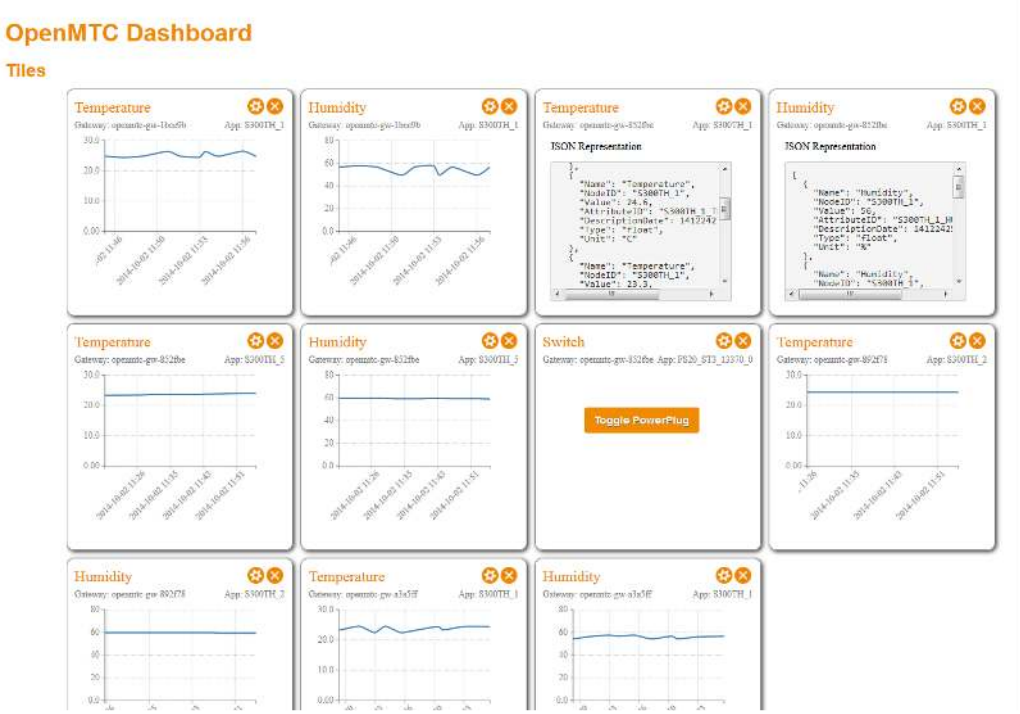


Figure 6.2: OpenMTC Dashboard Interface

6.2.2 Core Service Capabilities

Following description of the main capabilities supported by the OpenMTC platform:

Register and Discovery: Both ETSI and oneM2M standards have adopted the RESTful architecture style in their specifications. This style governs how M2M Applications (xA) and/or xSCL are exchanging information with each other. This information is presented by uniquely addressable resources, each represents an entity in the system, such as application or SCL, or operation related information e.g., access rights and subscriptions. Figure 5.3 shows part of the M2M resource tree, which is considered in OpenMTC platform. Each resource has a representation that shall be transferred and manipulated with CRUD verbs (Create, Retrieve, Update and Delete). The operations of the CRUD request and response primitives are implemented as described in [174, 175]. The discovery functionality allows searching for applications and devices based on information provided as part of associated resources and attributes. The discovery request shall be sent as a retrieve request. The obtained result depends upon the filter criteria and is subject to access control policy. The originator of the discovery request could be an AE or another SCL/CSE.

Data Repository: The main object of this module is to store information related to entities in the M2M system, i.e. applications, SCLs/ CSEs, and devices,

as well as the data pushed from connected sensors/devices. To persistently store data of any kind, OpenMTC internally uses a database abstraction layer. By using different database adapters, OpenMTC can be configured to store data in different data back-ends with different characteristics depending on the usage scenario at hand.

Subscription and Notification: This capability manages the subscription and notification mechanism, which handles the notify process of updated data and status to subscribed entities. Through this mechanism, M2M nodes are able to receive event notifications asynchronously from devices and gateways. This process is further enhanced in OpenMTC by the Store And Forward (SAF) functionality that enables the handling of different notification streams based on their priority. Requests from applications in network, gateway, or device side are forwarded to this capability be means of the application enablement capability, which serves as a single contact point for M2M applications.

Location: This capability allows AEs to obtain geographical location information of different nodes (e.g., ASN, MN) that could be used on some location-based services. The device management server is used to issue commands to obtain the current location from a Global Positioning System (GPS) module.

Device Management (DM): The OpenMTC platform integrates a DM implementation based on the OMA LWM2M protocol [67]. A library for LWM2M message parsing and creating, and managing communication back to the registered clients is implemented on both front-end and back-end servers. The LWM2M library creates tree dictionaries of the supported management objects and is easy to extend with another one by adding new entries in the dictionary. The processing is then uniform for each of the management objects when it comes to parsing, storing, updating information related to the management objects. The LWM2M client-server interaction is done using the underlying data transfer protocol over CoAP/UDP, which decreases the message's overhead and improves efficiency [176]. The DM message's exchange is based on RESTful operations on resources bound to the devices by following the CoRE link format. The related resources are grouped together and handled by objects. The predefined set of objects includes Access Control, Connectivity Monitoring and Statistics, Device, Firmware Update and Location management. In order to extend the model, organizations can define and propose new objects that could be important for a robust device management platform.

Group Management: Responsible for handling group related requests in a One-to-Many interaction model, where the request is sent to manage a group and its membership nodes. Bulk operations include read, write, subscribe, notify, device management, etc.

Application Management: The main function of this capability is to provide a

single contact point for M2M applications to access the OpenMTC platform and expose available functionalities implemented in all xSCL/SCEs. The application management capability is responsible for routing the applications' requests to corresponding capabilities in the platform, in addition to routing requests between applications connected to the same domain. Through this functionality, the service requirements and the specific data exchanged between the applications and the OpenMTC are forwarded to the appropriate local handling functionality in both the devices and in the network platform.

Security: Two aspects of security have to be considered in the openMTC implementation. The access control aspect including identification, authentication and authorization is based on applying access policies as defined by *AccessRight* resource (from ETSI resource tree) or *accessControlPolicy* resource (from oneM2M recourse tree), where a set of privileges is defined and applied. Another aspect is using the Transport Layer Security (TLS) protocols to support sensitive data handling over TCP and UDP. The TCP security is commonly done using the SSL standard libraries and is applicable to both HTTP and MQTT. Security over UDP raises the complexity because of the nature of the communication. Since no connection is usually set up between the end points, a new mechanism has to be built wrapping the classical UDP socket. This is done using DTLS protocol.

6.2.3 Standardized Open Interfaces

OpenMTC supports a client/server based RESTful architecture with the hierarchical resource tree defined by ETSI M2M and oneM2M. The data exchange communication over open interfaces is independent of the transport protocol. OpenMTC supports both of MIa interface specified by ETSI as well as Mca interface specified by oneM2M. In order to support the development of M2M applications and make the core assets and service capabilities available to 3rd party developers, the OpenMTC application enablement layer defines a set of high-level abstraction APIs, which are categorized under three groups: Device, Data and Network API [31], as shown in Figure 6.1. These APIs hide the internal system complexity and allow the developer to focus on the implementation of the application logic. The openMTC platform supports both XML and JSON format for data representation.

In addition, OMA NGSI-9 and NGSI-10 interfaces for context management are supported on the backend server, to allow seamless integration of M2M platforms and other context management systems. The main benefit of adopting the NGSI standard API in OpenMTC is to enable the sharing of M2M data between heterogeneous context management platforms via standards interface points [170]. Thus, data gathered from "things" (i.e. sensors and devices) can be managed by the same services developed for other context-management platforms and supports the fusion of M2M data to enterprise-level. The main operations of both NGSI-9 and NGSI-10 and how to map the OpenMTC APIs to OMA NGSI specifications was presented

in [170].

6.3 Adaptable M2M Transport (AdM2M) Framework

The AdM2M framework is the author's own concept and the reference implementation was realized as part of the OpenMTC platform for connectivity management. The connectivity management module of OpenMTC provides the fundamental communication functionally to supporting interactions over managed or unmanaged core using HTTP, which is widely considered as the de-facto Internet standard. Based on the specification from Section 5.4, the AdM2M framework could be integrated into an ETSI/OneM2M compliant platform to enhance the adaptability of the connectivity management component. Figure 6.3 shows the high level architecture of OpenMTC and illustrates the position of the AdM2M framework within the connectivity management layer.

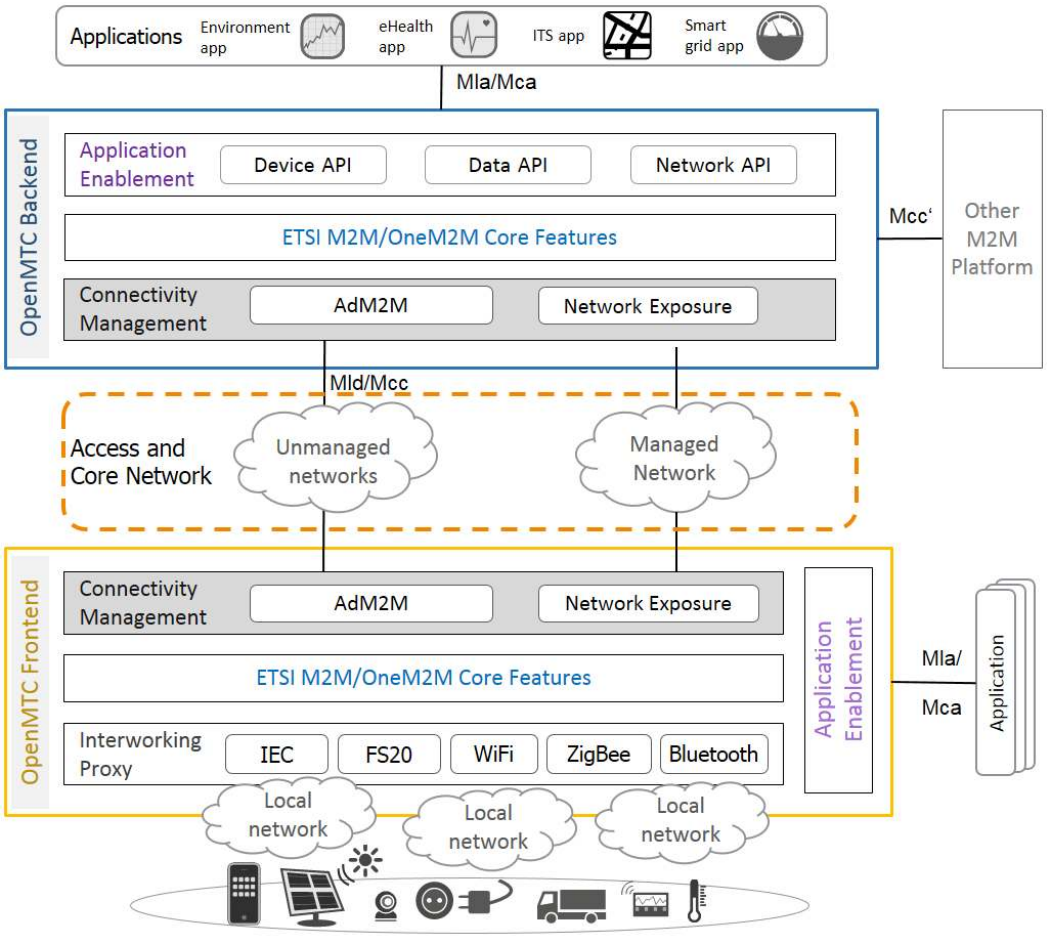


Figure 6.3: The Adaptable M2M (AdM2M) Transport Framework Part of the OpenMTC

The overall architecture, core capabilities and standard interfaces of the OpenMTC platform was described in a previous section. The connectivity management layer is responsible for the communication establishment and messages relaying with other interworkable M2M nodes; and the AdM2M framework will provide the proposed adaptability feature into it. The AdM2M framework includes pluggable transport protocols such as HTTP and CoAP, which could be dynamically plugged in and out to adapt to the network and application conditions. The objective of this functionality is to enable a more reliable end-to-end data delivery in Smart City systems.

Figure 6.4 shows the internal architecture of the proposed framework for M2M. The AdM2M framework consists of a Monitoring Module that constantly collects a set of parameters related to the data stream generated by registered ASN, i.e. rate and data size. The data are presented in *constanceInstance* resources created in response to application's POST requests, as specified by oneM2M. A Policy and Rules component is necessary to store the provisioned or learned policies and rules while the Connectivity Manager operations enable and disable interfaces and plugins depending on the operating system (Android or Linux) APIs. The AdM2M framework provides different transport protocol stacks, such as HTTP and CoAP as plugins to support the communication with M2M devices using the proper stack based on traffic condition. The framework is extendable to add more protocol libraries and plugins.

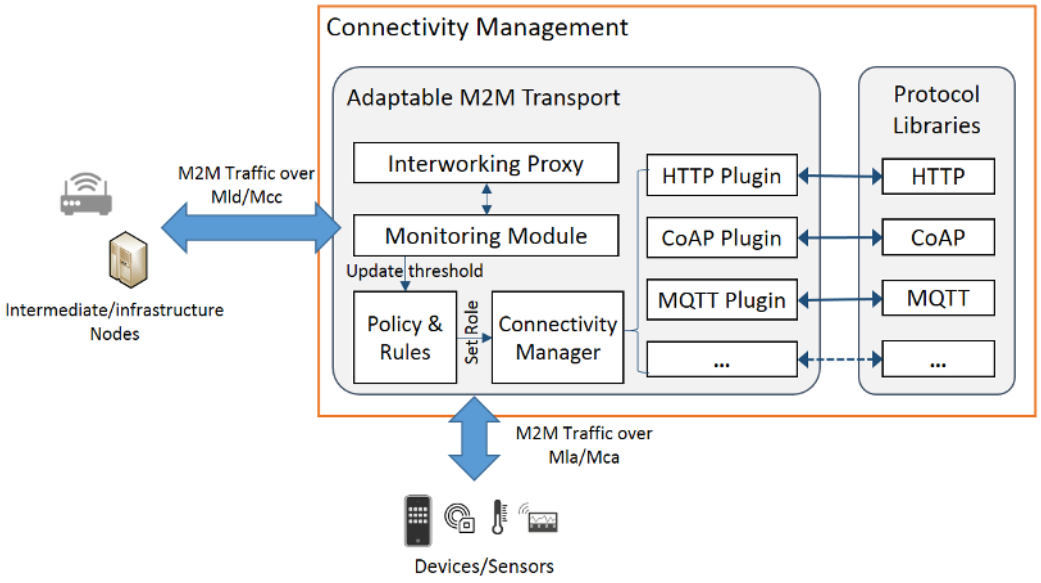


Figure 6.4: High-level Architecture of Adaptable M2M Transport Framework

6.3.1 Plug-in Protocols and Selection Module

As part of the AdM2M framework, several protocol libraries have been developed and used as a pluggable component, Generally, the AdM2M framework is integrated to a MN or an IN providing M2M core capabilities, the transport protocols shall be enabled and configured as shown in Listing 6.1.

Listing 6.1: Configuration of Transport Plugins

```
"plugins": [
  {  "name": "HTTPTransportPlugin",
          "package": "openmtc_scl.plugins.transport_gevent_http"
             ,
           "disabled": false,
           "config": {
           "connectors": [
                 {  "interface": "",
                     "host": "localhost",
                     "port": 14000,
                     "is_wan": false    },
                {   "interface": "",
                     "host": "localhost",
                     "port": 15000,
                     "is_wan": true     },
                {    "interface": "",
                     "host": "localhost",
                     "port": 16000,
                     "is_wan": true,
                     "key": "certs/CA_signed_certs/
                        http_server_localhost/server_keycert.
                        pem",
                     "crt": "certs/CA_signed_certs/
                        http_server_localhost/server_cert.pem"
                           }
                  ]
               }
         },
        {    "name": "COAPTransportPlugin",
             "package": "openmtc_scl.plugins.
                transport_gevent_coap",
             "disabled": false,
             "config": {
                  "client_port": 6684,
                  "connectors": [
                      {     "interface": "",
                            "host": "localhost",
                            "port": 14000,
                            "is_wan": false
                        },
                        {  "interface": "",
```

```
36                                  "host": "localhost",
37                                  "port": 15000,
38                                  "is_wan": true
39                              },
40                              {   "interface": "",
41                                  "port": 16000,
42                                  "is_wan": true,
43                                  "key": "certs/CA_signed_certs/
                                        dtls_server/server_keycert.pem
                                        ",
44                                  "crt": "certs/CA_signed_certs/
                                        dtls_server/server_cert.pem"
45                              }
46                          ]
47                      }
48                  }
49              ],
```

In Figure 6.5, an overview of request flow taking place between different M2M nodes is presented. M2M applications at ASNs could interact with the OpenMTC front-end and back-end in different ways such as sending data/measurements in a telemetry manner, or receiving notifications in response to events that were previously subscribed to. Possible interactions of different use cases have been discussed in Chapter 4.

An ASN within the field domain could send requests to the IN-CSE directly or via an MN-CSE (a gateway where the ASN is registered). In the latter case, the MN will retarget these requests to the IN-CSE on behalf of the ASN. The request destination address (represented by the resource URL) is analyzed and if it is not matched to any address on the machine IP address list, the message will be forwarded to the destination IP. This mechanism is implemented as a method to forward requests on a multi-hop route, bringing the advantages of enabling also transport protocol translation and channel selection based on different criteria, such as the cost or reliability. These channels could be also assimilated to physical interfaces, as Wi-Fi or Ethernet. The ASN uses a transport protocol to send its request to the CSE, where the corresponding server treats the requests and creates a RequestIndication. This request is transferred to the MethodDomain, in charge of triggering the internal process, like accessing or creating a resource. A ResponseConfirmation is sent back to the server, which converts it to a reply matching the original transport protocol used. The format of the request and reply payload can be specified by a client. Both JSON and XML format are supported by OpenMTC.

The AdM2M framework includes a set of different transport protocol stacks (HTTP, and CoAP) that are implemented as plug-ins, which could be used based on user's pre-configuration or based on the policy status. The plug-ins are using libraries, one for each protocol that provides the required logic for each. For example, the CoAP plug-in is using the Coapy library that was updated to be compatible with the latest version of the CoAP standard [64] and support the required features [177].

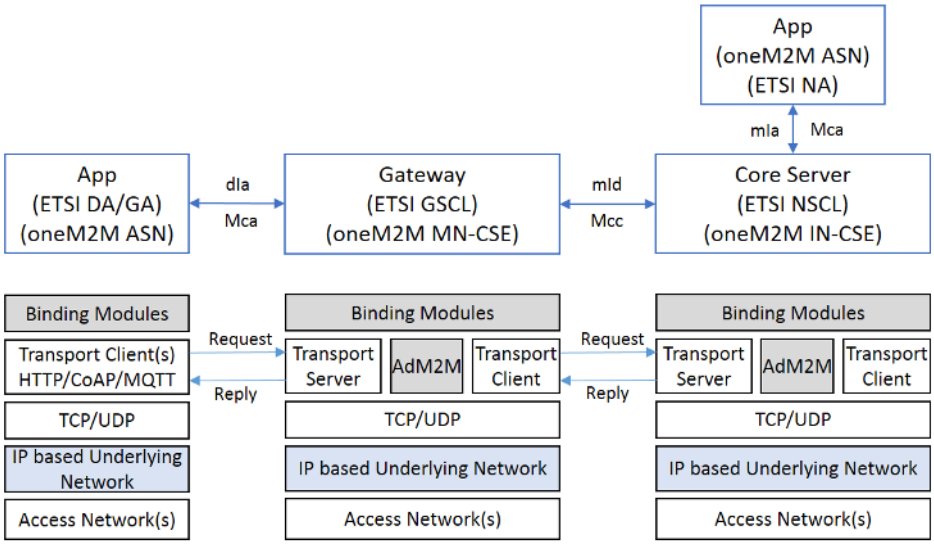


Figure 6.5: Overview of Transport Request Routing between M2M Nodes

The changes developed include adding the Observe, Uri-Query, Uri-host and Uri-port options, and updating the response codes names to the Y.XX format. On top of the protocols libraries, the integration with GEvent was built to handle received packets and convert to objects using the corresponding library. Similarly, objects are encoded into properly formatted byte strings before being sent on UDP/TCP channels. GEvent [178] is an event-based library providing asynchronous I/O API that can scale its number of execution units according to the processing load. The system could be easily extended to add more plug-ins for new protocols.

In Figure 6.6, the requests flow of a common M2M scenario is depicted, which illustrates the functionality of the AdM2M modules. As a first step, registration requests are issued from the gateway towards the core server, and from the applications toward associated front-end/back-end server (step ①). The successful registration of M2M nodes with unique IDs is required prior to further interactions. Applications have the ability to create multiple *Container* resources that could be used later as a mediator for data buffering.

