The Development of M2M Standards for Ubiquitous Sensing Service Layer

Asma Elmangoush, *Student Member, IEEE*, Adel Al-Hezmi, and Thomas Magedanz, *Member, IEEE*

***Abstract*— Currently, a lot of research efforts are ongoing in the Machine-to-Machine (M2M) communication area, with the object of creating an ubiquitous sensing framework to connect real and virtual things. In the same context, various standards developing organizations (SDO) have recently promoted standardization activities in the M2M domain. In this paper, we investigate the standardization efforts toward a common M2M service layer that provides end-to-end service delivery and integrates heterogeneous devices and technologies. In addition, we present an ETSI-compliant implementation of M2M service layer that include advanced features for M2M solutions and able to extend to oneM2M standards to be released end of this year.**

***Index Terms*— Machine-to-Machine; ETSI; oneM2M; OMA**

I. Introduction

Machine-To-Machine (M2M) technologies aim to enable variety of objects and devices to communicate, collect and share data through wired and wireless networks. The term is broadly used with estimations of high future market potentials. M2M services involves a set of different technology components required to work together to provide a particular M2M solution. For Smart Cities, the need for standardization is highly recognized to remove the technical barriers and support interoperability between connected systems and services.

The standardization process refers to the developing of technical specifications, which aims to maximize compatibility, interoperability, safety, repeatability, or quality. The process shall involve industry, consumers, public authorities and other interested parties based on consensus. 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 M2M 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.

Several standardization bodies have established technical committees or working groups responsible for specific topics related to M2M and future Internet. Just to mention a few: The 3rd Generation Partnership Project (3GPP), refers to M2M communication as Machine Type Communication (MTC), started standardization activities on MTC in September 2008 as part of 3GPP Rel. 10 specifications. The service requirements working group (3GPP SA WG1) has specified a number of use cases and scenarios and derived a set of service requirements accordingly. In 3GPP Rel- 11, some of the proposed MTC features were finalized, such as addressing and device triggering. Importantly is the enhancement of the 3GPP architecture to support MTC applications [1], by introducing a Machine Type Communications-InterWorking Function (MTC-IWF) to interact with external MTC Servers. The Internet Engineering Task Force (IETF) has established several Working Groups [2], such as the 6LoWPAN, ROLL and CoRE aiming to enable contained devices to integrate into the Internet of things (IoT), by making the IPv6 protocol compatible with constrained devices and unreliable networks.

Development standards shall cover various areas such as data modules and access networks. However, in this paper we focus on standards addressing general M2M service layer. The main contribution of this paper is analysing the standard efforts to specify a general M2M service layer supporting interoperability and scalability.

The rest of this paper is organized as following: Section II previews the main capabilities required to build a common M2M architectural platforms. Section III reviews the standardization efforts toward a common M2M framework. Section IV tries to answer the question of how standard bodies address the requirements towards realizing all required capabilities. Section V introduces the openMTC platform; which provide a M2M service layer prototype compliant with ETSI M2M standard and currently upgraded to support oneM2M standard. Finally, the paper is concluded in Section VI.

II. Key Building Capabilities for M2M Platforms

Based on analyzing the main requirements of Smart City and M2M services, five basic capabilities can be noticed in all

Asma Elmangoush is with the Technical University of Berlin, Marchstraße 23, 10587 Berlin - Germany (email: asma.a.elmangoush@campus.tu-berlin.de)

Adel AlHezmi is with Fraunhofer Institute FOKUS - competence center Next Generation Network Infrastructures (NGNI), Kaiserin-Augusta-Allee 31 10589 Berlin Germany (email: adel.al-hezmi@fokus.fraunhofer.de)

Thomas Magedanz is with Fraunhofer Institute FOKUS - competence center Next Generation Network Infrastructures (NGNI), Kaiserin-Augusta-Allee 31, 10589 Berlin Germany (email: thomas.magedanz@ fokus.fraunhofer.de)

platforms:

A. Connectivity (Communication 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 traffic according to traffic pattern and profiles.

B. 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. The solution shall enable discover, control and mange the sensors and actuators with support various abstraction level, e.g. supporting virtualization of these devices.

C. 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. M2M platforms shall help abstract application development and ongoing management. 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.

