TOWARD A STANDARDIZED COMMON M2M SERVICE LAYER PLATFORM: INTRODUCTION TO ONEM2M

JÖRG SWETINA, GUANG LU, PHILIP JACOBS, FRANCOIS ENNESSER, AND JAESEUNG SONG

ABSTRACT

At present, most M2M solutions in different industries use proprietary systems that often comprise all layers, from physical to application, to provide their specialized M2M services to customers. These proprietary systems make it difficult to extend systems to support new services, integrate new data, and interoperate with other M2M systems. This issue motivated various standard organizations to establish a new partnership project, the "oneM2M Global Initiative," to standardize a common M2M service layer platform for globally applicable and access-independent M2M services. This article presents a snapshot of the latest progress in oneM2M standardization such as the requirements, agreed architecture, candidate protocols, security aspects, and device management and abstraction technologies.

INTRODUCTION

Today many industries rely on vertically specified machine-to-machine (M2M) solutions that involve extensive use of specifically customized hardware and software, which are typically designed only for that industry's service. Design, deployment, and maintenance of such proprietary solutions can cause high capital and operational expenditures (CAPEX and OPEX). Thus, there is a strong need to develop a horizontal common platform across a number of industry verticals, which can increase the efficient development of M2M solutions [1, 2]. Furthermore, a common M2M service platform is also needed to facilitate multi-industry M2M applications such as smart grids and smart cities, and to enable seamless M2M deployments among heterogeneous M2M systems [3].

The oneM2M Global Initiative[1] is an international partnership project, established in order to cooperate in the production of globally applicable, access-independent M2M service layer specifications related to M2M solutions. This common service layer is intended to be embedded within various hardware and software components to ensure M2M devices can communicate on a global scale. oneM2M was initiated by seven telecom standards defining organizations: Association of Radio Industries and Businesses (ARIB) and Telecommunication Technology Committee (TTC), Japan; the Alliance for Telecommunications Industry Solutions (ATIS) and Telecommunications Industry Association (TIA), United States; the China Communications Standards Association (CCSA), China; the European Telecommunications Standards Institute (ETSI), Europe; and the Telecommunications Technology Association (TTA), Korea. Their member companies, currently around 270, actively contribute to oneM2M. In addition, various alliances and industry fora such as the Open Mobile Alliance (OMA), Broadband Forum (BBF), Continua Health Alliance, and the Home Gateway Initiative (HGI) have recently become oneM2M partners.

The main objective of oneM2M is to minimize M2M service layer standards market fragmentation by consolidating currently isolated M2M service layer standards activities and jointly developing global specifications. Toward this goal, the seven founding standards development organizations (SDOs) pooled their existing M2M documents, stopped overlapping M2M service layer work, and reached out to other SDOs and fora. oneM2M is reusing, as far as possible, existing specification work, such as that by the Broadband Forum (BBF) and Open Mobile Alliance (OMA). Specifically, oneM2M is planning to provide the following benefits to the M2M ecosystem:

- Boost M2M economies of scale and shorten time to market by removing the need to develop common components.
- Simplify development of applications by providing a common set of application programming interfaces (APIs).
- Leverage existing worldwide networks for enhanced potential of services and to expand business opportunities based on interoperable standards.
- Provide evolution and interoperability of standard functions support.

Relative to the broad body of research work and various other standards aiming to provide reference architectures, models, and frameworks

Joerg Swetina is with NEC Laboratories Europe.

Guang Lu is with InterDigital Communications.

Philip Jacobs is with Cisco Systems.

Francois Ennesser is with Gemalto.

JaeSeung Song is with NEC Laboratories Europe and Sejong University.

[1] *http://www.onem2m.org/.*

1536-1284/14/$25.00 © 2014 IEEE

for M2M/Internet of Things (IoT), oneM2M through its global membership is generally aware of other efforts, which enables consideration of supporting best of breed features of research and standards plus interworking with other standards. For example, IoT Architecture (IoT-A) [4], which is a EU project for developing a unified IoT reference model, has many similarities with the oneM2M reference architecture in terms of terminologies and the defined domain model for the basic concepts of the IoT domain. Regarding interworking with other standards, for example, oneM2M is working on a protocol binding for Message Queuing Telemetry Transport (MQTT). In the case of collaboration with other standardization bodies, such as the Third Generation Partnership Project (3GPP), which have been working on M2M capabilities [2], liaisons between the two are examining how information should be exchanged between the two solutions.