Based on ETSI M2M and oneM2M specifications, resources presenting applications and containers at the gateway/MN shall be announced to other SCLs (step ②). An announced resource shall point to the original resource and consists of only a limited set of attributes such as the link to the original resource and *searchStrings* (step ③). Providing this information over announced resource, facilities the discovery process specified by ETSI/OneM2M for searching of data and devices. Thus, the issuer of the *discovery* request does not have to contact all SCLs in order to find the resources.

The monitoring model of AdM2M measures the response time of each request issued by the gateway or core server as shown in step ④. This measurement provides

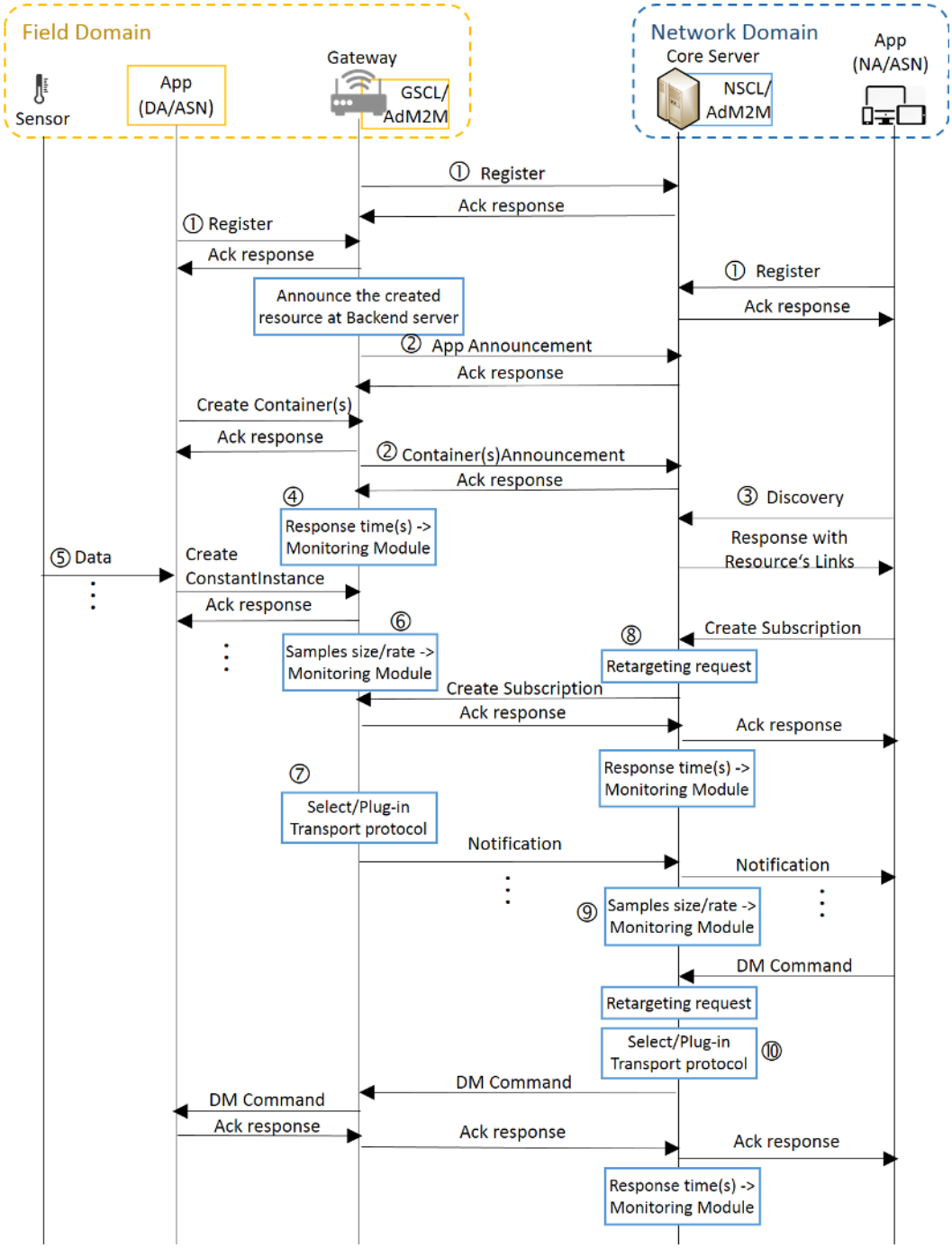


Figure 6.6: Requests Flow of a Common M2M Scenario

the M2M node with insights about the connection status and therefore used later in selecting proper protocol in line with the defined policy.

Registered application starts to push data, aggregated from attached sensors, by issuing a create *ContentInstance* resource request (step ⑤). The AdM2M monitoring

model measures the size of data and the sampling rate of the create *ContentInstance* resource requests of each App, as shown in step ⑥.

On the Network domain, applications interested in receiving notifications related to data/events from applications/devices located in the field domain could send a request to create *Subscription* resource. As the target resource of the subscription request is not stored in the core server, the retargeting mechanism is used to forward the request to its destination. Whenever the notification process is triggered (i.e. the specified filter criteria are met), an evaluation process is executed to provide a decision on selecting which protocol to use in forwarding notifications (step ⑧).

Similarity, the monitoring module shall collect measurements from forwarded notifications (step ⑨), to upgrade the policy rules and used this information to plug-in transport protocol for further requests, such as sending a device management request to devices (step ⑩).

6.3.2 Interworking Proxy

The partners of the UNIFI project [173] have established several activities with the aim of federate FI testbeds, as the objective of the UNIFI project is to build sustainable teaching and research infrastructures in the areas of Future Internet and Smart Cities through global collaboration among academic institutions. This project also includes the creation of Competence Centers for a sustainable development and bundling of local expertise in Chile, Vietnam, South Africa and Thailand with a strong collaboration with TUB in Germany. Part of the activates within this project was building a federation of labs that support Smart City prototyping. To this end, an interworking proxy was required to support the interoperability of existing testbeds.

The department of Electrical Engineering, faculty of Engineering at Chulalongkorn University-Thailand has implemented a Building Energy Management System (BEMS), called "CUBEMS", based on the IEEE1888 standard [84]. This system is composed of 200 ZigBee and 6LowPan sensors and 20 smartmeters. Additionally, 4 points in-house displays are installed in three buildings of the Chulalongkorn University, and used by the CU-BEMS App for displaying overall energy consumption.

To design the interworking between IEEE1888 and ETSI M2M standards, two alternatives exist depending on where the interworking process is instantiated, namely, at the Front-end using a GIP (gateway interworking proxy) or at the Back-end server using NIP (network interworking proxy).

In CU-BEMS, gateways are originally designed for collecting the sensor data from ZigBee networks and converting to IEEE1888 format as well as finally submitting data to the IEEE1888 storage. The gateways have been implemented on specific micro-controller platforms, Arduino Mega 2560, with limited processing capability of 16 MHz and memory size of 256KBytes. Therefore, it was decided to implement all the data synchronization logic within a NIP, rather than a GIP.

As specified in Section 5.4.3, The interworking proxy performs two main functions. First mapping the operation and protocol requests between interconnected

platforms in both directions. Second mediating the data model used by the interconnected platforms, i.e. REST/JSON and SOAP/XML.

Figure 6.7 depicts the end-to-end request flow for data transfer in the direction from sensors connected to IEEE1888 gateway towards the ETSI M2M gateway. Initially, The NIP sends a TRAP query to the IEEE1888 Storage component to receive the sensor data when available (step ①). In a similar way, the GIP at the ETSI gateway sends a Create Subscription request for subscribe on Container resource(s) used by the NIP to store updates received from interworking platform (step ②). Upon IEEE1888 gateway receives data from sensor, it sends the data to IEEE1888 Storage component for collecting data using WRITE protocol (step ③). Simultaneously, IEEE1888 storage return the callback data to NIP TRAP protocol previously issued (step ④). The NIP mediates data to the supported model in the ETSI M2M structure and sends Create requests to store the data in the server repository i.e. NRAR (step ⑤). The back-end server forwards notify message to all subscribers, thus the GIP receives the data notifications in step ⑥. As a response the gateway sends actuating commands to connected actuators.

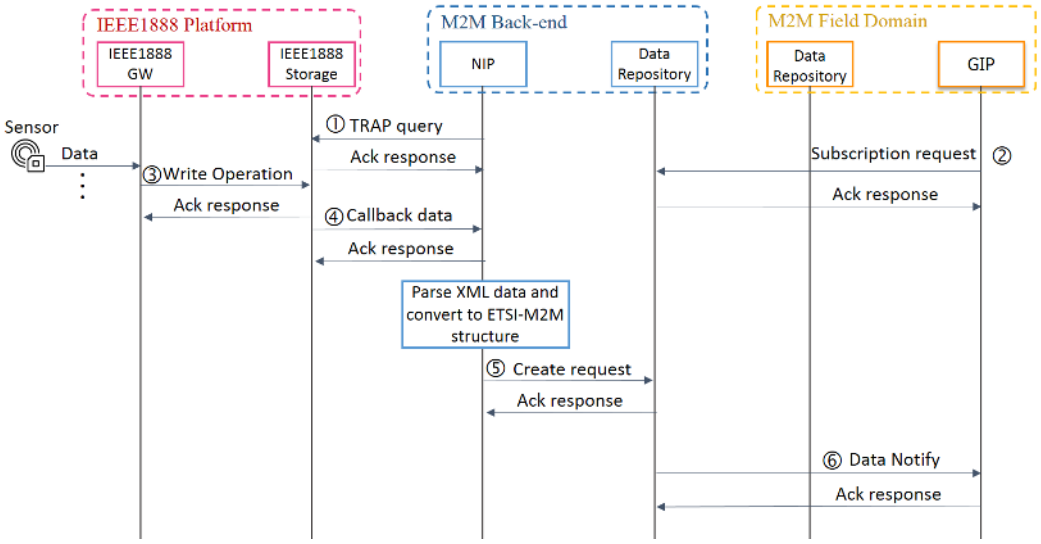


Figure 6.7: Data Synchronization from IEEE1888 Platform to ETSI Repository

Similarity, Figure 6.8 depicts the end-to-end request flow of exchanging data in the other direction from ETSI M2M gateway to IEEE1888 gateway. Where the IEEE1888 gateway and NIP receive updated data by TRAP protocol and NOTIFY method, respectively. Initially, NIP sends Create request for data subscription to NRAR after that IEEE1888 gateway send a TRAP query to IEEE1888 Storage. The GIP stores the sensors data at the gateway repository (GRAR) using Create requests receives. Simultaneously, the gateway returns notification messages to NIP and any other subscribers. The data received within the NOTIFY is encoded to

base64 format within a JSON object, as shown below:

Listing 6.2: Notification Received by The NIP

```
{"notify": {"statusCode": "STATUS_OK",
        "representation":{"$t":"eyJkYXRhIjp7InRpbWVzdGFtcCI6
            IjIwMTQtMTAtMTRUMTk6NTI6NDYuMDAwKzA3OjAwIiwiY29uc3
            VtZWQiOiJPRkYifX0=",
                "contentType": "application/json"},
        "subscriptionReference":"/m2m/applications/NIPA_IEEE1888/containers/
            elevetorfront_z1_sensor2_monitor_pir/contentInstances/
            subscriptions/subscription5428231"}
        }
```

The NIP converts the received data to IEEE1888 structure, as shown in Listing 6.3, and send the data in XML format to be stored in IEEE1888 storage by using WRITE protocol.