D. Data Processing and Semantic

Overall mobile data traffic is expected to grow to 15.9 Exabytes per month by 2018 [3], this increase is partly due to the continues increase in mobile-connected devices including M2M devices. That trend in ubiquitous sensing and inexpensive storage capabilities, leads to the 'Big Data' concept [4]. Both M2M and big data technologies are expected to play important roles in the construction of intelligent societies such as the Smart city framework. Furthermore, there are certain applications that require real-time data processing and provision. For instance, the response times of various smart grid applications range between 10 and 1000 ms [5].

E. Security

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. Authors in [6] categorize the main security vulnerabilities in M2M systems.

III. Standards Summary

Standardization is essential to remove the technical barriers and ensure interoperable M2M services and networks. In this section, we preview the M2M reference architectures proposed by standard organizations working in the M2M domain.

A. The European Telecommunications Standards Institute (ETSI)

Aiming at an efficient end-to-end delivery of the M2M services, ETSI created a Technical Committee (TC) to develop M2M standards in 2009. The TC M2M standardization work mainly focuses on the service middleware layer [7]. Starting by defining a set of use case's requirements. These requirements address mainly 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 defines M2M communication as "the communication between two or more entities that do not necessarily need any direct human intervention" [8]. ETSI M2M work defines a middleware Service Capability Layer (SCL) that interact with M2M nodes over open interfaces: mIa, dIa and mId [9]. These interfaces offer generic and extendable mechanism for interactions with the SCLs at both device and gateway domain and network domain. As shown in Fig 1, the ETSI M2M reference architecture consists of three parts:

1) *M2M area network:*
   This consists of heterogeneous endpoint devices, such as sensors and actuators, connected through a sensor network based on various technologies, e.g., ZigBee, M-BUS, or Bluetooth. This part of the network ends with an M2M gateway, which hides the complexity of the area network from the rest of the communicating entities. The gateway provides a set of service capabilities to M2M applications in this domain, including the Generic Communication (GC) capacity, which handle transport and session management functionalities.
2) *M2M Middleware core:*
   The M2M core implements functionality to facilitate the communication between devices (in the M2M area network) and the network application domain. The M2M core provides 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 information received from the device, and transmitted to applications that shows interest of that information 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:*
   The M2M middleware allows the connection of multiple applications addressing very heterogeneous use cases in different industries, such as energy, automotive, health, transportation etc. The main function of any M2M

application is to aggregate data presenting measurements form surrounded environment, perform some calculations on them prior to decision making, and finally send commands to act according to that decision.

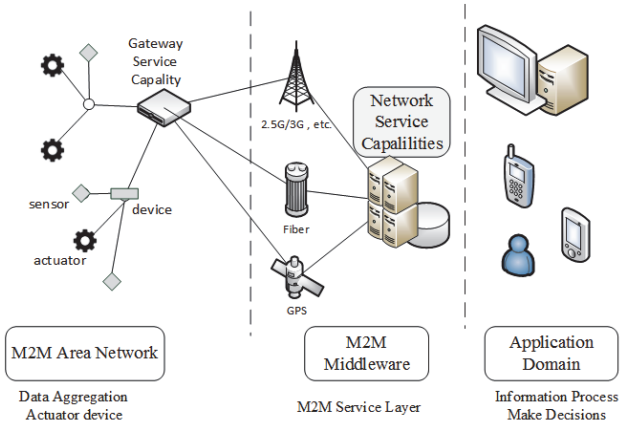

Fig. 1. ETSI M2M Architecture

B. *oneM2M Partnership Project*

In 2012, the oneM2M consortium [10] was established with the aim of consolidating the standardization work in M2M communication in global level. oneM2M is a consortium of seven standards development bodies working in the M2M communication standardization. More than 260 participating partners and members has joint 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. Authors in [11] present an overview of the progress in oneM2M standardization. The consortium has released its candidate specifications in August 2014 for review.

oneM2M object is unifying the global M2M Community, by enabling the federation and interoperability of M2M systems, across multiple networks and topologies. The oneM2M functional architecture comprises of following entities:

1) *Application Entity (AE):*
The AE is responsible of providing end-to-end M2M logic solution, i.e. ehealth, logistic, Smart Energy, etc.

2) *Common Services Entity (CSE):*
The 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 Mca, Mcn, Mcc and Mcc'. 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. A 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):*
The ENSE provides services to the CSEs, such as device management, location services and device triggering.