Many parts of the architecture and protocol work in oneM2M are currently ongoing, and the first release of a set of specifications is expected to be released by the third quarter of 2014. At the time of writing, fundamental parts of the architecture work have already been developed.

The oneM2M technical working groups are structured as follows: Requirements (WG1), Architecture (WG2), Protocols (WG3), Security (WG4), and Management, Abstraction and Semantics (WG5). Each working group performs technical work and targets completion of their work within the agreed timeframe resulting in technical specifications. This article introduces the latest standard activities of each working group in subsequent sections. After that, we discuss future directions and challenges in the domain of M2M followed by conclusions.

Use Cases and Requirements

oneM2M started its work in WG1 by collecting a large number of use cases to explore and extract aspects that are relevant to M2M. The next phase consisted of formulating and clarifying requirements on the oneM2M system. One explicit goal of WG1 was to identify the specific needs of the different ecosystems of the industry involved in M2M. It is obvious that oneM2M should interwork with existing M2M solutions that are already deployed as well as provide a basis for future services that do not yet exist.

Use Cases and Definitions

The use cases that were collected in oneM2M in a published technical report[2] [5] cover a wide area of industry segments, including energy, enterprise, healthcare, public services, residential, transportation, and others. Each use case includes description, actors, pre-conditions, triggers, flow of sequence of interactions among actors and system, post-conditions, illustrations, and potential requirements. The use cases express the needs of the affected industry segment on its specific M2M systems. However, in many cases they were written in a way that facilitates identification of features that a "general-purpose" M2M system should support. Another goal achieved was the "translation" and consolidation of specific M2M terms used in various industry segments into more generic definitions that can apply to different industries [6].

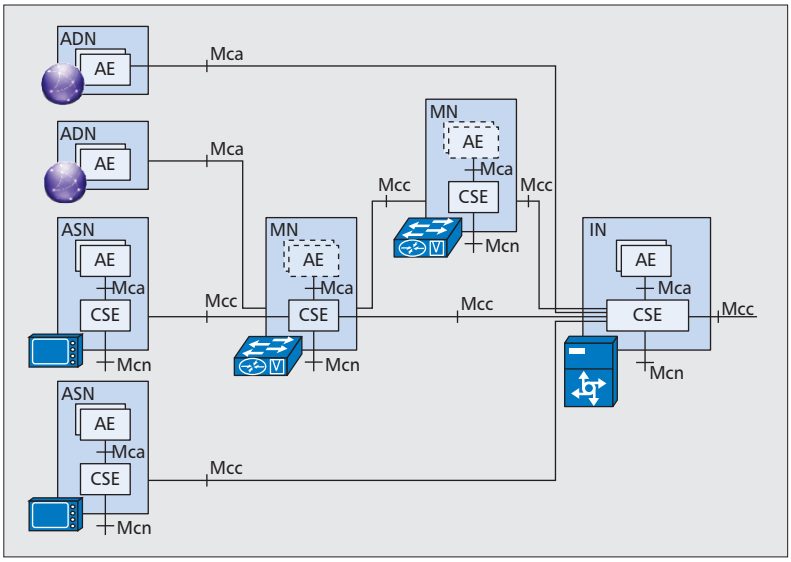

Figure 1. oneM2M functional architecture.

Requirements

Starting from the collected use cases, the next step was to formulate requirements for the oneM2M service layer. The requirement specification for oneM2M release 1 has been approved [7]. The requirements for oneM2M assume existence of the following roles:

- The *user* (individual or company) uses an end-to-end M2M solution. Some parts of that M2M solution include M2M common services, compliant with oneM2M specifications.
- An *M2M application* contains all industry (application) specific aspects. The application service provider provides the M2M application service.
- *M2M common services*, operated by an M2M service provider, provide support for a broad range of applications. These M2M common services are the target of oneM2M specification work.
- Finally, an *underlying network* may provide connectivity and related services. Wide area networks are usually operated by a network operator.

Requirements were categorized into seven functional requirement groups: Overall System, Management, Abstraction and Semantics, Security, Charging, Operations, and Communication and Processing. Additionally, a small set of non-functional recommendations was formulated.