Listing 6.3: Notification Converted to IEEE1888 Structure

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
        <soapenv:Body>
        <ns2:dataRQ xmlns:ns2="http://soap.fiap.org/">
        <transport xmlns="http://gutp.jp/fiap/2009/11/">
        <body>
        <point
                 id="http://bems.ee.eng.chula.ac.th/eng4/fl13/corridor/
                     elevetorfront/z1/sensor2/monitor/pir">
        <value
                time="2014-10-14T19:52:41.000+07:00">
                OFF
        </value>
        </point>
</body>
</transport>
</ns2:dataRQ>
</soapenv:Body>
</soapenv:Envelope>
```

As the IEEE1888 gateway has previously sent a TRAP query to the storage component, the IEEE1888 storage component returns the callback data to the IEEE1888 gateway. Depending on the application logic, the IEEE1888 gateway sends actuation command to actuators.

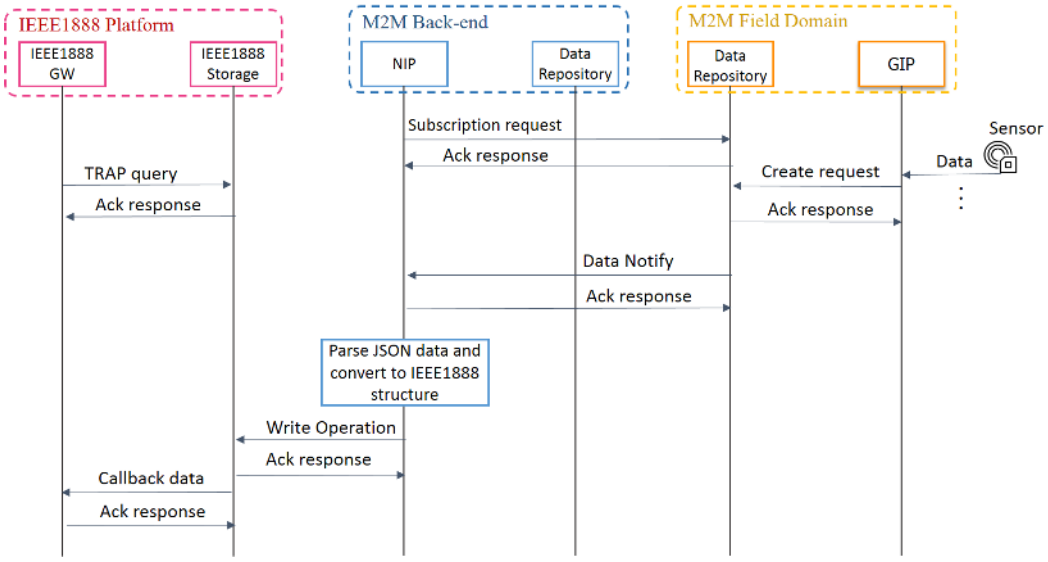


Figure 6.8: Data Synchronization from ETSI Gateway to IEEE1888 Platform

7 Evaluation

7.1 Introduction

This chapter presents the evaluation and verification of the AdM2M system, which has been specified in Chapter 5, and based on the implementation described in Chapter 6. To this end, a series of testbed deployment setups were conducted using the FUture SEamless COmmunication (FUSECO) Playground at Fraunhofer FOKUS institute, as well as the interworking testbed of the UNIFI and Testbeds for Reliable Smart City Machine to Machine Communication (TRESCIMO) projects. The main results have been published to certify the approach by the scientific community, with some results based on [45, 39, 40, 44].

A proof-of-concept validation of developed components is presented to assess the performance of the implemented platform and protocol libraries, as well as providing insights on protocol selection when considering the usage scenario. Further, verifications on specific domains are evaluated in the context of research projects. Finally, a comparison with other approaches of open source M2M platforms is summarized.

7.2 Proof-of-Concept Verification within the FUSECO Playground

The FUSECO Playground offers an independent and open testbed for research and prototype development of mobile broadband communication and service platform. Due to the flexible and modular design of FUSECO Playground, simplified Proof-of-Concept validation spanning from devices over access and core network technologies is possible [179]. It has been used on many academic and industrial projects as an open environment for early prototyping of new services, related components, protocols, and applications.

In this Section, the results of the conducted tests during this research are presented and results are discussed. The prototype implementation is deployed with different testbed setups and with consideration to various target scenarios. The following subsections introduce these setups that validate the functionalities of the reference implementation for delivering M2M services.

The testbed setup within the laboratory of FUSECO Playground, depicted in Figure 7.1, is comprised of an M2M front-end gateway, an M2M back-end server and several nodes to emulate device applications interacting with the front-end and back-end server. For both the M2M front-end gateway and the M2M Back-end server, the OpenMTC was used to operate the M2M service functionalities. In the performance evaluation presented in this chapter, two types of hardware deployment was used for the OpenMTC gateway: i) an embedded system operated with Raspberry Pi (model B) BCM2708 processor, uses the ARMv6 instruction set with 400MB RAM, and ii) a PC with 2.4GHz quad-core Intel processor, 8GB RAM.

As a first stage of verifying the performance of the implemented protocols plug-ins and libraries, some experiments were executed using the M2M testbed. The

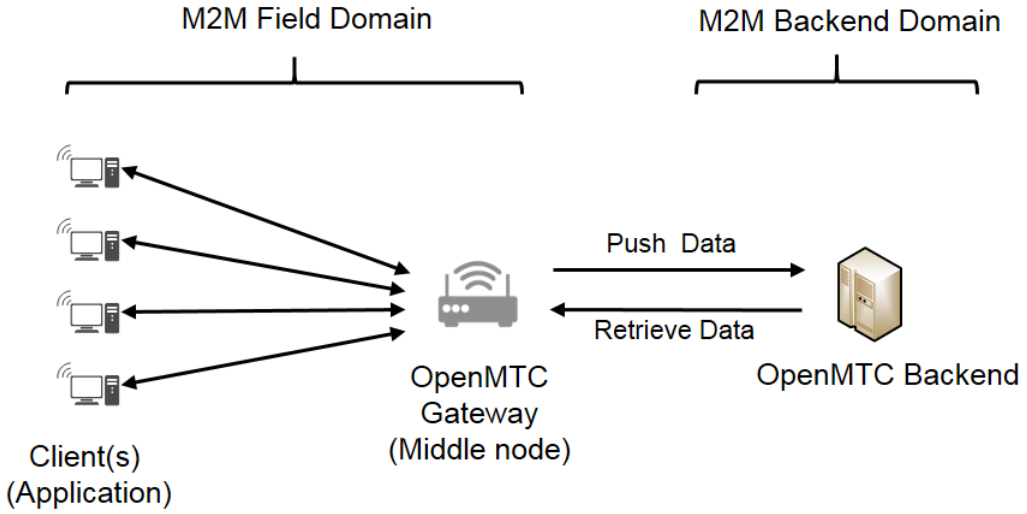


Figure 7.1: The M2M Testbed Architecture

experiments support gaining a better understanding of the expected performance of M2M applications associated with a special traffic model when using any of these protocols in sending or retrieving data to an M2M front-end server.

Apache JMeter [180] was used to carry out all evaluation scenarios. Apache JMeter is a pure Java application designed for load testing and performance measurement of different servers/protocols such as HTTP, HTTPS, REST, etc. Also, it could be extended to support other protocols, such as CoAP and MQTT through multiple pluggable samplers. Additional sampler plugins for CoAP, MQTT and AMQP were deployed to perform the presented performance evaluation in this chapter.

Part of tests were conducted using one device that interact with the M2M gateway over one-hop communication in order to observe the performance under specific traffic patterns. Moreover, to emulate the use case of having multi-devices connected to the OpenMTC gateway, we have used the parallel JMeter Ant task. In fact, Apache Ant is a Java library and command-line tool whose mission is to drive processes described in build files as targets and extension points dependent on each other. For the high sample rates, we have used the JMeter distributed testing, which involves running the same test scenario on different virtual machines on the cloud to overcome the one machine limitation. One machine (master) can control any number of other machines (JMeterEngines/slaves) and collect all the data from them. All the test machines clocks have been synchronized using the Network Time Protocol (NTP).

7.2.1 Effect of Payload Size

During the evaluation process, the tests consider two types of M2M transactions between devices and a gateway. That include re-pushing data (telemetric interaction) and retrieving data (inquiry interaction). Therefore, we will study the case of having different devices/sensors pushing/retrieving different types of data to/from the platform. All the interactions and the M2M traffic exchange will be established through both HTTP and CoAP protocols.

In this test, all testbed components attach to the same LAN and use a wired Ethernet connection (1Gbps). By choosing wired connection we aim to eliminate any problems related to the network transmission such as packet loss, congestion, etc. Each test scenario lasts longer than 5 minutes and is repeated at least 5 times in order to get accurate results. The aim of this test is to evaluate the effect of the payload size on the performance of pushing data in a telemetric manner to the gateway.

Figure 7.2 plots the response time and the 95% confidence interval for POST requests generated on a fixed sampling rate of 100Hz to the resource-rich gateway against different payload sizes. Similarly, Figure 7.3 shows the plots of the response time and the 95% confidence interval on a fixed sampling rate of 1Hz to the resource-constrained gateway in different payload sizes. It is clear that the impact of payload sizes, smaller than 1 Kb, is negligible for both HTTP and CoAP protocols and both platforms. The allowed block size with the CoAP block-wise transfer option

is between 16 and 1024 bytes, which leads to sending only one block in each test scenario (payload size<1Kb). In addition, the Ethernet MTU is equal to 1500 bytes at the network layer, leading to no IP fragmentation of CoAP messages (UDP supports larger payloads through IP fragmentation), and no TCP fragmentation of HTTP messages. Although CoAP has the advantage of a smaller overhead of only 4 bytes, the encoding/decoding time is slightly affected by the payload size. Similar results were observed with the constrained-resource gateway for fixed sampling rate of 1 Hz.

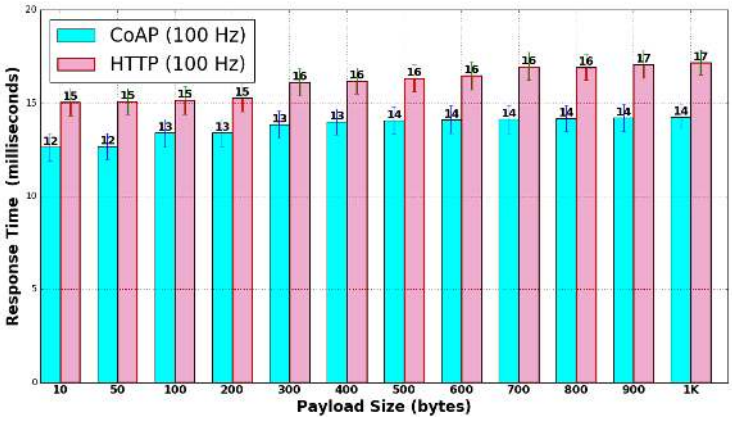


Figure 7.2: Response Time of Push Requests with Different Payload Size on A Resource-Rich Gateway (Linux PC)

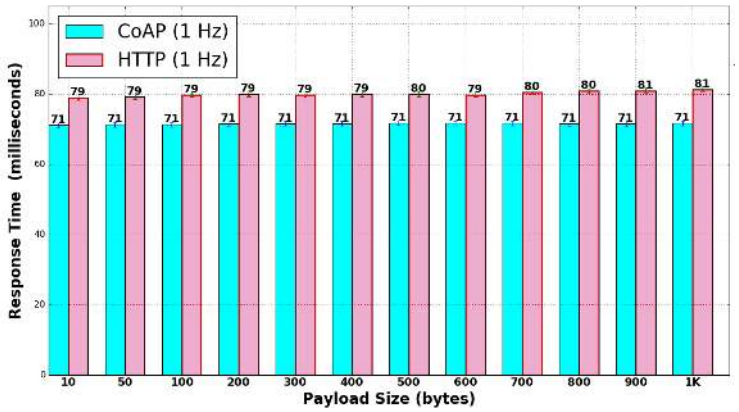


Figure 7.3: Response Time of Push Requests with Different Payload Size on A Resource-Constrained Gateway (Raspberry Pi)

Table 7.1 presents the obtained results for different payload sizes higher than 1Kb. The results prove the scalability of the OpenMTC gateway, deployed at both platforms, to handle different payload sizes in an acceptable period. On the other hand, they show the slight effect of the packet fragmentation on the response time using HTTP. However, as shown in Table 7.2 the response time was greater affected

when using CoAP. The block-wise transfer option for CoAP allows for block size between 16 Byte and 1024 Byte only, in this test the biggest block size (i.e. 1024byte) was used.

Table 7.1: Effect of High Payload Sizes on the Response Time Using HTTP

Platform	Rate (Hz)	Payload Size (Kb)	Response Time (ms)
Raspberry Pi	1	2 - 100	81 - 106
PC	1	2 - 100	6 - 9
		500 - 10000	16 - 244
	100	2- 20	17 – 25
		30 - 100	344 – 1158

Table 7.2: Effect of High Payload Sizes on the Response Time Using CoAP

Platform	Rate (Hz)	Payload Size (kb)	Response Time (ms)
Raspberry Pi	1	2 - 100	104 - 480
PC	1	2 - 50	12 - 181
		100 - 300	369 - 1014
	100	2 - 5	15 – 185
		10 - 30	379 – 1113

7.2.2 Effect of Request Rate

To study the effect of request rate on the performance of the data aggregation process, POST requests have been sent at a fixed payload size of 200bytes, which can be considered as the mean size of payload generated by sensors/devices likely connected to M2M systems. In these tests, a slight increase in the response time values is observed for the sampling rates lower than 250Hz (for CoAP) and 270 Hz (for HTTP); this is followed by a strong surge that exceeds one second for rates higher than 350Hz with both protocols. A deeper look at the changes behavior on the response time over a specific period when using HTTP allows us to distinguish between two states for rates lower than 270 Hz. At the beginning of each connection, the delay is extremely high and unstable, and, later on, it settles with lower values during the remaining time. These steep values at the starting of each connection could be attributed to the fact that HTTP is transported on a TCP connection which causes higher latency due to the connection setup process over the three-way handshake. However, for rates higher than 280Hz, we have only one state as the response time values remain high during the period. The processing time within the OpenMTC platform becomes longer. Using CoAP, we notice a strong rise in the delay values for rates beyond 250Hz. The same surge has also occurred with

lower payload size messages (around 10 bytes). This could be related to the CoAP protocol implementation according to [64] where it was mentioned that a 16-bit size message can enable up to about 250 messages per second from one endpoint to another with default protocol parameters similar to this case.

Figure 7.4 and Figure 7.5 plot the response time for pushing data request and the 95% confidence interval using Linux and Respberry Pi platform respectively. The resources usage during the tests was also monitored. The memory usage is relatively the same for both protocols and for different sample rates. However, the CPU utilization appears slightly higher when using HTTP. The resource consumption differences between HTTP and CoAP is clear when using the Raspberry Pi. Generally, CoAP performs better than HTTP for the high sample rates with both platforms.

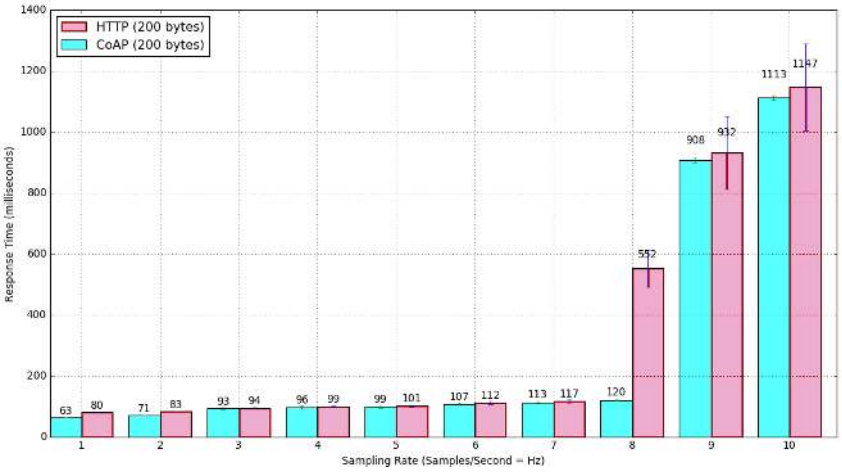


Figure 7.4: The Response time of Pushing Data to Resource-Constrained Gateway (Raspberry Pi)

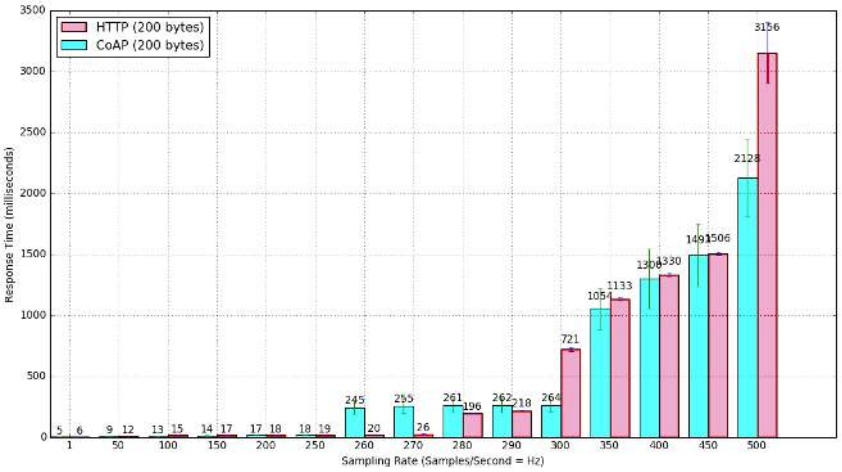


Figure 7.5: Response time of Pushing Data to Resource-Rich Gateway (Linux PC)

Figure 7.6 plots the average response time for different sampling rates using multiple connected devices. The requests were pushing data to the OpenMTC gateway deployed on Linux PC with fixed payload size of 200Bytes. The threshold limit of connected devices could be observed from the plot for a given sampling rate. Generally, it was noticed that the response time increases rapidly when the number of total transactions is between 240 and 320 per second for HTTP. In the case of using CoAP, the increment of the response time was variable depending on the number of devices and sample rates.

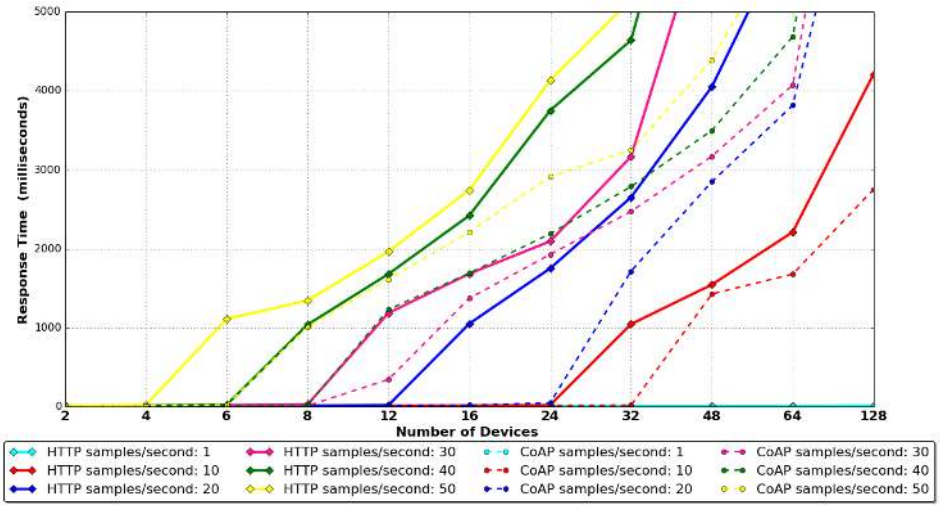


Figure 7.6: Response Time of Multiple Nodes Connected to Resource-Rich Gateway (Linux PC)

7.2.3 Discussion

Based on previously presented results, the deployment of the OpenMTC platform is showing an acceptable level of reliability and scalability taking into account the hardware resource-capabilities and number of connected sensor nodes. Furthermore, the results provide more insight on the importance of selecting a proper protocol stack with certain M2M use cases. The advantage of the CoAP protocol was more noticeable on the resource-constrained gateway, as the response time of data pushing and retrieving requests is less affected than with HTTP. Similar observation is obtained for the consumption level of hardware resources.

In general, HTTP is an ideal transport protocol for requesting data from known sources over the Internet. However, the high demanding of bandwidth and computation resources limits its suitability for a number of M2M applications. Also, it does not provide scalable means for bi-direction communication such as sending notifications, and the textual encoding of HTTP headers obtains unnecessary overhead for parsing. Consequently, HTTP is a good choice for forwarding data over a reliable connection to back-end servers, while CoAP is better suited to exchange data and control commands between gateways and constrained-devices. For exam-

ple, in the case of a Smart-Building gateway connected to multiple sensors nodes, with limited resources, and forwarding aggregated data/events to a backup server once a day. It is advisable to deploy a connectivity stack using CoAP to interact with the sensors, while using an HTTP connectivity stack for forwarding the big payload of aggregated data.

Similar tests have been conducted to evaluate the payload size in pushing data using both MQTT and AMQP protocols which follow the Pub/Sub approach. The RabbitMQ library [181] was used in these tests although the integration of MQTT and AMQP library within the OpenMTC platform is dedicated to future work. Both MQTT and AMQP were able to handle payload messages bigger than 100KB. Table 7.3 shows the threshold limit of message rate with different payload sizes. It is observed that MQTT has a slightly higher message limit than AMQP, that is mainly related to the difference in frame header sizes of each protocol, as shown in Table 3.5. In contrast to CoAP and HTTP, MQTT performs much better with big payload messages. However, CoAP is more efficient in energy usage and produces less extra traffic in the case of small-size messages.

The ordering of messages delivered to subscribers is guaranteed with both protocols, only when using the same QoS level with MQTT, and the same queue in AMQP. Both protocols provide efficient mechanisms for transferring one-to-many notifications, which makes them a good choice for forwarding notifications towards network applications.

Table 7.3: Threshold Limits of Message Rate with MQTT and AMQP

Payload Size (Byte)	AMQP Message rate (mps)	MQTT Message rate (mps)
10	1022	1044
100	1020	1039
1K	1002	1010
10K	1001	1005
100K	497	986
1M	-	231

7.3 Specific Domain Experimentation

The experimental observations presented and discussed on the previous section has helped in defining the guidelines of protocol and has supported the further implementation steps of the proposed framework. In the following subsections, further experiments considering specific domains and use-case conditions are presented to evaluate the system performance in the context of Smart City services.

The features of the AdM2M framework, integrated within the OpenMTC platform, map to the requirements, previously defined in Table 4.4. Generally, The

OpenMTC platform is compatible with both ETSI M2M specifications and oneM2M specification. The end-to-end architecture is based on IP protocol stack, however the integration of non-IP access technologies is possible over inter-working proxies. This enables the platform to support the interaction with heterogeneous devices over multiple-protocols. Extending the AdM2M capabilities is possible by adding further protocols and interworking proxies in a pluggable manner.

The developed notification mechanism and the SAF functionality allow connected nodes at both front-end and back-end to perform data reporting in different patterns (e.g., periodical, on-demand or event-based). This allows for well-adaptation to different services and topologies. Furthermore, the overall system is able to handle the increase in the number of requests or connected devices to an acceptable level.

The underlining design supports the need to perform accounting and charging depending on application usage, however, the charging process is out of this thesis scope.

7.3.1 Experimentation Related to Smart Energy Domain

In this section, a Smart Energy application is considered, which is the Substation Automation application for monitoring, protection and control functions performed on a Smart Grid substation and feeder equipment. The analysis of communication requirements and traffic loads for this application presented in [156] was adopted in our experiment.

It was concluded, that CoAP is recommended to be used for pushing data flows at low rate frequency carrying small size payloads, while HTTP is recommended for data flows at high rate frequency with relative bigger payloads. In this experiment, we measured the end-to-end response time of two data flows: the first presents sampling data from emulated sensor, the second flow presents control signal flow from an OpenMTC front-end gateway. The data flows are pushed over the testbed using PC-Linux front-end OpenMTC. Each POST request carries a JSON formatted payload similar to the example shown in Listing 7.1.

Listing 7.1: Example of ContentInstance Resource