Each node at oneM2M architecture consists of at least one functional entities. In [12] oneM2M described four type of nodes: the Application Service Node (ASN), the Application Dedicated Node (ADN), the Middle Node (MN), and the Infrastructure Node (IN). Currently oneM2M specifies four reference points:

1. Mca reference point: for interaction communication between an AE and CSEs, that enable the AE to use the expose 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 anther 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.

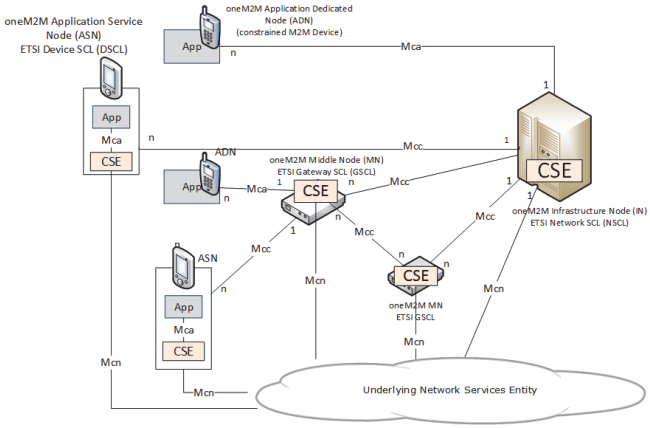

Fig. 2. oneM2M overview architutture

C. *Open Mobile Alliance (OMA)*

OMA has approved and released the final version of OMA NGSI in May 2012 [13], 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 depicts the relation of NGSI interfaces and functional areas. The functional areas for these interfaces are the following:

- Data Configuration and Management: Responsible for creating, reading, updating and deleting data of 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. Also call conferencing control is 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.

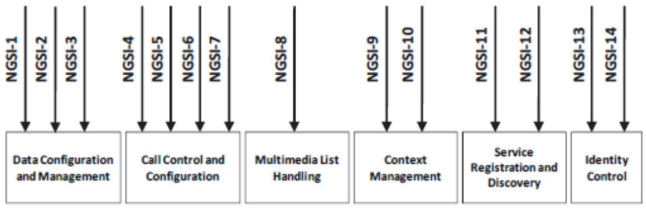

Fig. 3. OMA NGSI Architectural.

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 Open Mobile Alliance (OMA) is providing platform-independent Device Management (OMA DM) protocol for general devices [14]. The DM protocol defines an interface between the DM Server and the DM Client to manage and configure devices on top of Hypertext Transfer Protocol (HTTP) transport protocol. Recently, OMA introduced the Lightweight M2M DM (LW M2M) [15], a device management protocol matching the constraint requirement for M2M domain by using Constrained Application Protocol (CoAP) [16] as transport protocol. The payload can be encoded as plain text, JSON object or TLV (Type-Length-Value) encapsulated, making the encoding very efficient. Security can be achieved by using Datagram Transport Layer Security (DTLS).

IV. ANALYSIS: HOW DO STANDARDS ADDRESS M2M FRAMEWORK REQUIREMENTS

As presented in previous section, 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 a set of distributed Service Capability Layers (xSCL), where the x stands for N (Network), G (Gateway), or D (Device). OneM2M architecture is almost similar to ETSI architecture, however oneM2M defines an open interface between interconnected Common Services Entities (CSE) to represent a mesh topology. In this section, we review the specified capabilities and functionalities by ETSI, oneM2M and OMA, trying to answer the question of how standards address the requirements towards realizing a reliable and secure architectural framework for M2M services. Table I summarize the supporting M2M capabilities by ETSI, oneM2M and OMA.

A. Connectivity Control

Each SCL in the ETSI architecture include the Generic Communication (xGC) Capability, which is responsible for the established transport session including encryption and reporting errors features. Similarly, the Communication Management and Delivery Handling (CMDH) CSF form oneM2M provides communications with other entities i.e. CSEs, AEs and NSEs, while the Service Session Management (SSM) CSF manages M2M service sessions between M2M Applications and CSEs.

For managing access of alternative networks, a communication service selection function is descripted by both ETSI (Communication Selection (xCS) capability) and oneM2M (Network Service Exposure, Service Execution and Triggering (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.