System Architecture

oneM2M system architecture design work is focused on providing both basic functionalities (e.g., registration and message handling) and various advanced functionalities (e.g., interworking with other systems). To achieve this, oneM2M defines a common service layer providing M2M services, which is independent of the underlying networks.

[2] *All oneM2M documents (technical specifications and reports) are available from ftp://ftp.onem2m.org/Deliverables/.*

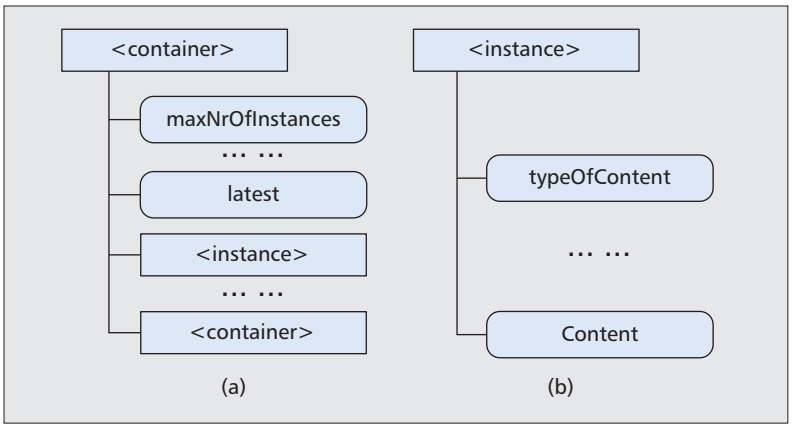

Figure 2. Example of oneM2M resources: a) `<container>` resource; b) `<instance>` resource.

Reference Architecture

The latest oneM2M functional architecture [8] is illustrated in Fig. 1. The oneM2M system is formed by functional entities called nodes. These are known as the application dedicated node (ADN), application service node (ASN), middle node (MN), and infrastructure node (IN). Nodes consist of at least one oneM2M common services entity (CSE) or one oneM2M application entity (AE). A CSE is a logical entity that is instantiated in an M2M node and comprises a set of service functions called common services functions (CSFs). CSFs can be used by applications and other CSEs. An AE is a logical entity that provides application logic, such as remote blood sugar monitoring, for end-to-end M2M solutions.

oneM2M currently defines three reference points (i.e., Mca, Mcc, and Mcn), as indicated in Fig. 1. The Mca reference point enables AEs to use the services provided by the CSE. The Mcc reference point enables inter-CSE communications. The Mcc' reference is similar to Mcc, but provides an interface to another oneM2M system. The Mcn reference point is between a CSE and the service entities in the underlying networks, such as device triggering service provided by Third Generation Partnership Project (3GPP) networks.

Common Service Entities and Functions

oneM2M has specified a set of core common service functions (CSFs) for its initial release. Some CSFs provide administrative functions for the service layer and other CSFs; for example, the registration (REG) CSF provides a means for an AE or a CSE to register to a CSE and be able to use the services provided by that CSE. The security (SEC) CSF enables secure establishment of service connections and data privacy. An AE and a service layer management (ASM) CSF provides functions to configure, troubleshoot, and upgrade CSEs and AEs. A device management (DMG) CSF manages device capabilities such as firmware updates. The communication management and delivery handling (CMDH) CSF is responsible for service layer message delivery. The network service exposure (NSE) CSF serves as the anchor point between the service layer and services provided by different underlying networks.

Some CSFs provide value-added services to registered AEs and CSEs. The data management and repository (DMR) CSF is responsible for user data storage and processing. Users can subscribe and get notifications of changes in the data. The discovery (DIS) CSF provides a means to make the services and resources discoverable by other CSEs and AEs. A subscription and notification (SUB) CSF manages subscriptions to changes on the oneM2M platform. The service session management (SSM) CSF supports end-to-end service layer sessions. The service charging and accounting (SCA) CSF provides mechanisms to support service-layer-based charging. A group management (GMG) CSF supports bulk operations and manages group membership. The location (LOC) CSF allows M2M AEs to obtain geographic location information of an entity and receive location-based services. In addition to CSFs, a CSE includes a service enabler to ensure the extensibility of services.