```
1 {"contentInstance":
2     {"content":
3             {"$t":"eyJkYXRhIjp7InRpbWVzdGFtcCI6IjIwMTQtMTAtMTRUMTk6NTI6
                NDYuMDAwKzA3OjAwIiwiY29uc3VtZWQiOiJPRkYifX0=",
4              "contentType": "application/json"}}}
```

Figure 7.7 [39] shows two plot diagrams for the response time of data flows, that represent aggregated data: i) sampling data (rate 1 sample/second and payload size 1600Byte), ii) control parameters (rate: 100 request/s, payload size 200Byte). As remarked in Table 4.2, the substation automation application is strictly delay sensitive. Therefore, it is required that the delay remains in the level of 200ms or less. The plots present the improvement on overall performing when using CoAP in transporting the control interactions instead of HTTP.

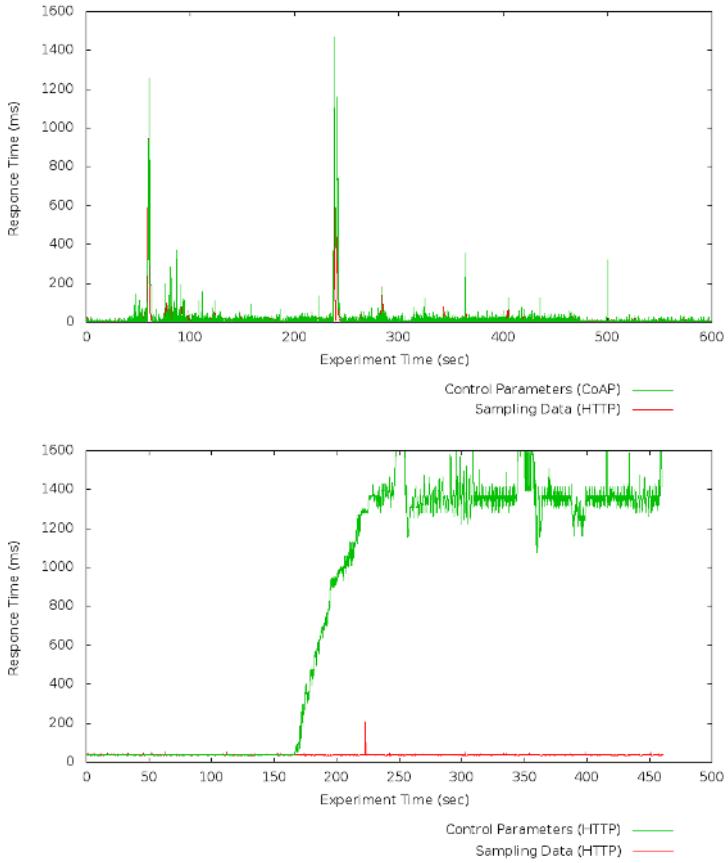


Figure 7.7: Performance of Emulated Substation Automation Application

7.3.2 EHealth Experimentations

In this section, the results of testing the concept of using multiple flows of data streams in an EHealth scenario is presented. In this case, the end user is assumed to be using some attached sensor to monitor his/her heart ECG signals in a Remote Patient Monitoring (RPM) service.

The traffic in this case experiment consists of multiple simultaneous data flows with different rates: 1, 10, and 100 req/s. The data flows are pushed to the Open-MTC front-end by a POST request. Figure 7.8 shows the response time of the data flows that represent aggregated data from: i) blood pressure sensor (rate 1 sample/second, sample size 16 bit), ii) ECG leads (rate: 1250 sample/s, sample size 12 bit), and iii) oxi meter (rate: 75 sample/sec, sample size 5 byte). The plot shows that the response time of all sent request during the experiment test was at an acceptable level, which was not the case when using either HTTP or CoAP for all flows. This proves the need to support multiple protocols in M2M nodes even if used in one type of application.

Usually, the data are stored in the user-owned gateway that implements some data analysis to detect any emergency case. In addition, a set of accumulated

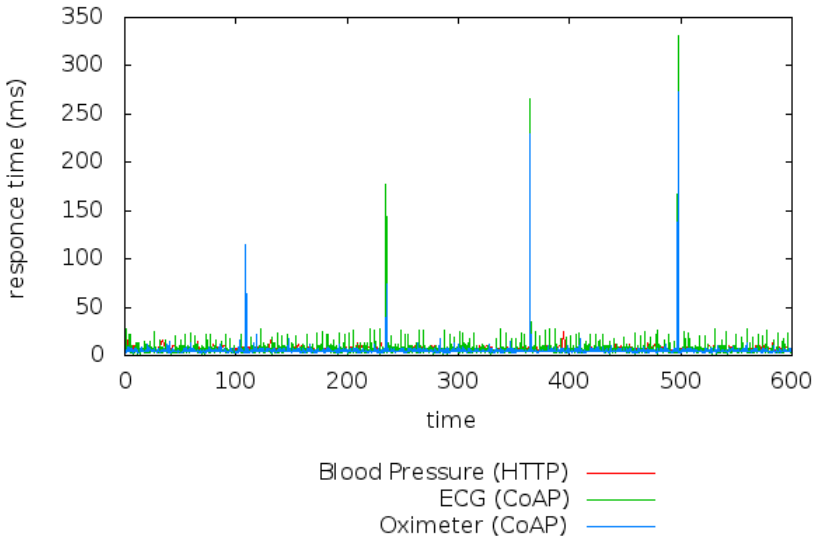


Figure 7.8: Response Time of Remote Patient Monitoring Service of Three Flows

data shall be sent to the central server once per day for further analysis, and it is possible that the gateway receives a software update, when available, from the service provider. This backup or update traffic has less priority than the vital signs, but will produce a huge amount of traffic with larger payload messages.

The measurements in Figure 7.9 plots the rolling average for the response time of the Post requests used to push the data from the sensor adapter to the gateway [40]. In this experiment, the ECG signals were considered to be generated at a data rate of 250 requests per second and a fixed payload size of 200 Bytes. The measurements show three scenarios, the first one is when the data is pushed without the existence of any background traffic in/out of the gateway. In the second scenario, an additional data flow is taking place that present firmware update downloading to the gateway on a low rate (1 Post request/s, 5Kbyte payload). The third scenario is similar but the proposed framework is splitting the update flow into two flows based on the BufferResource parameters, as described on Section 5.4.2. From the plot, the performance's improvement of the proposed framework is observed.

7.4 Interworking M2M Platforms Experimentations

The interoperable system was tested and evaluated in an end-to-end synchronous data exchange scenario. The developed testbed is depicted in Figure 7.10. The data aggregated from sensors connected to the OpenMTC gateway, shown on the left, are mediated to control the actuators on the right side.

The system enables the control and data exchange in the reverse direction as well. The message flow between both platforms is presented in Figure 6.7 and Figure 6.8. The gateways of the CU-BEMS system are designed to collect the sensor's data,

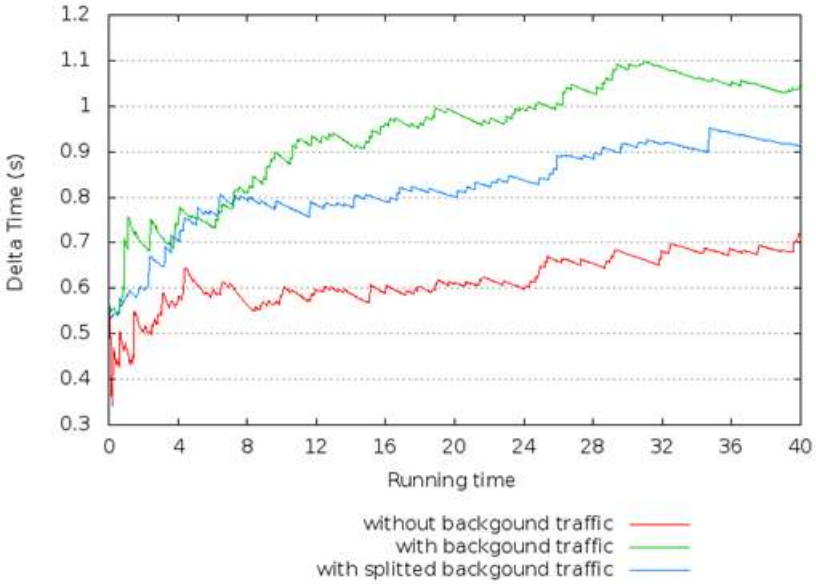


Figure 7.9: Performance of Ehealth Service

convert it to IEEE1888 format and store it in the IEEE1888 storage component. Similarly, the OpenMTC gateway was configured to forward sensor's data to the back-end server.

The gateways have been implemented on specific micro-controller platform, Arduino Mega 2560, with limited processing capability of 16 MHz and memory size of

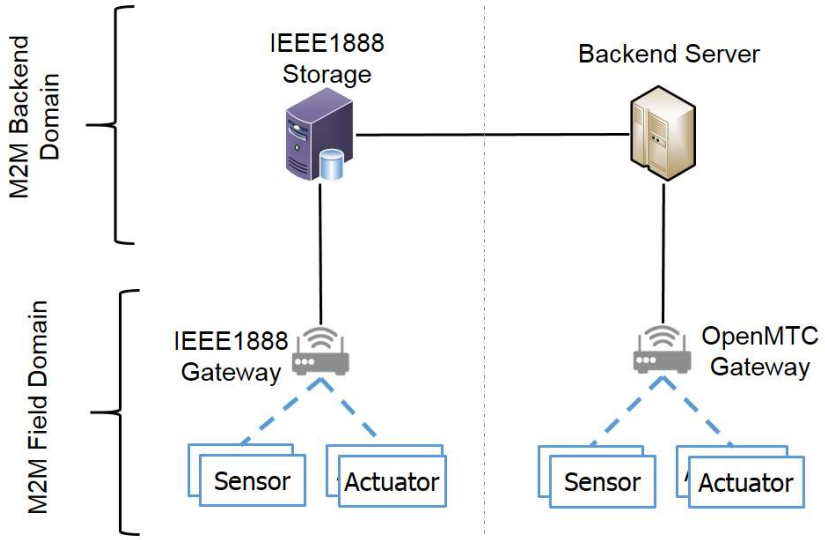
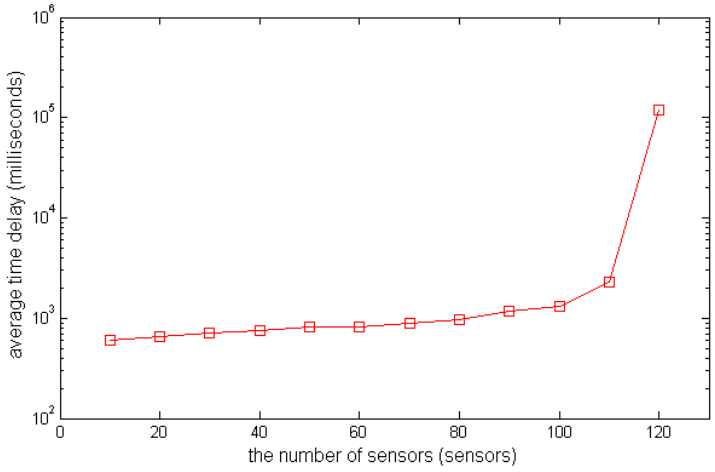


Figure 7.10: Interworking Testbed Used within the UNIFI Project

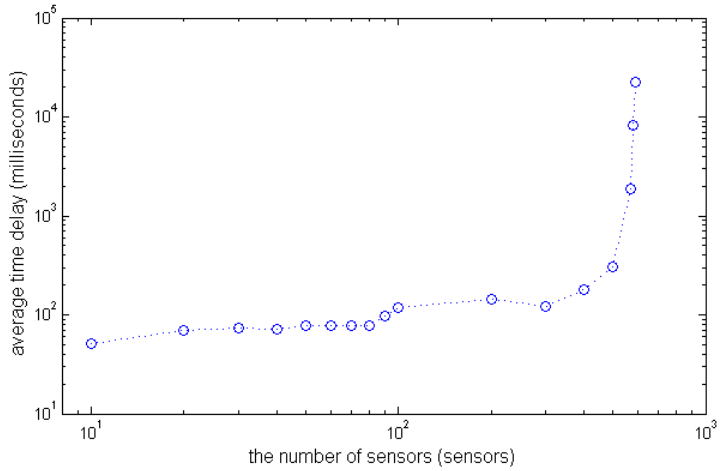
256 KBytes. The sensors are connected to the gateway via ZigBee network. As the gateways were implemented in resource-constrained setup, we have chosen to implement all the data mediation process not on the gateway, but rather on the back-end server as a Network Interworking Proxy (NIP). The NIP implementation can then be ported easily to locate at the high-performance server currently responsible for serving the central storage of CU-BEMS.

On the evaluating process, the measured parameters include the delay time, bandwidth and CPU usage. Figure 7.11a and Figure 7.11b show the average delay of end-to-end data exchange in both directions between ETSI M2M and IEEE1888 gateways.

The load test of the system was conducted by means of emulation of multiple



(a) Average Delay for Data Transfer Towards the IEEE1888 GW

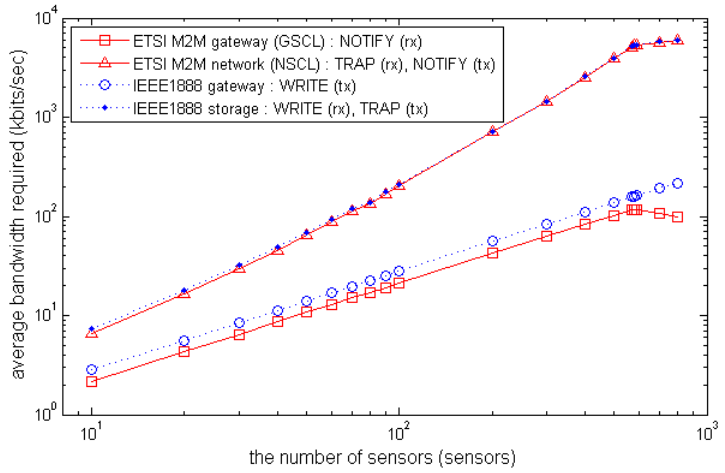


(b) Average Delay for Data Transfer Towards the ETSI M2M GW

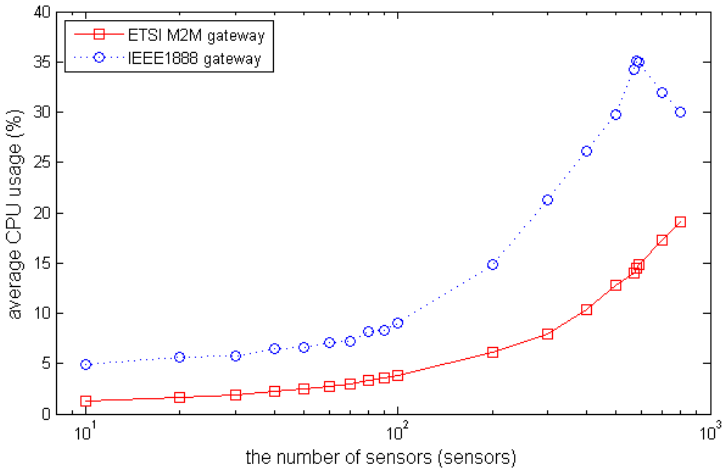
Figure 7.11: Performance of IEEE1888 and ETSI M2M Interworking

sensors to figure out the saturation boundary of the system in terms of contacted nodes pushing data simultaneously, while ensuring the receive of data notifications on the other side with acceptable delay. The end-to-end delay was significantly higher in the case of transforming data to the IEEE1888 GW compared to the case of transforming data to the ETSI M2M GW. That was due to the queuing delay at the IEEE1888 storage to perform notify callbacks. However, the system has successfully transformed data aggregated from 120 sensors at average delay of 1 second, which is considered acceptable for BEMS and Smart Building applications.

Figure 7.12a shows the average throughput of different messaging flows between interworking components while Figure 7.12b shows the CPU utilization at gateways of both interworking systems.



(a) Average Throughput for Data Transfer Towards the ETSI M2M GW



(b) CPU Utilization at Gateways for Data Transfer Towards the ETSI M2M GW

Figure 7.12: Resource Utilization of the IEEE1888 and ETSI M2M Interworking

The Internetwork proxy was also tested within the premises of the Telecommunication System Research Laboratory at Chulalongkorn University. The data of 28 sensors connected are part of the UNIFI testbed, the sensors consist of temperature, humidity, luminance and passive infra-red (PIR) movement detection which send data to the gateway once per minute with 28 Bytes payload. Moreover, the PIR sensors send data immediately upon a change of people motion being detected with 22 Bytes per time. This action results in an asynchronous data transformation that could be used for a real-time lighting actuator control.

7.5 Federated Testbed for Smart Cities

The need for large-scale testbeds for testing and evaluating Smart City services has been recognized by industry and academia [5, 43]. The FP7 TRESCIMO project [182] aims to create an infrastructure for the experimental evaluation of different Smart City related use cases [183]. The project mainly focuses on the collaboration between Europe and South Africa in building a federation of testbeds for Smart Cities applications. The federated testbed allows for experimentation with enabling technologies, standardized platforms and Smart Cities applications with different configurations. TRESCIMO approach to address different issues in the context of both developing and developed world is to interweave a sophisticated Smart City platform developed by Council for Scientific and Industrial Research (CSIR) and the ETSI/OneM2M compliant M2M communication framework Open Machine Type Communication [41] as well as a Delay-Tolerant Networks with the M2M framework.

The TRESCIMO federated testbed, depicted in Figure 7.13, consists of three interconnected sites, namely the TUB testbed located in Berlin (Germany), CSIR testbed in Pretoria (South Africa) and the UCT testbed in Cape Town (South Africa). The three testbeds are interconnected via a Virtual Private Network (VPN) connection that is managed by OpenVPN [184], which is an open source program for handling of VPN connections. At each site, various physical devices and an OpenStack installation are available. The latter is used to provide virtualized instances of the Smart City Platform, OpenMTC Server, OpenMTC Gateway, OpenMTC IWP (Inter Working Proxies) and emulated devices depending on the resources available at the location [185].

The reference architecture was designed to allow multiple vertical domain and heterogeneous devices to use a common infrastructure stack providing Smart City services. The architecture was used to deploy applications implemented as part of a Smart Energy trial in the Gauteng (South Africa) and a Smart Green City trial in Sant Vicenç dels Horts (Barcelona, Spain) respectively [186]. As a proof of concept, Smart City applications for other domains such as eHealth, Smart Home, Educational M2M system have been developed using the TRESCIMO federated Testbed. Due to the fact that the interconnected testbeds are located at different continents, the connection was subjected to different problems and limitations. Part of these problems is due to the reliability of sea cables. Latency is certainly a

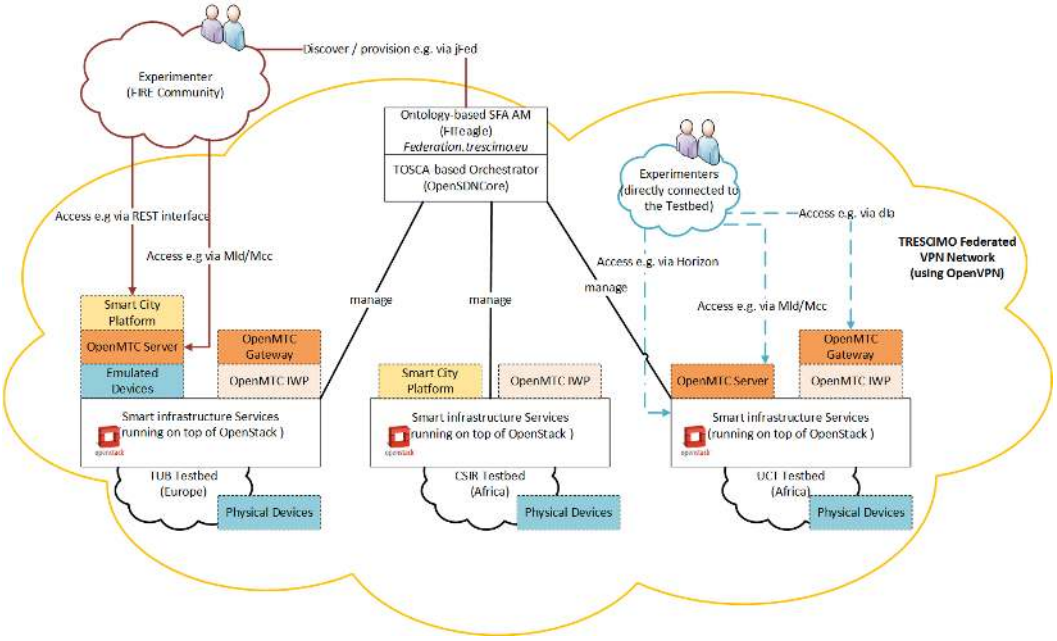


Figure 7.13: TRESCIMO Federated Architecture, based on [185]

problem because of the great distance and number of routers between testbeds and the limitation due to the speed of light. However, this leveraged the realistic behavior of the obtained results [43].

The TRESCIMO Educational M2M platform provides an on demand M2M infrastructure for students to do in class experiments by utilizing resources (i.e. devices and servers) from different testbeds. In an education use case scenario, a university teacher wants to provide an on demand M2M infrastructure for his class, thus enabling the students to experiment M2M state of the art technology. The teacher thereby uses the jFed experimenter client (http://jfed.iminds.be/) developed in the scope of the Fed4Fire project [187]. This client utilizes the Slice Federation Architecture (SFA) interface of the TRESCIMO testbed provided by FITeagle platform [188]. After login with a valid X509 certificate the teacher creates a new experiment and specifies the topology to be used in class. After successfully creating the topology, the teacher then provides the students with details of the accessible resources and their endpoints. Figure 7.14 shows the result of successful provision of selected resources across two testbed site [186].

As part of the Smart Energy System developed as proof-of-concept experiments in the South African context, the CSIR Smart City Platform was used as departure point for the technical implementation. The used sense and actuating devices include a built-in CoAP library used to communicate metering and actuating information. The high-level APIs provided by the Smart City Platform could be used to interact with active components and OpenMTC over HTTP and CoAP. One particular design consideration for applications executing on the Smart City Platform

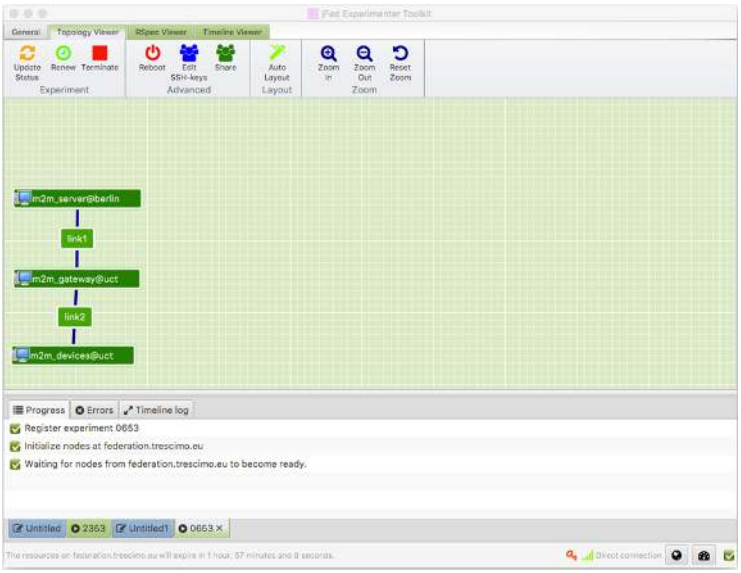


Figure 7.14: Snapshot of Successful Provisioning of Resources in Educational Use Case of TRESCIMO Project[186]

is the potential impact on home-owners. When influencing load utilization through actuation on Internet connected devices (e.g. house-hold appliances), the home-occupant might be adversely affected. This can lead to the home-owner overriding or disabling the devices leading to poorer impact than what was expected [189].

Furthermore, an application for environmental monitoring, named the Green City application, has been deployed in order to provide a visualization tool for the environmental data collected in the TRSCIMO trial at the EU. Data collection is performed by an OpenMTC gateway device installed in a public transportation bus in Sant Vicenç dels Horts-Spain. The gateway is equipped with a DTN-based gateway and allows for bidirectional interaction with the low-power wake-up sensor units to get the reported information and allow the configuration of the devices. The gateway uses JSON coding and HTTP/CoAP for data communication on the uplink direction. For the downlink direction for configuration of the gateway, Lightweight M2M protocol (LWM2M) over CoAP was considered. Several environmental parameters could be monitored using the integrated sensors including temperature, humidity and UV intensity [189]. Each time the bus gets close to a sensor station, the gateway sends a wake-up signal to it, receives the monitored information and forwards it, through the OpenMTC platform, to the remote Smart City Platform. The OpenMTC server, which is located in the TUB Cloud, was used as the M2M platform for this trial. The communication between the data collector and the OpenMTC server is performed through a VPN connection for enabling notifications coming from the OpenMTC platform based on the created subscriptions.

7.6 Comparison with other Solutions

In this section, the OpenMTC platform is assessed against the developments proposed by other researchers as part of individual solutions and standard solutions, which are reviewed in Section 3.5. The approaches are classified based on the core M2M capabilities and functionalities discussed in Chapter 2. For each solution, a summative assessment is presented in Table 7.4 based on the core M2M capabilities and functionalities.

The OpenMTC platform combined with the AdM2M framework fulfill all the functional and non-functional requirements listed in Table 4.4. The overall system provides an end-to-end IP connectivity that could support interoperability between different M2M platforms based on several standardized protocols. The AdM2M framework handles the adaptability of the M2M node to according to the service requirements and connection status by using pluggable transport protocols. The framework is designed to allow extendability by adding more protocols libraries and plugins, as well as interworking proxies. The first implementation of the OpenMTC platform was based on the ETSI M2M release 1 specifications and has been upgraded to the oneM2M specification later on. Nevertheless, the references interfaces of both standards are supported by OpenMTC platform.

Data reporting and notification could be performed in different patterns using OpenMTC. The subscription and notification mechanism, as specified by ETSI and oneM2M, allows applications to define certain criteria to filter the received notification. Furthermore, the SAF function is allowing better control on forwarding the notification streams.

Generally, the reliability and scalability of the OpenMTC platform have been verified by the conducted performance evaluation tests and deployments within several M2M/IoT related research projects.

Table 7.4: Comparison of Reviewed M2M Platforms

Capability	Functionality	Fi-Ware IoT	OpenIoT	BUTLER	OM2M	OpenMTC
Connectivity	Communication selection	Usign Interface to Networks and Devices (I2ND) GEs	✗	✗	✗	✓
	Transport Protocols	HTTP, CoAP, and MQTT	HTTP	HTTP, CoAP /6LowPAN	HTTP and CoAP	HTTPv1, CoAP, [MQTT ongoing work]
	Support of Heterogeneous Devices	IPv4/IPv6, ZigBee, or Z-Wave	✗	ZigBee, NFC	✓	FS20, Wifi, ZigBee, and Bluetooth
	Subscription and Notification	✓	✓	✓	✓	✓
	Cloud Hosting	✓	✓	✓	✗	✓
Device Management	Supported protocols	UL2.0/HTTP, MQTT, OMA LWM2M/CoAP	-	-	-	LWM2M/CoAP
	Localization Management	✓	✓	✓	✗	✓
Application Management	Context Management	Complex Event Management GE	Based on FI-Ware Complex Event Management GE	-	-	interpretable with FI-Ware Complex Event Management GE
	Development tools	-	Virtual Development Kit (OpenIoT-VDK)	Client-side library (BUTLER.js)	-	openMTC SDK
Data Processing	Discovery	Using IoT Discovery GE, and NGSI-9 interface	Semantic discovery	✓	Based on ETSI API	Based on ETSI or oneM2M API

	Data Storage	IoT Broker GE uses the embedded HSQLDB database	Linked Stream Middleware(LSM) cloud storage	NoSQL database (CouchDB)	H2 database	SQL database
	Semantic Annotation	Based on RDF/OWL	W3C SSN compliant RDF format	Using W3C SSN ontology	✗	-
	Data Modelling	OMA NGSI Context Management Information Model	CSV, XML, JSON	JSON	ETSI M2M (XML)	ETSI M2M XML and JSON
Security	Authentication	OAuth 2.0, Based on Security GEs	✓	OAuth have been extended for low cost devices	-	OAuth 2.0
	Recourse Access Control	✓	✓	✓	✓	✓
	Encryption	✓	✗	✓	✗	✗

8 Conclusions and Further Work

This dissertation has extensively studied the M2M communication and related protocols, discussed Smart City services and the impact of M2M communication to realize large-scale deployments, and proposed a novel approach to adaptable M2M communication.

This research work contributed to several research projects such as EU FP7 TRESCIMO and DAAD UNIFI. A large amount of the research in the form of practical realization was included in the Fraunhofer FOKUS OpenMTC toolkit, which is used as the basis for several projects such as FI-PPP FI-WARE, FI-PPP FI-STAR, EU FP7 TRESCIMO, and EU H2020 reTHINK. This chapter summarizes achievements of this work and describes ongoing future extension directions in some of the research projects.

8.1 Summary Overview

This dissertation proposed an adaptable M2M framework (AdM2M) that addresses the need to manage heterogeneous traffic patterns within an M2M system. The objective of the framework is to increase the adaptability of M2M nodes in handling flows of requests from different objects, being resource-constrained or resource-rich, and different applications demanding heterogeneous QoS requirements. The developed adaptable functionality aims to enhance the data transmission process between different M2M entities by dynamically plugging in and out transport protocols to match the network conditions in order to enable a reliable end-to-end data delivery. This work can be considered as a foundation for an M2M-based Smart City platform that addresses the requirements of heterogeneous vertical domains.