B. Device Management

OMA provides standard mechanisms and protocols for device management in wire and wireless areas, which has been mapped by other standards bodies. ETSI M2M committee has specified three OMA Device Management (DM) compliant Management Objects (MO) [17]. The configuration of MOs is provided by the Remote Entity Management (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 (DMG), and device triggering. The Location capability form oneM2M specifying 3 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.

C. Application Management

The ETSI xSCL 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. In addition, 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 Application Entities (AEs) and CSEs by utilizing DMG functions.

D. Data Processing and Semantic

ETSI WG3 has focused on the hierarchical representation of M2M resources as well as on standard APIs for accessing them by the CRUD (Create, Retrieve, Update and Delete) verbs. The Reachability, Addressing and Repository (xRAR) capability is the cornerstone for ETSI M2M platforms, responsible of data storage and exchange between applications and SCLs. This capability include also the subscription/notification mechanism, which enables applications to receive events notifications from gateways, also supports information searching based on defined criteria.

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 operations of the NGSI Context Management interfaces (NGSI-9 and NGSI-10) fits to exposing M2M service's capabilities to 3rd party developers.

TABLE I. SPECIFIED FUNCTIONS BY M2M STANDARDS

Capability	Functions	ETSI	oneM2M	OMA
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	●	●	○

E. Security

ETSI TC M2M addressed the 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, supporting roles and context attributes
- Support of dynamic configurations, involving a Centralized Key Distribution.

Table 1 shows a comparison between these three standards: ETSI M2M, oneM2M and OMA NGSI against the requirements discussed above. It is obvious that oneM2M specification tries to cover all aspects related to an end-to-end M2M solution. However, oneM2M specification has not been finalized yet.

V. OPENMTC: UPGRADING M2M ETSI IMPLEMENTATION TO ONEM2M

OpenMTC is a cooperative development of Fraunhofer FOKUS and Technical University Berlin (TUB), designed to act as a horizontal convergence M2M layer supporting multiple vertical domains [18]. The first release of OpenMTC features are aligned with ETSI M2M Rel. 1 specifications [7][9], providing an implementation of ETSI specified Service Capability Layers (SCL) at the Frontend (Gateway GSCL) and Backend (Network NSCL) M2M architecture. Currently, the oneM2M specifications are adapted on the design of the OpenMTC release 3. OpenMTC supports a client/server based RESTful architecture with a hierarchical resource tree defined by ETSI, and communication over all interfaces is independent of the transport protocol. The upgrading to oneM2M specification will not require high-level changing on the general architecture; however, minor enhancements might be needed to adapt modifications of the resource tree. As illustrated in Figure 4, the OpenMTC platform includes a Generic Transport (GT) layer that enables the interaction between the frontend and backend over unmanaged access, as well as managed access networks by integrating with the OpenEPC framework. The GT layer includes an Adaptable M2M Transport (AM2MT) module, which provides pluggable transport protocols such as HTTP and CoAP [16]. This allow supporting different type of domain-specific applications with various interaction models, such as push/pull or subscribe/notification model.

The architecture comprises a DM server based on OMA lightweight M2M protocol [15], which enables exchanging DM object information according to the resource models for handling access control, connectivity monitoring, and location management. 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 consists of a set of high-level abstraction Application Programming Interfaces (APIs), which hide internal system complexity, and allow the developer to focus on the implementation of the application logic. In addition, it supports the OMA NGSI 9 and 10 interfaces for context management on the gateway and the backend server [19]. The system supports either XML or JSON format for data representation.