Resource Model and Structure

oneM2M adopted a resource-based data model. All services are represented as *resources*. A resource is a data structure that can be uniquely addressed by a uniform resource identifier (URI), which is a string of characters used for identifying a resource. Resource(s) are associated with the above mentioned CSFs to support CSF functions. Figures 2a and 2b show the `<container>` and `<instance>` resources, respectively, for the DMR CSF to store user data. The `<container>` resource contains multiple attributes (e.g., maximum number of instances, latest instance) and sub-resources (e.g., `<instance>` and `<container>`). The `<instance>` resource represents a data instance in the container, and the content of the data instance may be opaque to the oneM2M platform. The operations of the resources can be achieved by the (`CREATE`, `RETRIEVE`, `UPDATE`, and `DELETE`) CRUD [9] verbs being applied to the resources and attributes.

oneM2M Protocols

The objective of WG3 is to standardize service layer protocols, API primitives, parameter definitions, and protocol bindings via the publication of technical specifications.

Aspects of the oneM2M Protocols

oneM2M is concerned with four aspects of protocols that are being covered in the oneM2M protocol: *service layer protocols*, *encapsulation of existing protocols*, *binding with underlying protocols*, and *interworking with non-oneM2M systems*.

Service Layer Protocols — The inputs to the development and specification process of the oneM2M protocol include multiple drivers: requirements, functional architecture, security specifications, RESTful design style, plus best practices and considerations of encapsulated usage of existing protocols, binding to underlying protocols, and potential interworking with protocols of non-oneM2M systems. Specifically, inputs include func-

tional descriptions, identifiers, URI and resource structures, generic flows and procedures, security procedures, and information flows. The oneM2M protocol specification will cover resource details (e.g., typing, relationships), procedures among oneM2M entities, and implementation procedures (e.g., message sequences), APIs, and resource representations (e.g., messages) that cross the reference points (i.e., Mca, Mcc, and Mcn).

Encapsulation of Existing Protocols — A goal of oneM2M protocol development is to leverage existing protocols where they fulfill oneM2M's requirements and functional architecture. As such, existing protocols for device management and security protocols are being encapsulated within the oneM2M protocol. oneM2M should provide capability translating oneM2M resources to non-oneM2M resources of encapsulated protocols.

Binding with Underlying Protocols — As a horizontal middleware protocol, oneM2M will rely on underlying protocols for message transport and delivery. In order to design simple and efficient mapping, the style, messages, and information model of underlying protocols are being considered in the design of the oneM2M protocol. WG3 is developing standardizations of bindings for flows to specific transport protocols.

Interworking with Non-OneM2M Systems — Since oneM2M systems will need to communicate with non-oneM2M systems in various industry segments, interworking of messages and the information model with underlying protocols are being considered in the design of the oneM2M protocol. Procedures and information flows to interface with non-oneM2M protocols will be standardized.

The Protocol WG is producing a protocol analysis technical report [10], which analyzes existing protocols such as CoAP [11], XMPP,[3] DDS,[4] and MQTT[5] in order to support the above aspects.

Current Message Sequences

While different messaging schemes will be supported by oneM2M, a distributed request/response message sequence is one scheme that will be supported over the Mca and Mcc reference points. AEs and CSEs can use request/response message sequences to transfer information among resources. The request and response messages each support a set of mandatory and optional indicators, allowing for both lightweight and rich messaging.

The emergent request message supports features including:

- An identifier for correlation with responses and so on
- Timestamps to indicate when the message was built, when its operation should be executed, and when it expires
- What kinds of responses are desired, where the results should be put, and how long they should persist
- How and when the message should be delivered
- Whether and how message aggregation to the same destination should occur

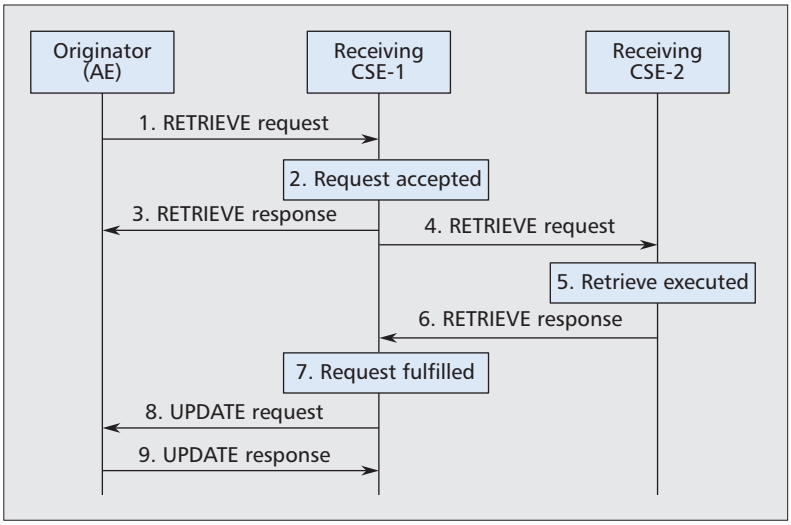

Figure 3. Non-blocking asynchronous mode.