In order to ensure the ability of M2M nodes to intercommunicate with as little as possible human intervention, it was important to understand the fundamentals of

M2M communication and different interaction patterns commonly used. In Chapter 2, common interaction models of M2M traffic was illustrated and a conceptual classification of different M2M traffic classes was discussed. The significant level of heterogeneity in M2M networks in terms of capabilities of connected devices and QoS requirements of emerging applications was mentioned. Furthermore, the disseminated estimations of global M2M connections in the next decade imply that M2M systems will be challenged by the number of connected objects as well.

Mainly, M2M applications are driven from enterprise or government needs. While the enterprise focus is on efficiency and cost reduction, the government focus is on sustainability, safety and socio-economic impacts. Thus, features and requirements of M2M applications differ in multiple aspects, and therefore the associated data flows will demand the assignment of different levels of priorities. It might be the case that these requirements differ based on the desire of the user or project owner and the available budget. For the deployment of an M2M system in any domain, such as health care, Smart Home or Industry, it's important to start by specifying the environment conditions and requirements. The understating of operating specifications including the integrated devices, traffic pattern, samples rate, and delay tolerance level, have a great impact on designing the final solution. Many options are available when selecting the components, platform capabilities and access technologies to be used in the system. The work here is motivated by the evolution of the M2M ecosystem towards Smart City realization which presumes a high number of heterogeneous devices communicating anywhere and anytime and interacting with various services.

The main outcome of the research work pursued is a framework to manage heterogeneous M2M traffic produced by a wide range of vertical application domains within a Smart City context. The framework consists of a set of plug-ins to adjust the communication stack used by an M2M node, based on the application requirements and priorities. The objective of the framework is to select the proper communication technology based on the exchanging traffic pattern from sensors/devices and gateways to an M2M server platform. The following contributions to the field of M2M communication have been made as part of this dissertation:

- An adaptable framework for M2M connectivity management that could be extended to support more protocols and IoT systems is specified.
- A validated generic M2M platform that has been deployed on several research projects in the context of Smart City services.
- A classification and analysis of M2M communication interaction models and guideline of M2M protocol usage in different Smart City applications.

In Section 1.5, two main research questions were highlighted. The first question, "*Are the current standardized protocols sufficient to support different kinds of Smart City services including those which comprise real-time M2M streams?*" To answer this question, it was required to analysis both requirements of M2M applications and

capabilities of existing protocols. The requirements and constraints of the communication services required by M2M/IoT systems raise new challenges to the traditional transport protocol layer. Several protocols have been proposed and standardized, each focus on a specific aspect of M2M/IoT communication. A sophisticated protocol able to handle all heterogeneous aspects related to this kind of communication and satisfying requirements of real-time streaming is still missing. Such protocol shall provide mechanisms to support self-configuration and self-adaptability to various network resources (e.g. wireless network systems) and various devices' capabilities. Additionally, it should be light enough to operate on resource-constrained devices. Nevertheless, a persistent connection might be essential for some cases. The approach proposed in this dissertation was to enable M2M nodes to adjust the utilized communication stack by using pluggable protocols libraries, in order to overcome the lack of sufficient protocol to different M2M/IoT applications. The proposed framework aims to enhance the adaptability of M2M nodes, in order to accommodate the needs of M2M communications and applications.

The second research question was: "*to which level could the interoperability of M2M application-agnostic platforms be realized?*". To answer this question, some standards efforts and research projects, which provide a middleware service layer for Smart City applications, have been surveyed. Theoretically, the interoperability and data sharing between different M2M platforms could be realized on multiple levels. The scope of work in this research was limited to connectivity oriented interoperability in large-scale M2M/IoT testbeds by supporting a syntactic level of interoperability. This level of interoperability allows interoperated systems to exchange data and events information, either by using a common protocol structure or by mediating the data format specified by each system. However, both systems are not aware of the meaning of the shared data. For an application-agnostic M2M system, a syntactic level of interoperability is adequate to provide data sharing service to multiple application domains, where the usage and interpretation of the data could vary depending on the applications logic.

Nevertheless, semantic interoperability could also be realized by using ontologies for terminology management in M2M application-agnostic platforms. This level of data-centric interoperability upgrades the M2M platforms to the Internet of Things (IoT) level. Pragmatic interoperability indicates the awareness of data usage and processing in interoperated systems. Therefore, it could be realized in application-specific platform only. From an application-agnostic M2M platform perspective, the semantic interoperability is the highest level to be reached between integrated platforms.

In the course of the research for this dissertation, several related publications have been disseminated that present the work within the topic of interest: eight conference papers reflecting the main contributions ([42, 43, 5, 32, 45, 46, 40, 39]), one book chapter ([190]) and six conference papers about related fields of application in the context of Smart City services ([24, 31, 35, 29, 38, 176]), and seven joint papers with other research projects members ([191, 136, 36, 37, 43, 44, 192]).

8.2 Future Work

M2M communication is an exciting field with outstanding possibilities and various challenges. This section explores some issues that have not been addressed in detail in this work. These issues are beyond the scope of this dissertation, and outline the basis for future work. Figure 8.1 depicts an overview of possible future work subjects based on the achieved framework.

The current wireless communication systems including 3G and LTE cannot satisfy the requirements of highly demanding M2M service such as Smart industry, Smart agriculture, and Smart transportation. The 5G system will combine multiple Radio Access Technology (RAT) networks including LTE, allowing more flexibility in radio resource utilization. This shall enable the prioritization of massive M2M traffic, in order to meet the ultra-low end-to-end latency requirements (less than 5ms). Three different radio access types are considered for Massive Machine Communications (MMC) within 5G systems [193]: i) Direct access (MMC-D) to the access network, ii) access via Accumulation point (MMC-A) that accumulates the traffic locally and send it to the access point, and iii) direct M2M communication (MMC-M) between M2M devices using Device-to-Device (D2D) communication. The MMC-D is suitable for devices with a high level of transmitting power. While the MMC-A type is more common to be used for low-power devices. The accumulation point can either be a dedicated gateway, a smartphone connecting personal devices, or a dynamically selected device. D2D communication requires a very high

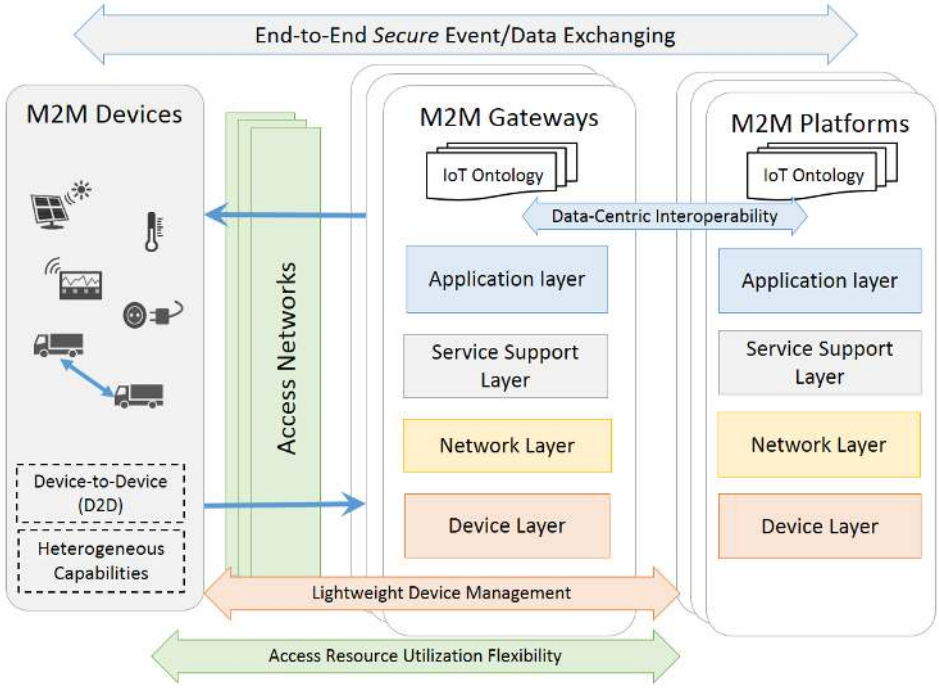


Figure 8.1: Potential Future Work Overview

protocol efficiency (i.e., very low signaling overhead) and devices with long battery life, in order to allow the creation of an ad-hoc mesh network. There are many open research issues to be addressed for the realization of this device-tier interaction [194]. However, it shall bring significant improvements in developing emergency applications and vehicle-to-vehicle (V2V) communication.

The M2M communication is seen to be one of the key drivers to guide the design of 5G network. However, during the work of this dissertation, the focus was on the transport messaging protocols of M2M. The constraints and capabilities rising from various access networks were considered out of scope. Nevertheless, the AdM2M framework could be further developed to cover the adaptability in utilizing communication resources. Initial work in this direction has been started by integrating Device Management protocols such as LWM2M, in order to control the M2M device connectivity with the radio signaling for 5G networks [176].

From the State-of-the-Art analysis presented in this dissertation, the competition of numerous standards in the field of M2M must be mentioned. It will be beneficial to all stakeholders in the M2M market if cooperation in this field would be significantly increased. Therefore, future work should focus on achieving semantic interoperability of M2M systems in order to share the meaning of the exchanged data. Semantic interoperability is achievable through the definition of a common set of ontologies that describe various system entities and the data produced, exchanged, and consumed by these entities. In this direction, the oneM2M partnership project aims to provide data abstraction and semantic interoperability in specification Release 2. The Semantic Sensor Network (SSN) ontology [195] provides the most important core vocabulary for sensing data and defines the notion of sensors and physical devices in general. However, several other initials are proposing ontologies for IoT [196, 197].

Security is an important aspect of IoT and Smart City services, where nearly all exchanged information should not be threatened by any kind of attack. With the quick spreading of IoT throughout home equipment, cars, and even human bodies, new vulnerabilities seem to emerge almost daily [198]. Cryptographic protocols designed for computer networks, such as TLS and DTLS, could be used to create an authenticated and encrypted channel between M2M nodes. However, such protocols have been designed for the Web and require further adaptation to be used on embedded devices with very limited resources.

Furthermore, a general security framework is required to realize the implementation of the "Internet of Secure Things" and avoid privacy violation. The IoT is not a single-use nor single-ownership system, a framework to secure-IoT shall incorporate efficient authentication, authorization, network policy and control components.

Bibliography

[1] Manyika, James, Michael Chui, Jacques Bughin, Richard Dobbs, Peter Bisson, and Alex Marrs: *Disruptive technologies: Advances that will transform life, business, and the global economy.* Technical Report May, San Francisco, 2013. http://www.mckinsey.com/insights/business{_}technology/disruptive{_}technologies.

[2] Intel: *Infographic: Guide to The Internet of Things.* http://www.intel.com/content/www/us/en/intelligent-systems/iot/internet-of-things-infographic.html.

[3] Giffinger, Rudolf, Christian Fertner, Hans Kramar, Robert Kalasek, Nataša Pichler-Milanović, and Evert Meijers: *Smart Cities: Ranking of European Medium-Sized Cities*, 2007. http://www.smart-cities.eu/download/smart_cities_final_report.pdf, visited on 07/11/12.

[4] Naphade, Milind, Guruduth Banavar, Colin Harrison, Jurij Paraszczak, and Robert Morris: *Smarter Cities and Their Innovation Challenges.* IEEE Computer, 44(6):32–39, 2011. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5875937&tag=1.

[5] Elmangoush, Asma, Hakan Coskun, Sebastian Wahle, and Thomas Magedanz: *Design Aspects for a Reference M2M Communication Platform for Smart Cities.* In *2013 9th International Conference on Innovations in Information Technology (IIT)*, pages 204–209, Abu Dhabi, mar 2013. IEEE, ISBN 978-1-4673-6203-0. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6544419.

[6] Tselikas, Nikolaos D., George S. Tselikis, and Nikos C. Sagias: *Software and Middleware Technologies Based on Open APIs and Protocols for Modern Service Provision in Telecoms.* In *2010 14th Panhellenic Conference on Informatics*, volume 33-37, pages 33–37. IEEE, September 2010,

ISBN 978-1-4244-7838-5. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5600290.

[7] Chen, Kwang Cheng and Shao Yu Lien: *Machine-to-machine communications: Technologies and challenges.* Ad Hoc Networks, 18(0):3 – 23, 2014, ISSN 1570-8705. http://www.sciencedirect.com/science/article/pii/S1570870513000395.

[8] Shafiq, Muhammad Zubair, Lusheng Ji, Alex X. Liu, Jeffrey Pang, and Jia Wang: *A First Look at Cellular Machine-to-Machine Traffic - Large Scale Measurement and Characterization.* In *Proc. 12th ACM SIGMETRICS/PERFORMANCE Jt. Int. Conf. Meas. Model. Comput. Syst.*, volume 40, page 65, London, 2012. ACM, ISBN 978-1-4503-1097-0.

[9] Mikóczy, Eugen, Ivan Kotuliak, and Oskar Van Deventer: *Evolution of the Converged NGN Service Platforms Towards Future Networks.* Future Internet, 3(1):67–86, March 2011, ISSN 1999-5903. http://www.mdpi.com/1999-5903/3/1/67/.

[10] Nikaein, Navid, Mahesh K Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, Christian Bonnet, and Sophia Antipolis: *OpenAirInterface : A Flexible Platform for 5G Research.* ACM SIGCOMM Computer Communication Review, 44(5):33–38, 2014.

[11] Zorzi, Michele, Alexander Gluhak, Sebastian Lange, and Alessandro Bassi: *From Today's INTRAnet of Things to A Future INTERnet of Things: A Wireless - and Mobility-Related View.* IEEE Wirel. Commun., 17(December):44–51, 2010. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5675777&tag=1.

[12] Fi-Ware Project: *Fi-Ware (Core Platform of the Future Internet).* http://www.fi-ware.eu/.

[13] FP7-BUTLER: *uBiquitous, secUre inTernet-of-things with Location and contExt-awaReness (BUTLER).* http://www.iot-butler.eu/, visited on 2015-03-20.

[14] 3GPP-TS22.368: *Service requirements for Machine-Type Communications (MTC) Rel-12*, 2013.

[15] ITU-T Y.2060: *Overview of the Internet of Things*, 2012.

[16] IERC: *IERC-European Research Cluster on the Internet of Things.* http://www.internet-of-things-research.eu/about_iot.htm, visited on 2015-04-16.

[17] *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*, 1991.

[18] Harrison, C, B Eckman, R Hamilton, P Hartswick, J Kalagnanam, J Paraszczak, and P Williams: *Foundations for Smarter Cities*. IBM Journal of Research and Development, 54(4):1–16, July 2010, ISSN 0018-8646. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5512826.

[19] Anton-Haro, Carles and Mischa Dohler (editors): *Machine-to-Machine (M2M) Communications: Architecture, Performance and Applications*. Elsevier Science, 2014, ISBN 1782421106.

[20] Webb, William: *Weightless: The Technology to Finally Realise the M2M Vision*. Int. J. Interdiscip. Telecommun. Netw., 4(2):30–37, January 2012, ISSN 1941-8663. http://www.igi-global.com/article/weightless-technology-finally-realise-m2m/67575.

[21] SIGFOX: *SigFox*. http://www.sigfox.com/en/.

[22] Uckelmann, Dieter, Mark Harrison, and Florian Michahelles (editors): *Architecting the Internet of Things*. Springer, 2011, ISBN 9783642191565.

[23] ITU-T: *ITU-T Focus Groups*. http://www.itu.int/en/ITU-T/focusgroups/Pages/default.aspx.

[24] Elmangoush, Asma, Hakan Coskun, Sebastian Wahle, Niklas Blum, and Thomas Magedanz: *Promoting M2M Application Development for Smart City*. In *Wireless World Research Forum Meeting 29 (WWRF)*, Berlin, 2012.

[25] Hauer, Jan Hinrich, Vlado Handziski, Andreas Köpke, Andreas Willig, and Adam Wolisz: *A Component Framework for Content-Based Publish/Subscribe in Sensor Networks*. In Verdone, Roberto (editor): *Wireless Sensor Networks*, volume 4913 of *Lecture Notes in Computer Science*, pages 369–385. 2008, ISBN 978-3-540-77689-5. http://link.springer.com/chapter/10.1007/978-3-540-77690-1_23.

[26] Borgia, Eleonora: *The Internet of Things vision: Key features, applications and open issues*. Computer Communications, 54(October):1–31, October 2014, ISSN 01403664. http://www.sciencedirect.com/science/article/pii/S0140366414003168http://linkinghub.elsevier.com/retrieve/pii/S0140366414003168.

[27] Ma, Hua Dong: *Internet of Things: Objectives and Scientific Challenges*. Journal of Computer Science and Technology, 26(6):919–924, 2011, ISSN 10009000. http://www.springerlink.com/index/10.1007/s11390-011-1189-5.

[28] Gubbi, Jayavardhana, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami: *Internet of things (IoT): A Vision, Architectural Elements, and Future Directions*. Future Generation Computer Systems, 29(7):1645–1660, September 2013, ISSN 0167739X. http://linkinghub.elsevier.com/retrieve/pii/S0167739X13000241.

[29] Corici, Andreea, Asma Elmangoush, Ronald Steinke, Thomas Magedanz, Joyce Mwangama, and Neco Ventura: *Utilizing M2M Technologies for Building Reliable Smart Cities.* In *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, March 2014, ISBN 978-1-4799-3223-8. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6814059.

[30] Boccardi, Federico, Robert W. Heath, Angel Lozano, Thomas L. Marzetta, and Petar Popovski: *Five disruptive technology directions for 5G.* IEEE Communications Magazine, 52(2):74–80, February 2014, ISSN 0163-6804. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6736746.

[31] Elmangoush, Asma, Thomas Magedanz, Alexander Blotny, and Niklas Blum: *Design of RESTful APIs for M2M Services.* In *16th International Conference on Intelligence in Next Generation Networks*, pages 50–56, Berlin, 2012. ISBN 9781467315265.

[32] Elmangoush, Asma, Adel Al-Hezmi, and Thomas Magedanz: *The Development of M2M Standards for Ubiquitous Sensing Service Layer.* In *2014 IEEE Globecom Workshops (GC Wkshps)*, pages 624–629. IEEE, December 2014, ISBN 9781479974702. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7063502.