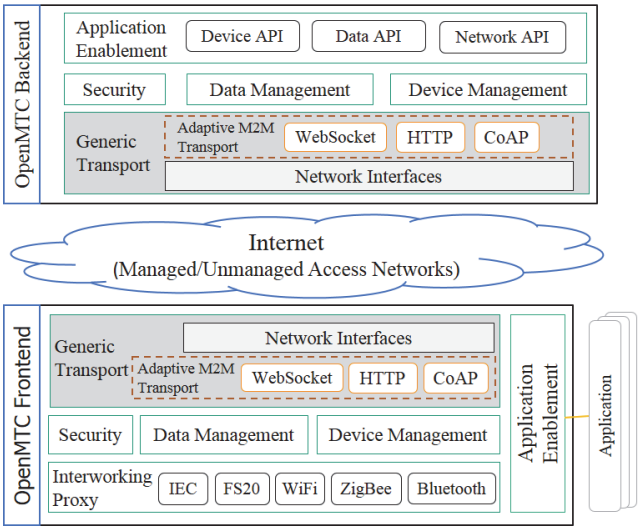

Fig. 4. openMTC architecture

VI. Conclusion

Standards aims to achieve interoperability and compatibility in the global IoT, independently of vertical market solutions. Many standard organizations working in specifying a horizontal M2M middleware layer that compatibility combines various technologies. In this paper, we reviewed the approaches of ETSI, OMA and oneM2M in this direction, and analyze the specified capabilities of each standard. We also present an extensible ETSI-compliant M2M service platform. The openMTC aims to foster the execution of applied research activities and prototype development of M2M/IoT applications. The ETSI approach provides an M2M architecture with a generic set of capabilities to enable M2M services, while OMA developed several enablers that fit in M2M scenarios in different ways covering some of the features needed for M2M applications. The oneM2M consortium is working to unify the global M2M community, by enabling the federation and interoperability of M2M systems across multiple networks and topologies. However, postponing the release of their first specification results in delaying the growth of standard based M2M framework solutions.

Acknowledgment

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 604691.

References

[1] 3GPP-TS23.682, “Architecture enhancements to facilitate communications with packet data networks and applications (Release 11).” 2013.

[2] I. Ishaq, D. Carels, G. Teklemariam, J. Hoebeke, F. Abeele, E. Poorter, I. Moerman, and P. Demeester, IETF Standardization in the Field of the Internet of Things (IoT): A Survey, vol. 2, no. 2. 2013, pp. 235–287.

[3] CISCO, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018,” 2014.

[4] A. Zaslavsky, C. Perera, and D. Georgakopoulos, “Sensing as a Service and Big Data,” in Proceedings of the International Conference on Advances in Cloud Computing (ACC), 2012.

[5] ETSI TS 102 935 v2.1.1, “Applicability of M2M architecture to Smart Grid Networks ; Impact of Smart Grids on M2M platform,” 2012.

[6] C. Hongsong, F. Zhongchuan, and Z. Dongyan, “Security and Trust Research in M2M System,” in 2011 IEEE International Conference on Vehicular Electronics and Safety (ICVES), 2011, no. 20090460245, pp. 286–290.

[7] ETSI TS 102 690 v1.1.1, “Machine-to-Machine communications (M2M); Functional architecture,” 2011.

[8] ETSI TS 102 689 V1.1.2, “Machine-to-Machine communications (M2M); M2M service requirements,” 2011.

[9] ETSI TS 102 921 v1.1.1, “Machine-to-Machine communications (M2M); mIa, dIa and mId interfaces,” 2012.

[10] oneM2M, “OneM2M.” [Online]. Available: http://onem2m.org/.

[11] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song, “Toward a standardized common M2M service layer platform: Introduction to oneM2M,” IEEE Wireless Communication, vol. 21, no. 3, pp. 20–26, Jun. 2014.

[12] oneM2M-TS-0001 -V0.8.0, “oneM2M Functional Architecture Baseline.” 2014.

[13] “OMA Next Gereration Servicres Interface V1.0.” [Online]. Available: http://technical.openmobilealliance.org/Technical/release_program/ngsi_v1_0.aspx .

[14] OMA-TS-DM_Protocol-V2, “OMA Device Management Protocol V2.0.” Open Mobile Alliance (OMA), 2013.

[15] OMA-TS-LightweightM2M-V1, “Lightweight Machine to Machine Technical Specification,” 2013.

[16] Z. Shelby, K. Hartke, and C. Bormann, “RFC 7252: The Constrained Application Protocol (CoAP).” p. 112, 2014.

[17] ETSI-TS103.092 v1.1.1, “Machine-to-Machine communications (M2M); OMA DM compatible Management Objects for ETSI M2M,” vol. 1. p. 20, 2012.

[18] “OpenMTC platform.” [Online]. Available: http://www.open-mtc.org/index.html.

[19] A. Elmangoush, H. Coskun, T. Magedanz, and N. Blum, “An Approach to Expose M2M Services over OMA Next Generation Service Interface,” in 17th International Conference on Intelligence in Next Generation Networks (ICIN), 2013, pp. 152–159.