The emergent response messages indicate whether a request has been accepted but not yet executed, or whether the operation execution was successful or unsuccessful. They support the request identifier, timestamps, and where the results have been put.

Blocking and Non-Blocking Synchronous and Asynchronous Modes

The oneM2M request/response message sequence is being designed to be able to support different modes of operation based on different use cases. In blocking mode, a request may be forwarded via multiple nodes before reaching the node which hosts the target resource, and the response may be forwarded via multiple nodes back to the originator. During this time, if the originator is single threaded, it is blocked from further processing. Furthermore, the round-trip time may be lengthened due to message aggregation or scheduled communications. If this delay is a problem for the originator, it can use a non-blocking mode.

In the non-blocking case, where an originator is not able to receive asynchronous messages, it will need to initiate all messages and receive responses synchronously (within some timeout period). The receiver of a request can respond with a response indicating that the message was received, but the operation has not yet been executed. Following this, the originator can repeat the request until the result is available. If the originator is able to accept asynchronous notification messages (Fig. 3), the originator sends a request to the receiver, which provides a timely message accepted response. Later, that receiver originates a request containing the result or reference to the result to the originator, which confirms receipt of this request with a response.

Security and Privacy

One of the main challenges for an M2M service platform is to accommodate a very diverse range of usage, including leisure applications with few security risks as well as life-critical ones such as

[3] *http://www.xmpp.org/.*

[4] *http://portals.omg.org/dds/.*

[5] *http://www.mqtt.org/.*

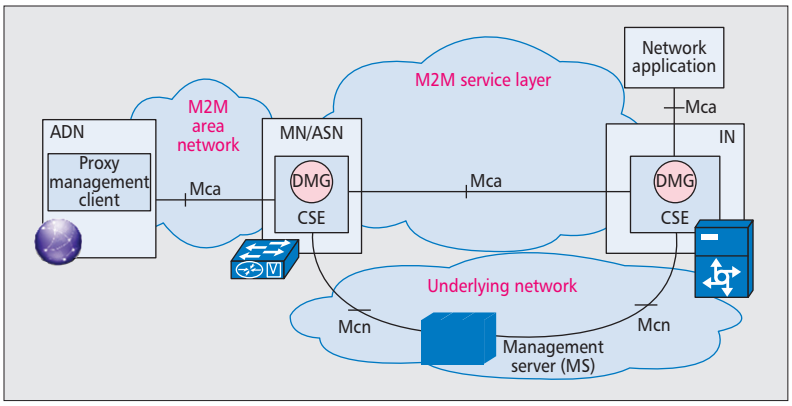

Figure 4. oneM2M device management architecture.

smart power distribution networks. Furthermore, even when there are no direct security risks, privacy exposure arising from personal devices providing information about ourselves without our awareness may generate important liabilities, such as targeted burglaries or kidnapping. Finally, many M2M devices have constrained resources, while security and privacy protection techniques tend to be resource hungry (e.g., the weight of a DTLS [12] implementation over Constrained Application Protocol, CoAP).

Security

The M2M service platform itself is an asset that requires appropriate security measures to protect its availability as well as integrity and nonrepudiation of service messages to support charging. However, in typical M2M use cases, the underlying network used for data transport, such as a cellular network, may already implement its own security measures, and regulatory requirements to avoid exposure of private data manipulated by M2M applications to third parties have to be acknowledged. The M2M application generally requires establishment of an additional end-to-end security layer at the application level. Resource constrained M2M devices may not be capable of accommodating multiple encryption layers, especially when they rely on different cryptographic algorithms, which may happen if they are specified independently by different actors. Therefore, an expected added value of an M2M service platform in terms of security is to hide transport network specificities, while relying on them when possible and completing them to expose end-to-end security services to the applications via an interoperable programming interface.