[33] Kosolworrawattanakul, Napat, Asma Elmangoush, Thomas Magedanz, and Chaodit Aswakul: *Development of Real-Time Data Synchronization for IEEE1888 and ETSI M2M Standards.* In *IEICE Technical Report*, pages 79–84, 2014. http://www.ieice.org/~ia/IA2014/wiki.cgi.

[34] Elmangoush, Asma, Ronald Steinke, Adel Al-hezmi, and Thomas Magedanz: *On The Usage of Standardised M2M Platforms for Smart Energy Management.* In *28th International Conference on Information Networking (ICOIN)*, pages 79–84, Phuket, February 2014. IEEE, ISBN 9781479936892. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6799669.

[35] Elmangoush, Asma, Adel Al-hezmi, and Thomas Magedanz: *Towards Standard M2M APIs for Cloud-based Telco Service Platforms.* In *Proceedings of International Conference on Advances in Mobile Computing & Multimedia - MoMM '13*, pages 143–149. ACM Press, 2013, ISBN 9781450321068. http://dl.acm.org/citation.cfm?doid=2536853.2536892.

[36] Mwangama, Joyce, Alexander Willner, Neco Ventura, Asma Elmangoush, Tom Pfeifer, and Thomas Magedanz: *Testbeds for Reliable Smart City Machine-to-Machine Communication.* In *Southern African Telecommunication Networks and Applications Conference (SATNAC)*, pages 339–344, South Africa, 2013. ISBN 978-0-620-57883-7. http://www.satnac.org.za/proceedings/2013/SATNAC{_}2013{_}Conference{_}Proceedings.pdf.

[37] Mwangama, Joyce, Asma Elmangoush, Joseph Orimolade, Neco Ventura, Ronald Steinke, Alexander Willner, Andreea Ancuta Corici, and Thomas Magedanz: *Prototyping Machine-to-Machine Applications for Emerging Smart Cities in Developing Countries.* In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2014*, pages 383–388, aug 2014, ISBN 978-0-620-61965-3. http://www.satnac.org.za/proceedings/2014/SATNAC2014ConferenceProceedings{_}USB{_}edition.pdf.

[38] Elmangoush, Asma, Andreea Corici, Marisa Catalan, Ronald Steinke, Thomas Magedanz, and Joaquim Oller: *Interconnecting Standard M2M Platforms to Delay Tolerant Networks.* In *2014 International Conference on Future Internet of Things and Cloud*, pages 258–263, Barcelona, 2014. IEEE, ISBN 978-1-4799-4358-6. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6984204.

[39] Elmangoush, Asma, Ronald Steinke, Thomas Magedanz, Andreea Ancuta Corici, Alex Bourreau, and Adel Al-Hezmi: *Application-derived communication protocol selection in M2M platforms for smart cities.* In *2015 18th International Conference on Intelligence in Next Generation Networks*, pages 76–82, Paris, France, feb 2015. IEEE, ISBN 978-1-4799-1866-9. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7073810.

[40] Elmangoush, Asma, Andreea Ancuta, Ronald Steinke, Marius Corici, and Thomas Magedanz: *A Framework for Handling Heterogeneous M2M Traffic.* Procedia Computer Science, pages 0–7, 2015.

[41] *OpenMTC platform.* http://www.open-mtc.org/index.html.

[42] Corici, Marius, Hakan Coskun, Asma Elmangoush, Agus Kurniawan, Tong Mao, Thomas Magedanz, and Sebastian Wahle: *OpenMTC : Prototyping Machine Type Communication in Carrier Grade Operator Networks.* In *4th International IEEE Workshop on Open NGN and IMS Testbeds (ONIT 2012) @ GLOBECOM 2012*, pages 1735 – 1740, Anaheim, CA, 2012. IEEE, ISBN 978-1-4673-4942-0. http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6477847.

[43] Corici, Andreea Ancuta, Asma Elmangoush, Thomas Magedanz, Ronald Steinke, Joyce Mwangama, and Neco Ventura: *An OpenMTC platform-based interconnected European: South African M2M Testbed for Smart City Services.* In *the first International Conference on the use of Mobile Informations and Communication Technology (ICT) in Africa - UMICTA 2014*, pages 35–39, Stellenbosch, 2014. ISBN 978-0-7972-1533-7.

[44] Klinpratum, Teerapan, Chaiyachet Saivichit, Asma Elmangoush, and Thomas Magedanz: *Performance of Interworking Proxy for Interconnecting IEEE1888 Standard at ETSI M2M Platforms.* Applied Mechanics and Materials, 781:141–144, 2015.

[45] Klinpratum, Teerapan, Chaiyachet Saivichit, Asma Elmangoush, and Thomas Magedanz: *Toward Interconnecting M2M / IoT Standards : Interworking Proxy for IEEE1888 Standard at ETSI M2M Platforms.* In *29th International Technical Conference on Circuit/Systems Computers and Communications (ITC-CSCC 2014)*, pages 763–766, Phuket, Thailand, 2014.

[46] Elmangoush, Asma, Ronald Steinke, and Thomas Magedanz: *AdM2M: Adaptable Machine-to-Machine Transport Framework.* In *The 17th International Conference on Information Integration and Web-based Applications & Services (iiWAS2015)*, pages 331–335, Brussels, dec 2015. ACM, ISBN 9781450334914.

[47] Lawton, George: *Machine-to-machine technology gears up for growth.* Computer, 37(9):12–15, 2004, ISSN 00189162. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1332996.

[48] Choi, Minkyu and Ronnie D Caytiles: *A Proposed Integration of Hierarchical Mobile IP based Networks in SCADA Systems.* International Journal of Smart Home, 7(5):49–56, 2013.

[49] *IEEE 802.16's M2M Task Group.* http://wirelessman.org/m2m/index.html, visited on 16/04/14.

[50] ETSI TS 102 690 v1.1.1: *Machine-to-Machine communications (M2M); Functional architecture.* Technical report, 2011.

[51] Mazhelis, Oleksiy, Eetu Luoma, and Henna Warma: *Defining an Internet-of-Things Ecosystem.* In *Internet of Things, Smart Spaces, and Next Generation Networking*, pages 1–14. Springer Berlin Heidelberg, 2012. http://link.springer.com/chapter/10.1007/978-3-642-32686-8_1#http://link.springer.com/10.1007/978-3-642-32686-8_1.

[52] OneM2M TR 0001: *oneM2M Use cases collection*, 2013.

[53] Mattern, Friedemann and Christian Floerkemeier: *From the Internet of Computers to the Internet of Things.* Volume 6462 of *Lecture Notes in Computer Science*, pages 242–259. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, ISBN 978-3-642-17225-0. http://www.springerlink.com/index/10.1007/978-3-642-17226-7.

[54] Hongsong, Chen, Fu Zhongchuan, and Zhang Dongyan: *Security and Trust Research in M2M System.* In *2011 IEEE Int. Conf. Veh. Electron. Saf.*, number 20090460245, pages 286–290, 2011, ISBN 9781457705779.

[55] Nikaein, Navid, Markus Laner, Kaijie Zhou, Philipp Svoboda, Dejan Drajic, Milica Popovic, and Srdjan Krco: *Simple Traffic Modeling Framework for Machine Type Communication.* In *The Tenth International Symposium on Wireless Communication Systems*, pages 783–787, 2013, ISBN 9783800735297.

[56] 3GPP TR 37.868: *Study on RAN Improvements for Machine-type communications (Release 11)*, 2011. http://www.qtc.jp/3GPP/Specs/37868-b00.pdf.

[57] Laner, Markus, Philipp Svoboda, Navid Nikaein, and Markus Rupp: *Traffic Models for Machine Type Communications.* In *Wireless Communication Systems (ISWCS 2013), Proceedings of the Tenth International Symposium on*, pages 651–655, Ilmenau, Germany, 2013. ISBN 978-3-8007-3529-7. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6629817&isnumber=6629683.

[58] Gotsis, Antonis G., Athanasios S. Lioumpas, and Angeliki Alexiou: *Evolution of packet scheduling for Machine-Type communications over LTE: Algorithmic design and performance analysis.* In *2012 IEEE Globecom Workshops, GC Wkshps 2012*, pages 1620–1625. Ieee, December 2012, ISBN 9781467349413. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6477828.

[59] Nie, Yaling and Yanchen Ma: *A First Look at AMI Traffic Patterns and Traffic Surge for Future Large Scale Smart Grid Deployments.* In *INFOCOMP 2012 : The Second International Conference on Advanced Communications and Computation*, pages 120–124, 2012, ISBN 9781612082264.

[60] Al-Khatib, Obada, Wibowo Hardjawana, and Branka Vucetic: *Traffic modeling for Machine-to-Machine (M2M) last mile wireless access networks.* In *2014 IEEE Global Communications Conference*, pages 1199–1204. IEEE, December 2014, ISBN 978-1-4799-3512-3. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7036972.

[61] Chihani, Bachir, Emmanuel Bertin, and Noël Crespi: *Enhancing M2M communication with cloud-based context management.* In *Proceedings - 6th International Conference on Next Generation Mobile Applications, Services, and Technologies, NGMAST 2012*, pages 36–41. Ieee, September 2012, ISBN 9780769548036. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6327928.

[62] Osawa, Tatsuzo: *Practice of m2m connecting real-world things with cloud computing.* FUJITSO Sci. Tech. J, 47(4):401–407, 2011.

[63] Issarny, Valérie, Nikolaos Georgantas, Sara Hachem, Apostolos Zarras, Panos Vassiliadist, Marco Autili, Marco Aurélio Gerosa, and Amira Ben Hamida: *Service-oriented middleware for the Future Internet: state of the art and research directions.* Journal of Internet Services and Applications, 2(1):23–45, May 2011, ISSN 1867-4828. http://www.springerlink.com/index/10.1007/s13174-011-0021-3.

[64] Shelby, Z, K Hartke, and C Bormann: *RFC 7252: The Constrained Application Protocol (CoAP)*, 2014.

[65] International Business Machines Corporation (IBM): *MQTT V3.1 Protocol Specification*. Technical report, 2010. http://mqtt.org/.

[66] Langhammer, Nils and Ruediger Kays: *Performance Evaluation of Wireless Home Automation Networks in Indoor Scenarios*. IEEE Transactions on Smart Grid, 3(4):2252–2261, December 2012, ISSN 1949-3053. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6287624.

[67] OMA-TS-LightweightM2M-V1: *Lightweight Machine to Machine Technical Specification*. Technical report, 2013.

[68] Cardoso, Jorge, Konrad Voigt, and Matthias Winkler: *Service Engineering for the Internet of Services*. In *Enterprise Information Systems*, pages 15–27. 2009.

[69] Severi, Stefano, Giuseppe Abreu, Friedbert Berens, Claudio Pastrone, Francesco Sottile, and Maurizio Spirito: *M2M Technologies: Enablers for a Pervasive Internet of Things*. In *The European Conference on Networks and Communications (EUCNC2014)*, pages 1–5, Bologna, 2014. IEEE, ISBN 9781479952809. https://www.academia.edu/6866526/M2M_Technologies_Enablers_for_a_Pervasive_Internet_of_Thingshttp://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6882661&tag=1.

[70] Perera, Charith, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos: *Context Aware Computing for The Internet of Things : A Survey*. IEEE COMMUNICATIONS SURVEYS & TUTORIALS, 16(1):414 – 454, 2014.

[71] Barnaghi, Payam, W E I Wang, Cory Henson, and Kerry Taylor: *Semantics for the Internet of Things : early progress and back to the future*. International Journal on Semantic Web and Information Systems (IJSWIS), 8(1):1–21, 2012.

[72] Lu, Rongxing, Xu Li, Xiaohui Liang, Xuemin Shen, and Xiaodong Lin: *Grs: The green, reliability, and security of emerging machine to machine communications*. Communications Magazine, IEEE, 49(4):28–35, April 2011, ISSN 0163-6804.

[73] Swetina, J., G. Lu, P. Jacobs, F. Ennesser, and J. Song: *Toward a standardized common M2M service layer platform: Introduction to oneM2M*. IEEE Wireless Communications, 21(3):20–26, June 2014, ISSN 1536-1284. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6845045.

[74] Taneja, Mukesh: *Lightweight security protocols for smart metering*. In *2013 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*, pages 1–5. IEEE, November 2013, ISBN 978-1-4799-1347-3. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6698743.

[75] ITU-T Y.2061: *Requirements for the support of machine-oriented communication applications in the next generation network environment*, 2012.

[76] ITU-T Y.2062: *Framework of Object-to-Object Communication for Ubiquitous Networking in Next Genration Networks*, 2012.

[77] ITU-T: *Internet of Things Global Standards Initiative*. http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx.

[78] ITU-T: *FG M2M*. http://www.itu.int/en/ITUT/focusgroups/m2m/Pages/default.aspx.

[79] OneM2M: *OneM2M*. http://onem2m.org/.

[80] OneM2M-TS-0001: *OneM2M Functional Architecture*, 2015. http://onem2m.org/technical/published-documents.

[81] *OMA Next Gereration Servicres Interface V1.0*. http://technical.openmobilealliance.org/Technical/release_program/ngsi_v1_0.aspx.

[82] OMA-TS-DM_Protocol-V2: *OMA Device Management Protocol V2.0*, 2013.

[83] IEEE-IoT: *IEEE-SA - Internet of Things*. http://standards.ieee.org/innovate/iot/stds.html, visited on 2015-07-29.

[84] IEEE Std 1888: *IEEE Standard for Ubiquitous Green Community Control Network Protocol*, 2013, ISBN 9780738165530.

[85] Ochiai, Hideya: *Power data management on the Internet space: Green ICT projects in Japan*. In *2012 IEEE Colombian Communications Conference (COLCOM)*, pages 1–2. IEEE, may 2012, ISBN 978-1-4673-1269-1. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6233680.

[86] Ninagawa, Chuzo, Hiroki Yoshida, Seiji Kondo, and Hiroyuki Otake: *Data transmission of IEEE1888 communication for wide-area real-time smart grid applications*. 2013 International Renewable and Sustainable Energy Conference (IRSEC), pages 509–514, mar 2013. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6529702.

[87] Dong, Liu and Gu Chen: *IEEE Standard 1888: Green Community Infrastructure & Protocol*. IEEE Standards Education e-Magazine, 2012. https://ieee-elearning.org/outreach/mod/book/view.php?id=3276&chapterid=257.

[88] IEEE-SA: *IEEE P2413 WG*. http://grouper.ieee.org/groups/2413/, visited on 2015-07-29.

[89] Minerva, Roberto, Abyi Biru, and Domenico Rotondi: *Towards a definition of the Internet of Things (IoT).* http://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf, 2015.

[90] Vermesan, Ovidiu and Peter Friess (editors): *Internet of Things - From Research and Innovation to Market Deployment.* River Publishess, 2014, ISBN 9788793102941.

[91] v1.1.1, ETSI TS103.092: *Machine-to-Machine communications (M2M); OMA DM compatible Management Objects for ETSI M2M*, 2012.

[92] Palattella, Maria Rita, Nicola Accettura, Xavier Vilajosana, Thomas Watteyne, Luigi Alfredo Grieco, Senior Member, Gennaro Boggia, and Mischa Dohler: *Standardized Protocol Stack for the Internet of (Important) Things.* IEEE COMMUNICATIONS SURVEYS & TUTORIALS, 15(3):1389–1406, 2013. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6380493.

[93] Berners-Lee, Tim, R. Fielding, and H. Frystyk: *RFC 1945: Hypertext Transfer Protocol – HTTP/1.0*, 1996. http://www.rfc-base.org/txt/rfc-1945.txt.

[94] Ludovici, Alessandro, Pol Moreno, and Anna Calveras: *TinyCoAP: A Novel Constrained Application Protocol (CoAP) Implementation for Embedding RESTful Web Services in Wireless Sensor Networks Based on TinyOS.* Journal of Sensor and Actuator Networks, 2(2):288–315, May 2013, ISSN 2224-2708. http://www.mdpi.com/2224-2708/2/2/288/.

[95] Fielding, R., J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and Tim Berners-Lee: *Hypertext Transfer Protocol – HTTP/1.1.* Technical report. http://www.w3.org/Protocols/rfc2616/rfc2616.html.

[96] Belshe, Mike, Roberto Peon, and Martin Thomson: *RFC 7540: Hypertext Transfer Protocol Version 2 (HTTP/2)*, 2015. https://tools.ietf.org/html/rfc7540.

[97] OASIS: *AMQP - Advanced Message Queuing Protocol V1.0.* http://docs.oasis-open.org/amqp/core/v1.0/amqp-core-complete-v1.0.pdf.

[98] Luzuriaga, Jorge E., Miguel Perez, Pablo Boronat, Juan Carlos Cano, Carlos Calafate, and Pietro Manzoni: *Testing AMQP Protocol on Unstable and Mobile Networks.* In Fortino, Giancarlo, Giuseppe Di Fatta, Wenfeng Li, Sergio Ochoa, Alfredo Cuzzocrea, and Mukaddim Pathan (editors): *Internet and Distributed Computing Systems*, volume 8729 of *Lecture Notes in Computer Science*, pages 250–260. Springer International Publishing, Cham, 2014, ISBN 978-3-319-11691-4. http://link.springer.com/10.1007/978-3-319-11692-1http://link.springer.com/10.1007/978-3-319-11692-1_22.

[99] Rodríguez-Domínguez, Carlos, Kawtar Benghazi, Manuel Noguera, José Luis Garrido, María Luisa Rodríguez, and Tomás Ruiz-López: *A communication model to integrate the Request-Response and the Publish-Subscribe paradigms into ubiquitous systems.* Sensors (Basel, Switzerland), 12(6):7648–7668, January 2012, ISSN 1424-8220. http://www.mdpi.com/1424-8220/12/6/7648/htm.

[100] Hunkeler, Urs, Hong Linh Truong Hong Linh Truong, and Andy Stanford-Clark: *MQTT-S A publish/subscribe protocol for Wireless Sensor Networks.* In *2008 3rd Int. Conf. Commun. Syst. Softw. Middlew. Work. (COMSWARE '08)*, pages 791–798. IEEE, January 2008, ISBN 978-1-4244-1796-4. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4554519.

[101] Davis, Ernesto García, Anna Calveras, and Ilker Demirkol: *Improving packet delivery performance of publish/subscribe protocols in wireless sensor networks.* Sensors (Basel, Switzerland), 13(1):648–680, January 2013, ISSN 1424-8220. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3574696&tool=pmcentrez&rendertype=abstract.

[102] Colitti, W., K. Steenhaut, N. De Caro, B. Buta, and V. Dobrota: *Evaluation of constrained application protocol for wireless sensor networks.* In *18th IEEE Workshop on18th IEEE Workshop on Local & Metropolitan Area Networks (LANMAN), 2011*, pages 1–6, 2011. http://ieeexplore.ieee.org/ielx5/6068505/6076913/06076934.pdf?tp=&arnumber=6076934&isnumber=6076913.

[103] 3GPP-TS23.682: *Architecture enhancements to facilitate communications with packet data networks and applications (Release 11)*, 2013.

[104] Jain, P., P. Hedman, and H. Zisimopoulos: *Machine type communications in 3gpp systems.* Communications Magazine, IEEE, 50(11):28–35, November 2012, ISSN 0163-6804.

[105] 3GPP-TR23.888: *System Improvements for Machine-Type Communications; (Release 11)*, 2012.

[106] Ratasuk, Rapeepat, Athul Prasad, Zexian Li, Amitava Ghosh, and Mikko Uusitalo: *Recent advancements in M2M communications in 4G networks and evolution towards 5G.* In *2015 18th International Conference on Intelligence in Next Generation Networks*, pages 52–57. IEEE, 2015, ISBN 978-1-4799-1866-9. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7073806.

[107] Jiang, Daniel and Luca Delgrossi: *IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments.* In *VTC Spring 2008 - IEEE Vehicular Technology Conference*, pages 2036–2040. IEEE,

May 2008, ISBN 978-1-4244-1644-8. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4526014.

[108] Hazmi, Ali, Jukka Rinne, and Mikko Valkama: *Feasibility Study of IEEE 802.11ah Radio Technology for IoT and M2M Use Cases.* In *2012 IEEE Globecom Workshops*, pages 1687–1692. IEEE, December 2012, ISBN 978-1-4673-4941-3. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6477839.

[109] Yan Zhang, Rong Yu, Shengli Xie, Wenqing Yao, Yang Xiao, and M Guizani: *Home M2M networks: Architectures, Standards, and QoS improvement.* IEEE Communications Magazine, 49(4):44–52, apr 2011, ISSN 0163-6804. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5741145.

[110] Siekkinen, Matti, Markus Hiienkari, Jukka K. Nurminen, and Johanna Nieminen: *How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4.* In *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 232–237. IEEE, apr 2012, ISBN 978-1-4673-0682-9. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6215496.

[111] Hui, Jonathan and Pascal Thubert: *RFC 6282: Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*, 2011. https://tools.ietf.org/html/rfc6282.

[112] Isomaki, Markus, Johanna Nieminen, Carles Gomez, Zach Shelby, Teemu Savolainen, and Basavaraj Patil: *RFC 7668: IPv6 over BLUETOOTH(R) Low Energy*, 2015. https://tools.ietf.org/html/rfc7668.

[113] Bai, Enjian and Xiaokui Zhang: *Performance Evaluation of 6LoWPAN Gateway Used in Actual Network Environment.* In *2012 Int. Conf. Control Eng. Commun. Technol.*, pages 1036–1039. IEEE, dec 2012, ISBN 978-1-4673-4499-9. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6413761.

[114] Withanage, Chathura, Rahul Ashok, Chau Yuen, and Kevin Otto: *A Comparison of the Popular Home Automation Technologies.* In *2014 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*, pages 600–605. IEEE, may 2014, ISBN 978-1-4799-1300-8. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6873860.

[115] Bassi, Alessandro, Stefan Meissner, Martin Bauer, Martin Fiedler, Thorsten Kramp, Rob van Kranenburg, and Sebastian Lange (editors): *Enabling Things to Talk - Designing IoT solutions with the IoT Architectural Reference Model.* Springer, 2013, ISBN 9783642404023.

[116] IoT-A Project: *Internet of Things-Architecture (IoT-A) Deliverable D1.5: Final architectural reference model for the IoT v3.0*, 2013.

[117] Krco, Srdjan, Boris Pokric, and Francois Carrez: *Designing IoT architecture(s): A European perspective*, 2014.

[118] Fi-Ware Project: *Internet of Things (IoT) Services Enablement Architecture.* https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Internet_of_Things_%28IoT%29_Services_Enablement_Architecture.

[119] *OpenIoT project | Open Source Solution for the Internet of Things into the Cloud.* http://www.openiot.eu/, visited on 2015-03-20.

[120] Soldatos, John, Nikos Kefalakis, Manfred Hauswirth, Martin Serrano, Jean Paul Calbimonte, Mehdi Riahi, Karl Aberer, Prem Prakash Jayaraman, Arkady Zaslavsky, Ivana Podnar Žarko, Lea Skorin-Kapov, and Reinhard Herzog: *OpenIoT: Open Source Internet-of-Things in the Cloud.* In Podnar Žarko, Ivana, Krešimir Pripužić, and Martin Serrano (editors): *Interoperability and Open-Source Solutions for the Internet of Things*, volume 9001 of *Lecture Notes in Computer Science*, pages 13–25. Springer International Publishing, Cham, 2015, ISBN 978-3-319-16545-5. http://link.springer.com/10.1007/978-3-319-16546-2http://link.springer.com/10.1007/978-3-319-16546-2_3.

[121] Vermesan, Ovidiu and Peter Friess (editors): *The Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems.* River Publishess, 2013, ISBN 9788792982735.

[122] Bhatti, Zubair Wadood, Nayyab Naqvi, Arun Ramakrishnan, Davy Preuveneers, and Yolande Berbers: *Learning Distributed Deployment and Configuration Trade-offs for Context-Aware Applications in Intelligent Environments.* Journal of Ambient Intelligence and Smart Environments, 6(5):541–559, 2014. https://lirias.kuleuven.be/handle/123456789/462726.

[123] Alaya, M. Ben, Y. Banouar, T. Monteil, C. Chassot, and K. Drira: *OM2M: Extensible ETSI-compliant M2M Service Platform with Self-configuration Capability.* Procedia Comput. Sci., 32:1079–1086, 2014, ISSN 18770509. http://linkinghub.elsevier.com/retrieve/pii/S1877050914007364.

[124] Serrano, Martin, Hoan Nguyen Mau Quoc, Danh Le Phuoc, Manfred Hauswirth, John Soldatos, Nikos Kefalakis, Prem Prakash Jayaraman, and Arkady Zaslavsky: *Defining the Stack for Service Delivery Models and Interoperability in the Internet of Things: A Practical Case With OpenIoT-VDK.* IEEE Journal on Selected Areas in Communications, 33(4):676–689, April 2015, ISSN 0733-8716. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7012040.

[125] Fi-Star Project: *Fi-Star (Future Internet Social and Technological Alignment Research in Healthcare).* https://www.fi-star.eu/home.html.

[126] Project, XIFI: *FI-XIFI Project*. https://fi-xifi.eu/home.html, visited on 2015-08-19.

[127] *FRACTALS*. http://fractals-fp7.com/, visited on 2015-09-02.

[128] University of Michigan: *Urbanization and Global Change*. http://www.globalchange.umich.edu/globalchange2/current/lectures/urban_gc/, visited on 2015-02-10.

[129] ISO/IEC JTC 1: *Smart cities-Preliminary Report*. Technical report, 2015. http://www.iso.org/iso/smart_cities_report-jtc1.pdf.

[130] ETSI TS 102 689 V1.1.2: *Machine-to-Machine communications (M2M); M2M service requirements*. 2011.

[131] IEEE802.16: *Machine to Machine (M2M) Communications Technical Report*, 2010. http://ieee802.org/16/m2m/#10_0005.

[132] Boswarthick, David, Omar Elloumi, and Olivier Hersent: *M2M Communications: A Systems Approach*. John Wiley & Sons, 2012, ISBN 1119940966. http://books.google.com/books?hl=en&lr=&id=bVaqAFpH6EgC&pgis=1.

[133] Beecham Research: *Towards Smart Cities and The Role of M2M Technologies*. Technical report, 2014. http://m2msummit.pl/wp-content/uploads/sites/6/2014/10/brl-m2m-towards-smart-cities-wp-2014-sep.pdf.

[134] Nikaein, Navid and Srdjan Krco: *Latency for Real-Time Machine-to-Machine Communication in LTE-Based System Architecture*. In *Wireless Conference 2011 - Sustainable Wireless Technologies (European Wireless), 11th European*, volume 7, pages 263–268, Vienna, April 2011. IEEE, ISBN 9783800733439. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5898065&tag=1.

[135] Kreps, Gary and Linda Neuhauser: *New directions in eHealth communication: opportunities and challenges*. Patient education and counseling, 78(3):329–36, March 2010, ISSN 1873-5134. http://www.ncbi.nlm.nih.gov/pubmed/20202779.

[136] Suryani, Vera, Achmad Rizal, Anton Herutomo, Maman Abdurohman, Thomas Magedanz, and Asma Elmangoush: *Electrocardiagram monitoring on OpenMTC platform*. In *38th Annual IEEE Conference on Local Computer Networks - Workshops*, number 2, pages 843–847, Sydney, NSW, Oct 2013. IEEE, ISBN 978-1-4799-0540-9. http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6758521{&}tag=1http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6758521.

[137] The European Commission: *eHealth Action Plan 2012-2020 - Innocation healthcare for the 21st century.* Technical report, 2012. http://ec.europa.eu/digital-agenda/en/news/ehealth-action-plan-2012-2020-innovative-healthcare-21st-century.

[138] Schweitzer, Julian and Christina Synowiec: *The Economics of eHealth and mHealth.* J. Health Commun., 17(sup1):73–81, 2012, ISSN 1081-0730.

[139] Lewis, Trevor, Christina Synowiec, Gina Lagomarsino, and Julian Schweitzer: *E-health in low- and middle-income countries: findings from the Center for Health Market Innovations.* Bulletin of the World Health Organization, 90(5):332–340, 2012, ISSN 00429686.

[140] Bashshur, Rashid, Gary Shannon, Elizabeth Krupinski, and Jim Grigsby: *The taxonomy of telemedicine.* Telemedicine journal and e-health : the official journal of the American Telemedicine Association, 17(6):484–94, 2011, ISSN 1556-3669. http://www.ncbi.nlm.nih.gov/pubmed/21718114.

[141] Skorin-Kapov, Lea and Maja Matijasevic: *Analysis of QoS requirements for e-Health services and mapping to evolved packet system QoS classes.* International Journal of Telemedicine and Applications, 2010:1–18, January 2010, ISSN 16876415. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2952804&tool=pmcentrez&rendertype=abstract.

[142] ETSI-TR102.732: *Machine-to-Machine Communications (M2M); Use Cases of M2M applications for eHealth.* Technical report, 2013.

[143] Aragüés, Antonio, Javier Escayola, Ignacio Martínez, Pilar Del Valle, Pilar Muñoz, Jesús D Trigo, and José García: *Trends and challenges of the emerging technologies toward interoperability and standardization in e-health communications.* IEEE Communications Magazine, 49(11):182–188, 2011, ISSN 01636804.

[144] Barua, Mrinmoy, M S Alam, Xiaohui Liang, and Xuemin Shen: *Secure and quality of service assurance scheduling scheme for WBAN with application to eHealth.* In *2011 IEEE Wireless Communications and Networking Conference, WCNC 2011*, pages 1102–1106, 2011, ISBN 9781612842547.

[145] Fan, L, W Buchanan, C Thummler, and D Bell: *DACAR Platform for eHealth Services Cloud.* In *2011 IEEE International Conference on Cloud Computing (CLOUD)*, pages 219–226, Washington, DC, 2011. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6008713&tag=1.

[146] Kailas, a, Chia Chin Chong, and F Watanabe: *From mobile phones to personal wellness dashboards.* IEEE pulse, 1(1):57–63, 2010, ISSN 2154-2287. http://www.ncbi.nlm.nih.gov/pubmed/20875965.

[147] IEEE: *IEEE-SA eHealth Standards & Projects*. http://standards.ieee.org/innovate/ehealth/stds.html, visited on 2015-04-14.

[148] The European Commission: *Europe 2020: EU-wide headline targets for economic growth*. http://ec.europa.eu/europe2020/targets/eu-targets/index_en.htm, visited on 2015-08-05.

[149] Gungor, V Cagri, Dilan Sahin, Taskin Kocak, Salih Ergut, Concettina Buccella, Carlo Cecati, and Gerhard P Hancke: *A Survey on Smart Grid Potential Applications and Communication Requirements*. IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, 9(1):28–42, 2013.

[150] Gharavi, Hamid and Reza Ghafurian: *Smart Grid : The Electric Energy System of the Future*. Proceedings of the IEEE, 99(6):917–921, 2011.

[151] NIST: *NIST framework and roadmap for Smart Grid interoperability standards, Release 3*. Technical report, 2014. http://www.nist.gov/smartgrid/upload/NIST-SP-1108r3.pdf.

[152] ETSI TS 102 935 v2.1.1: *Applicability of M2M architecture to Smart Grid Networks ; Impact of Smart Grids on M2M platform*. Technical report, 2012.

[153] Rohjans, Sebastian, Mathias Uslar, Robert Bleiker, Jose Gonzalez, Michael Specht, Thomas Suding, and Tobias Weidelt: *Survey of Smart Grid Standardization Studies and Recommendations*. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 583–588. Ieee, October 2010, ISBN 978-1-4244-6510-1. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5621999http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5638886.

[154] Fan, Zhong, Parag Kulkarni, Sedat Gormus, Costas Efthymiou, Georgios Kalogridis, Mahesh Sooriyabandara, Ziming Zhu, Sangarapillai Lambotharan, and Woon Hau Chin: *Smart Grid Communications: Overview of Research Challenges, Solutions, and Standardization Activities*. IEEE COMMUNICATIONS SURVEYS & TUTORIALS, 15(1):21–38, 2013.

[155] Ritzer, G., P. Dean, and N. Jurgenson: *The Coming of Age of the Prosumer*. American Behavioral Scientist, 56(4):379–398, March 2012, ISSN 0002-7642. http://abs.sagepub.com/cgi/doi/10.1177/0002764211429368.

[156] Khan, Reduan H. and Jamil Y. Khan: *A comprehensive review of the application characteristics and traffic requirements of a smart grid communications network*. Computer Networks, 57(3):825–845, February 2013, ISSN 13891286. http://www.sciencedirect.com/science/article/pii/S1389128612003751.

[157] Kuzlu, Murat and Manisa Pipattanasomporn: *Assessment of Communication Technologies and Network Requirements for Different Smart Grid*

Applications. In *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES*, pages 1–6, Washington, DC, February 2013. ISBN 9781467348966. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6497873&isnumber=6497783.

[158] E&E Publishing: *ENERGY POLICY: Survey shows huge challenges for 'smart grid,' efficiency efforts*. http://www.eenews.net/stories/75281, visited on 2015-08-12.

[159] Yan, Ye, Yi Qian, Hamid Sharif, and David Tipper: *A Survey on Smart Grid Communication Infrastructures : Motivations , Requirements and Challenges*. IEEE COMMUNICATIONS SURVEYS & TUTORIALS, 15(1):5–20, 2013.

[160] Lien, Shao yu and Kwang cheng Chen: *Toward Ubiquitous Massive Accesses in 3GPP Machine-to-Machine Communications*. (April):66–74, 2011.

[161] Hauser, Carl H., David E. Bakken, Ioanna Dionysiou, K. Harald Gjermundrod, Venkata Irava, Joel Helkey, and Anjan Bose: *Security, trust, and QoS in next-generation control and communication for large power systems*. International Journal of Critical Infrastructures, January 2008. http://www.inderscienceonline.com/doi/abs/10.1504/IJCIS.2008.016088.

[162] Fettweis, Gerhard and Ernesto Zimmermann: *ICT Energy Consumption-Trends and Challenges*. In *Proc. 11th Int. Symp. Wirel. Pers. Multimed. Commun.*, 2008.

[163] Siemens: *Energy Efficiency*. http://w3.siemens.com/mcms/topics/en/electrical-wholesale/energy-efficiency/pages/default.aspx.

[164] Reinisch, Christian, Wolfgang Kastner, Georg Neugschwandtner, and Wolfgang Granzer: *Wireless Technologies in Home and Building Automation*. In *2007 5th IEEE International Conference on Industrial Informatics*, volume 1, pages 93–98. IEEE, July 2007, ISBN 978-1-4244-0850-4. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4384737.

[165] Evangelatos, Orestis, Kasun Samarasinghe, and Jose Rolim: *Evaluating design approaches for smart building systems*. 2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012), pages 1–7, October 2012. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6708524.

[166] Tariq, M., Z. Zhou, J. Wu, M. Macuha, and T. Sato: *Smart grid standards for home and building automation*. In *2012 IEEE International Conference on Power System Technology (POWERCON)*, pages 1–6. IEEE, October 2012, ISBN 978-1-4673-2868-5. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6401448.

[167] Vähä, Pentti, Tapio Heikkilä, Pekka Kilpeläinen, Markku Järviluoma, and Ernesto Gambao: *Extending Automation of Building Construction - Survey on potential sensor technologies and robotic applications.* Automation in Construction, 36:168–178, dec 2013, ISSN 09265805. http://www.sciencedirect.com/science/article/pii/S0926580513001167.

[168] ITU-T L.1400: *L.1400:Overview and general principles of methodologies for assessing the environmental impact of information and communication technologies.* Technical report, 2011. https://www.itu.int/rec/T-REC-L.1400.

[169] Cohen, Barney: *Urbanization in developing countries: Current trends, future projections, and key challenges for sustainability.* Technol. Soc., 28(1-2):63–80, January 2006, ISSN 0160791X. http://linkinghub.elsevier.com/retrieve/pii/S0160791X05000588.

[170] Elmangoush, Asma, Hakan Coskun, Thomas Magedanz, and Niklas Blum: *An approach to expose M2M services over OMA next generation service interface.* In *2013 17th International Conference on Intelligence in Next Generation Networks (ICIN)*, pages 147–154. IEEE, Oct 2013, ISBN 978-1-4799-0980-3. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6670906.

[171] Liu, Haiyang, Yuan an Liu, and Jinchun Gao: *Research on Pub / Sub with Predictable Mechanism Based on Caching Architecture of Cluster.* Journal of Information & Computational Science, 10(2):599–608, 2013, ISSN 15487741.

[172] Tolk, Andreas, Saikou Y Diallo, and Charles D Turnitsa: *Applying the Levels of Conceptual Interoperability Model in Support of Integratability , Interoperability , and Composability for System-of-Systems Engineering.* Journal of Systemics, Cybernetics and Informatics, 5(5):65–74, 2007. http://ldsp01.columbusstate.edu:8080/xmlui/handle/11075/412.

[173] *UNIversities for Future Internet (Unifi) Project.* http://daad-unifi.org/unifi-project/.

[174] ETSI TS 102 921 v1.1.1: *Machine-to-Machine communications (M2M); mIa, dIa and mId interfaces.* Technical report, 2012.

[175] OneM2M-TS-0004: *Service Layer Core Protocol Specification*, 2015. http://onem2m.org/technical/published-documents.

[176] Corici, Andreea Ancuta, Ranjan Shrestha, Giuseppe Carella, Asma Elmangoush, Ronald Steinke, and Thomas Magedanz: *A solution for provisioning reliable M2M infrastructures using SDN and device management.* In *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, pages 81–86. IEEE, May 2015, ISBN 978-1-4799-7752-9. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7231401.

[177] Bourreau, Alex: *Design and Implementation of self-adaptable multi transport protocol solution for M2M systems.* Master's thesis, Technische Universitat Berlin, 2014.

[178] GEvent: *Gevent, Python network library.* http://www.gevent.org.

[179] Fraunhofer FOKUS: *FUSECO Playground.* https://www.fokus.fraunhofer.de/eeca1c58a14c2be0/welcome-to-the-fuseco-playground, visited on 2015-09-21.

[180] *Apache JMeter.* http://jmeter.apache.org/, visited on 2015-10-12.

[181] *RabbitMQ - Installing on Debian / Ubuntu.* http://www.rabbitmq.com/install-debian.html, visited on 2015-11-20.

[182] FP7-TRESCIMO: *TRESCIMO (Testbeds for Reliable Smart City Machine to Machine Communication) Project.* www.trescimo.eu.

[183] Coetzee, Louis: *TRESCIMO Scenario Specification.* Technical report, apr 2014. http://trescimo.eu/wp-content/uploads/2015/01/TRESCIMO{_}D2.1{_}v1.0.pdf.

[184] *OpenVPN - Open Source VPN.* https://openvpn.net/, visited on 2015-7-02.

[185] Ventura, Neco and Joyce Mwangama: *Evaluation Environment.* Technical report, 2015. http://trescimo.eu/wp-content/uploads/2015/11/D4.2-Evaluation-Environment-v1{_}final.pdf.

[186] Catalan, Marisa: *TRESCIMO Experiment Results.* Technical report, 2015. http://trescimo.eu/wp-content/uploads/2015/11/D4.3-TRESCIMO{_}final.pdf.

[187] *Fed4Fire.* http://www.fed4fire.eu/, visited on 2015-11-16.

[188] *FITeagle- Future Internet Testbed Experimentation and Management Framework.* http://fiteagle.org/.

[189] Corici, Andreea Ancuta: *Architecture Specification.* Technical report, 2015. http://trescimo.eu/wp-content/uploads/2015/01/D3.1{_}final.pdf.

[190] Elmangoush, Asma, A. Corici, Adel Al-Hezmi, and Thomas Magedanz: *Mobility Management for Machine-to-Machine (M2M) Communications.* In Anton-Haro, Carles and Mischa Dohler (editors): *Machine-to-machine (M2M) Communications*, pages 187–206. Elsevier, 2015, ISBN 978-1-78242-102-3. http://www.elsevier.com/books/machine-to-machine-m2m-communications/anton-haro/978-1-78242-102-3http://www.sciencedirect.com/science/article/pii/B9781782421023000113http://linkinghub.elsevier.com/retrieve/pii/B9781782421023000113.

[191] Abdurohman, Maman, Anton Herutomo, Vera Suryani, Asma Elmangoush, and Thomas Magedanz: *Mobile tracking system using OpenMTC platform based on event driven method.* In *38th Annual IEEE Conference on Local Computer Networks - Workshops*, pages 856–860. IEEE, Oct 2013, ISBN 978-1-4799-0540-9. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6758523.

[192] Elmangoush, Asma, Adel Alhazmi, Thomas Magedanz, Wolfgang Schuch, Claudio Estevez, Alfonso Ehijo, and Jinsong Wu: *Towards Unified Smart City Communication Platforms.* In *RedSTI 2015 - Workshop on Research in Information Systems and Technologies*, Chillán, 2015. http://conference.researchbib.com/view/event/48789.

[193] Ratasuk, Rapeepat, Athul Prasad, Zexian Li, Amitava Ghosh, and Mikko Uusitalo: *Recent advancements in M2M communications in 4G networks and evolution towards 5G.* In *2015 18th International Conference on Intelligence in Next Generation Networks*, pages 52–57. IEEE, 2015, ISBN 978-1-4799-1866-9. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7073806.

[194] Tehrani, Mohsen Nader, Murat Uysal, and Halim Yanikomeroglu: *Device-to-device communication in 5G cellular networks: challenges, solutions, and future directions.* IEEE Communications Magazine, 52(5):86–92, may 2014, ISSN 0163-6804. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6815897.

[195] W3C Semantic Sensor Network Incubator Group: *Semantic Sensor Network Ontology.* https://www.w3.org/2005/Incubator/ssn/ssnx/ssn, visited on 2014-01-26.

[196] Hachem, Sara, Thiago Teixeira, and Valérie Issarny: *Ontologies for the internet of things.* In *Proceedings of the 8th Middleware Doctoral Symposium on - MDS '11*, pages 1–6, New York, New York, USA, 2011. ACM Press, ISBN 9781450310727. http://dl.acm.org/citation.cfm?doid=2093190.2093193.

[197] Alaya, Mahdi Ben, Samir Medjiah, Thierry Monteil, and Khalil Drira: *Toward semantic interoperability in oneM2M architecture.* IEEE Communications Magazine, 53(12):35–41, dec 2015, ISSN 0163-6804. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7355582.

[198] Grau, Alan: *Can you trust your fridge?* IEEE Spectrum, 52(3):50–56, mar 2015, ISSN 0018-9235. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7049440.

[199] Wu, Geng, Shilpa Talwar, Kerstin Johnsson, Nageen Himayat, and Kevin D. Johnson: *M2M: From mobile to embedded internet.* IEEE Communica-

tions Magazine, 49(4):36–43, 2011, ISSN 01636804. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5741144&tag=1.

[200] Ye, Sigen, Shin Horng Wong, and Chandrika Worrall: *Enhanced Physical Downlink Control Channel in LTE Advanced Release 11.* Communications Magazine, IEEE, 51(February):82–89, 2013.

[201] Trakoma, Sasu and Artem Katasonov: *Internet of Things Strategic Research Agenda.* Technical report, 2011. www.internetofthings.fi.

[202] RFID WORKING GROUP-EPoSS: *Internet of things in 2020: Roadmap for the future.* Technical report, 2008. http://www.smart-systems-integration.org/public/documents/publications/Internet-of-Things_in_2020_EC-EPoSS_Workshop_Report_2008_v3.pdf.

[203] Wan, Jiafu, Min Chen, Feng Xia, Li Di, and Keliang Zhou: *From machine-to-machine communications towards cyber-physical systems.* Computer Science and Information Systems, 10(3):1105–1128, 2013, ISSN 1820-0214. http://www.doiserbia.nb.rs/Article.aspx?ID=1820-02141300018W.

[204] Bartoli, Andrea, Mischa Dohler, Juan Hernández-Serrano, Apostolos Kountouris, and Dominique Barthel: *Low-power low-rate goes long-range: The case for secure and cooperative machine-to-machine communications.* In Casares-Giner, Vicente, Pietro Manzoni, and Ana Pont (editors): *NETWORKING 2011 Workshops*, volume 6827 of *Lecture Notes in Computer Science*, pages 219–230. Springer Berlin Heidelberg, 2011, ISBN 978-3-642-23040-0. http://dx.doi.org/10.1007/978-3-642-23041-7_21.

[205] Ashton, Kevin: *That 'Internet of Things' Thing* -, 2009. http://www.rfidjournal.com/articles/view?4986, visited on 02/07/14.

[206] Cisco: *The Internet of Everything.* http://share.cisco.com/IoESocialWhitepaper/#/, visited on 2015-04-16.

[207] Vera, David Díaz Pardo de, Alvaro Sigüenza Izquierdo, Jesús Bernat Vercher, and Luis Alfonso Hernández Gómez: *A ubiquitous sensor network platform for integrating smart devices into the semantic sensor web.* Sensors (Basel, Switzerland), 14(6):10725–52, January 2014, ISSN 1424-8220. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4118357&tool=pmcentrez&rendertype=abstract.

[208] Sanchez, L., J.A. Galache, V. Gutierrez, J.M. Hernandez, J. Bernat, A. Gluhak, and T. Garcia: *SmartSantander: The meeting point between Future Internet research and experimentation and the smart cities*, 2011.

[209] Li, Yun, Kok Keong Chai, Yue Chen, and Jonathan Loo: *Duty cycle control with joint optimisation of delay and energy efficiency for capillary machine-to-machine networks in 5G communication system.* Transactions on Emerging

Telecommunications Technologies, pages n/a–n/a, nov 2014, ISSN 21613915. http://doi.wiley.com/10.1002/ett.2891.

[210] Lee, Gyu Myoung, Jungsoo Park, Ning Kong, and Noel Crespi: *The Internet of Things - Concept and Problem Statement*. Technical report, Internet Draft, Internet Research Task Force, 2012. https://tools.ietf.org/html/draft-lee-iot-problem-statement-05.

[211] Gartner: *Gartner's 2014 Hype Cycle for Emerging Technologies Maps the Journey to Digital Business*. http://www.gartner.com/newsroom/id/2819918, visited on 2015-01-22.

[212] FP7-ICT-2007.1.1-215923: *Project SENSEI*. www.sensei-project.eu.

[213] FP7-ICT-2009-5-257992: *Project SmartSantander*. http://www.smartsantander.eu/, visited on 2015-02-27.

[214] FP7-ICT-2007.1.6-224460: *Project WISEBED*. http://www.wisebed.eu/.

[215] *FInish: Future Internet Accelerator for Food, Perishables and Logistics*. http://www.finish-project.eu/, visited on 2015-09-02.

List of Acronyms

6LoWPAN	IPv6 for Low-power Wireless Personal Area Network
3GPP	3rd Generation Partnership Project
AdM2M	Adaptable M2M
ADN	Application Detected Node
AE	Application Entity
xAE	(Device/Gateway/ Network) Application Enablement
AMI	Advanced Metering Infrastructure
AMR	Automatic Meter Reading
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
ARM	Architecture Reference Model
AS	Application Server
ASM	Application and Service Layer Management
ASN	Application Service Node
BEMS	Building Energy Management System
BIM	Building Information Modelling
BLE	Bluetooth Low Energy
xCB	(Device/Gateway/ Network) Compensation Broker
CMDH	Communication Management and Delivery Handling
CoAP	Constrained Application Protocol
CoRE	Constrained RESTful Environments
xCS	(Device/Gateway/ Network) Communication Selection

CSE	Common Services Entity
CSF	Common Service Function
D2D	Device-to-Device
DM	Device Management
DMR	Data Management and Repository
DTLS	Datagram Transport Layer Security
DTN	Delay Tolerant Network
ECG	Electrocardiogram
EHR	Electronic Health Record
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
FG SSC	Focus Group on Smart Sustainable Cities
FI	Future Internet
FISTAR	Future Internet Social and Technological Alignment Research in Healthcare
FOKUS	Fraunhofer Institute for Open Communication Systems
FQDN	Fully Qualified Domain Name
FUSECO	FUture SEamless COmmunication
xGC	(Device/Gateway/ Network) Generic Communication
GPS	Global Positioning System
GUI	Graphical User Interface
H2H	Human-to-Human
H2M	Human-to-Machine
HATEOAS	Hypermedia as the Engine of Application State
xHDR	(Device/Gateway/ Network) History and Data Retention
HMI	Human Machine Interface
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IEEE-SA	IEEE-Standards Association
IEC	International Electrotechnical Commission
IERC	European Research Cluster on the Internet of Things

IETF	Internet Engineering Task Force
IMS	IP-Multimedia Subsystem
IN	Infrastructure Node
IoC	Internet of Content
IoT	Internet of Things
IP	Internet Protocol
xIP	(Device/Gateway/ Network) Interworking Proxy
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-T	ITU - Telecommunication Standardization Sector
JSON	JavaScript Object Notation
LAN	Local Area Network
LWM2M	LightweightM2M
LR-WPAN	Low-Rate Wireless Personal Area Network
LTE	Long Term Evolution
LTE-A	LTE-advanced
M2M	Machine-to-Machine
MAC	Medium Access Control
MME	Mobility Management Entity
MMC	Massive Machine Communications
MN	Middle Node
MQTT	Message Queue Telemetry Transport
MSC	Master of Science
MTC	Machine-Type Communications
MTC-IWF	Machine-Type Communications (MTC)-InterWorking Function
MTU	Maximum Transmission Unit
NGN	Next Generation Network
NGSI	Next Generation Service Interfaces
NIST	National Institute of Standards and Technology
NSE	Network Service Entity
NSSE	Network Service Exposure, Service Execution and Triggering
NTOE	Network Telco Operator Exposure
NTP	Network Time Protocol

OMA	Open Mobile Alliance
OpenMTC	Fraunhofer FOKUS Machine-Type Communication
PAN	Personal Area Network
PHD	Personal Health Device
PMU	Phasor Measurement Unit
Pub/Sub	Publish/Subscribe
QoS	Quality of Service
RAT	Radio Access Technology
xRAR	(Device/Gateway/ Network) Reachability, Addressing and Repository
xREM	(Device/Gateway/ Network) Remote Entity Management
Req/Res	Request/Response
RAT	Radio Access Technology
REST	Representational State Transfer
SCADA	Supervisory Control and Data Acquisition
SAF	Store And Forward
SCL	Service Capability Layer
SDO	Standards Developing Organizations
SDK	Software Development Kit
xSEC	(Device/Gateway/ Network) Security
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
SSN	Semantic Sensor Network
TCP	Transmission Control Protocol
TIA	Telecommunications Industry Association
TRESCIMO	Testbeds for Reliable Smart City Machine to Machine Communication
TLS	Transport Layer Security
TLV	Type-Length-Value
xTM	(Device/Gateway/ Network) Transaction Management
UDP	User Datagram Protocol
UE	User Equipment/Endpoint

UNIFI	Universities for Future Internet
URI	Uniform Resource Identifier
VPN	Virtual Private Network
WAN	Wide Area Network
WBAN	Wireless Body Area Network
WSN	Wireless Sensors Network
XaaS	Everything-as-a-Service
XML	Extensible Markup Language
XSD	XML Schema Definition

Author's Publications

[1] Elmangoush, Asma, Hakan Coskun, Sebastian Wahle, Niklas Blum, and Thomas Magedanz: *Promoting M2M Application Development for Smart City.* In *Wireless World Research Forum Meeting 29 (WWRF)*, Berlin, 2012.

[2] Elmangoush, Asma, Thomas Magedanz, Alexander Blotny, and Niklas Blum: *Design of RESTful APIs for M2M Services.* In *16th International Conference on Intelligence in Next Generation Networks*, pages 50–56, Berlin, 2012. ISBN 9781467315265.

[3] Corici, Marius, Hakan Coskun, Asma Elmangoush, Agus Kurniawan, Tong Mao, Thomas Magedanz, and Sebastian Wahle: *OpenMTC : Prototyping Machine Type Communication in Carrier Grade Operator Networks.* In *4th International IEEE Workshop on Open NGN and IMS Testbeds (ONIT 2012) @ GLOBECOM 2012*, pages 1735 – 1740, Anaheim, CA, 2012. IEEE, ISBN 978-1-4673-4942-0. http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6477847.

[4] Coskun, Hakan, Tom Pfeifer, Asma Elmangosh, and Adel Al-Hezmi: *Open M2M Data - Position paper.* In *38th Annual IEEE Conference on Local Computer Networks - Workshops*, pages 904–911. IEEE, oct 2013, ISBN 978-1-4799-0540-9. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6758530.

[5] Suryani, Vera, Achmad Rizal, Anton Herutomo, Maman Abdurohman, Thomas Magedanz, and Asma Elmangoush: *Electrocardiagram monitoring on OpenMTC platform.* In *38th Annual IEEE Conference on Local Computer Networks - Workshops*, number 2, pages 843–847, Sydney, NSW, oct 2013. IEEE, ISBN 978-1-4799-0540-9. http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6758521{&}tag=1http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6758521.

[6] Abdurohman, Maman, Anton Herutomo, Vera Suryani, Asma Elmangoush, and Thomas Magedanz: *Mobile tracking system using OpenMTC platform based on event driven method.* In *38th Annual IEEE Conference on Local Computer Networks - Workshops*, pages 856–860. IEEE, oct 2013, ISBN 978-1-4799-0540-9. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6758523.

[7] Elmangoush, Asma, Adel Al-hezmi, and Thomas Magedanz: *Towards Standard M2M APIs for Cloud-based Telco Service Platforms.* In *Proceedings of International Conference on Advances in Mobile Computing & Multimedia - MoMM '13*, pages 143–149. ACM Press, 2013, ISBN 9781450321068. http://dl.acm.org/citation.cfm?doid=2536853.2536892.

[8] Elmangoush, Asma, Hakan Coskun, Thomas Magedanz, and Niklas Blum: *An Approach to Expose M2M Services over OMA Next Generation Service Interface.* In *2013 17th International Conference on Intelligence in Next Generation Networks (ICIN)*, pages 147–154. IEEE, Oct 2013, ISBN 978-1-4799-0980-3. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6670906.

[9] Elmangoush, Asma, Hakan Coskun, Sebastian Wahle, and Thomas Magedanz: *Design Aspects for a Reference M2M Communication Platform for Smart Cities.* In *2013 9th International Conference on Innovations in Information Technology (IIT)*, pages 204–209, Abu Dhabi, Mar 2013. IEEE, ISBN 978-1-4673-6203-0. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6544419.

[10] Mwangama, Joyce, Alexander Willner, Neco Ventura, Asma Elmangoush, Tom Pfeifer, and Thomas Magedanz: *Testbeds for Reliable Smart City Machine-to-Machine Communication.* In *Southern African Telecommunication Networks and Applications Conference (SATNAC)*, pages 339–344, South Africa, 2013. ISBN 978-0-620-57883-7. http://www.satnac.org.za/proceedings/2013/SATNAC{_}2013{_}Conference{_}Proceedings.pdf.

[11] Corici, Andreea Ancuta, Asma Elmangoush, Thomas Magedanz, Ronald Steinke, Joyce Mwangama, and Neco Ventura: *An OpenMTC Platform-Based Interconnected European – South African M2M Testbed for Smart City Services.* In *the first International Conference on the use of Mobile Informations and Communication Technology (ICT) in Africa - UMICTA 2014*, pages 35–39, Stellenbosch, 2014. ISBN 978-0-7972-1533-7.

[12] Corici, Andreea Ancuta, Asma Elmangoush, Ronald Steinke, Thomas Magedanz, Joyce Mwangama, and Neco Ventura: *Utilizing M2M Technologies for Building Reliable Smart Cities.* In *Smart City Workshop @ NTMS2014*, pages 1–5. IEEE, mar 2014, ISBN 978-1-4799-3223-8. http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6814059.

[13] Elmangoush, Asma, Adel Al-Hezmi, and Thomas Magedanz: *The Development of M2M Standards for Ubiquitous Sensing Service Layer.* In *2014 IEEE Globecom Workshops (GC Wkshps)*, pages 624–629. IEEE, dec 2014, ISBN 9781479974702. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7063502.

[14] Elmangoush, Asma, Andreea Ancuta Corici, Marisa Catalan, Ronald Steinke, Thomas Magedanz, and Joaquim Oller: *Interconnecting Standard M2M Platforms to Delay Tolerant Networks.* In *2014 International Conference on Future Internet of Things and Cloud*, pages 258–263, Barcelona, 2014. IEEE, ISBN 978-1-4799-4358-6. http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6984204.

[15] Elmangoush, Asma, Ronald Steinke, Adel Al-hezmi, and Thomas Magedanz: *On The Usage of Standardised M2M Platforms for Smart Energy Management.* In *28th International Conference on Information Networking (ICOIN)*, pages 79–84, Phuket, feb 2014. IEEE, ISBN 9781479936892. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6799669.

[16] Klinpratum, Teerapan, Chaiyachet Saivichit, Asma Elmangoush, and Thomas Magedanz: *Toward Interconnecting M2M / IoT Standards: Interworking Proxy for IEEE1888 Standard at ETSI M2M Platforms.* In *29th International Technical Conference on Circuit/Systems Computers and Communications (ITC-CSCC 2014)*, pages 763–766, Phuket, Thailand, 2014.

[17] Kosolworrawattanakul, Napat, Asma Elmangoush, Thomas Magedanz, and Chaodit Aswakul: *Development of Real-Time Data Synchronization for IEEE1888 and ETSI M2M Standards.* In *IEICE Technical Report*, pages 79–84, 2014. http://www.ieice.org/{~}ia/IA2014/wiki.cgi.

[18] Mwangama, Joyce, Asma Elmangoush, Joseph Orimolade, Neco Ventura, Ronald Steinke, Alexander Willner, Andreea Ancuta Corici, and Thomas Magedanz: *Prototyping Machine-to-Machine Applications for Emerging Smart Cities in Developing Countries.* In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2014*, pages 383–388, aug 2014, ISBN 978-0-620-61965-3. http://www.satnac.org.za/proceedings/2014/SATNAC2014ConferenceProceedings{_}USB{_}edition.pdf.

[19] Elmangoush, Asma, Adel Al-Hezmi, and Konrad Campowsky: *Automating Building Management Based on M2M Communication Platform.* Internet of Things-Success Stories, pages 40–44, 2015. https://www.smart-action.eu/fileadmin/smart-action/publications/IoT{_}Success{_}Stories{_}2.pdf.

[20] Corici, Andreea Ancuta, Ranjan Shrestha, Giuseppe Carella, Asma Elmangoush, Ronald Steinke, and Thomas Magedanz: *A Solution for Provisioning*

Reliable M2M Infrastructures Using SDN and Device Management. In *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, pages 81–86. IEEE, May 2015, ISBN 978-1-4799-7752-9. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7231401.

[21] Elmangoush, Asma, Adel Alhazmi, Thomas Magedanz, Wolfgang Schuch, Claudio Estevez, Alfonso Ehijo, and Jinsong Wu: *Towards Unified Smart City Communication Platforms.* In *RedSTI 2015 - Workshop on Research in Information Systems and Technologies*, Chillán, 2015. http://conference.researchbib.com/view/event/48789.

[22] Elmangoush, Asma, A. Corici, Adel Al-Hezmi, and Thomas Magedanz: *Mobility Management for Machine-to-Machine (M2M) Communications.* In Anton-Haro, Carles and Mischa Dohler (editors): *Machine-to-machine (M2M) Communications*, pages 187–206. Elsevier, 2015, ISBN 978-1-78242-102-3. http://www.sciencedirect.com/science/article/pii/B9781782421023000113.

[23] Elmangoush, Asma, Andreea Ancuta Corici, Ronald Steinke, Marius Corici, and Thomas Magedanz: *A Framework for Handling Heterogeneous M2M Traffic.* Procedia Computer Science, 63:112–119, 2015. http://www.sciencedirect.com/science/article/pii/S1877050915024473.

[24] Elmangoush, Asma, Ronald Steinke, Thomas Magedanz, Andreea Ancuta Corici, Alex Bourreau, and Adel Al-Hezmi: *Application-derived Communication Protocol Selection in M2M Platforms for Smart Cities.* In *2015 18th International Conference on Intelligence in Next Generation Networks*, pages 76–82, Paris, France, Feb 2015. IEEE, ISBN 978-1-4799-1866-9. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7073810.

[25] Klinpratum, Teerapan, Chaiyachet Saivichit, Asma Elmangoush, and Thomas Magedanz: *Performance of Interworking Proxy for Interconnecting IEEE1888 Standard at ETSI M2M Platforms.* Applied Mechanics and Materials, 781:141–144, 2015.

[26] Elmangoush, Asma, Ronald Steinke, and Thomas Magedanz: *AdM2M : Adaptable Machine-to-Machine Transport Framework.* In *The 17th International Conference on Information Integration and Web-based Applications & Services (iiWAS2015)*, pages 331–335, Brussels, 2015. ISBN 9781450334914.

TECHNISCHE UNIVERSITÄT BERLIN
School IV - Electrical Engineering and Computer Science
Department of Telecommunication Systems
Next Generation Networks (AV)

- Engineering Doctorate Dissertation -

Author: Asma Abdalla Elmangoush – Berlin 2016

asma.a.elmangoush@campus.tu-berlin.de