Privacy

In addition to applying privacy by design principles and enabling data subjects to control access to their data, the key to addressing privacy concerns lies in the establishment of a trusted security ecosystem among the multiple stakeholders of M2M applications. No matter how the problem is addressed, security and especially data confidentiality rely on secrets (called credentials) that need to be distributed and shared between communicating entities. A unified infrastructure for credential distribution, operated by a party that is trusted by all stakeholders, is therefore the best way to mutualize common security requirements in M2M devices and optimize deployment costs, while ensuring that all actors can trust preservation of the confidentiality of their data. To enable flexible ecosystems even in the most complex environments such as smart grids or smart cities, standards must support independent actors acting as trusted third parties to provide M2M security. This requires M2M service platform specifications that allow dissociation between roles involved in data dissemination, which do not require special trust, and those responsible for establishing and preserving security, which bear specific liabilities in case of security breaches. The SEC WG is analyzing [13] use cases for threats, mapping them to the security requirements and derives possible security mechanisms to realize the security features for oneM2M Release 1.

Device Management and Device Abstraction and Semantics

The Device Management and Device Abstraction & Semantics (MAS) WG focuses on two aspects: device management (DM) and device abstraction including semantics consideration (ABS). DM provides solutions for various aspects, such as configuration, fault management, and diagnostics, while ABS supports interworking mechanisms that enable the oneM2M system to communicate with external systems and technologies.

Device Management

Since there are several state-of-the-art technologies (e.g., OMA DM [14] and BBF TR069 [15]) that are relevant to oneM2M management capabilities, the MAS WG first collects their specifications and performs analysis on the collected technologies to match with oneM2M requirements on management aspects. The technologies are then evaluated to see the possibility of leveraging all or part of those technologies by the oneM2M system to enable its management capability. These activities have delivered a technical report and triggered the initiation of two technical specifications, one for OMA DM and the other for BBF TR069.

For device management, oneM2M considers adopting an architecture depicted in Fig. 4. In this architecture style, the underlying network operator and the M2M service provider cooperate to control the end device through the management server provided by the underlying network operator. In general, an end device contains a management client that is connected to the management server so that an M2M application can manage (e.g., application software installation, configuration, firmware updates) the end device using services exposed over Mca. The DM (DMG) CSF in ASN/MN/IN manages resources representing the information. The management server provides protocol translation between CRUD commands and DM commands corresponding to the specific DM technology such as OMA-DM [14] and TR-069 [15].

Let us consider a DM example where an application wants to read the current value of a

humidity sensor that is connected to the M2M area network via BBF TR-069. The network application sends a RETRIEVE command to IN over the Mca reference point. The message is then processed by the DMG and converted into a GetParameterValues remote procedure call (RPC) method for TR-069. Based on this proprietary command, the humidity sensor reports the current humidity value to the MN, which is then forwarded to the application via the management server and IN.

At the time of writing this article, oneM2M is considering various deployment scenarios such as one in which the M2M service provider has a management server and uses the server together with the underlying network operator to control the M2M devices, which will leverage the functional architecture introduced earlier.

Device Abstraction and Semantics

Many systems in the context of the M2M domain that have been developed outside of oneM2M use different access technologies to exchange their data. For example, in the area of home automation, ZigBee Smart Energy 6 (SE2.0) and BACnet provide functionality to manage electrical appliances. Since these systems use different information models and processes, it is difficult to enable communications between these heterogeneous devices. oneM2M initiated a study item called *"Study on abstraction and semantics enablement"* [16] in order to tackle this interworking issue.

Semantic Interworking — oneM2M has to provide semantic interworking with external M2M systems in order to allow an M2M application to use devices from other technologies such as ZigBee,[7] Bluetooth,[8] and BACnet.[9] One solution described in the technical report [16] is to introduce interworking proxy functions, which could be realized as an M2M application or as part of some CSE, that map the native interface of the device into oneM2M resources that can be accessed by M2M applications. In this case, the M2M application only needs to know the information model and the semantics of the native interface provided by oneM2M.

Abstraction — Device abstraction is a concept of generalizing all of the most commonly exposed attributes in the resource representation of external (non-oneM2M) devices. This enables the oneM2M system to hide the complexity of the specific technologies used by heterogeneous devices through providing a common information model and semantics of the native interface.

A possible way described in the technical report [16] is to introduce *abstract devices* to the oneM2M system. Figure 5 shows a potential oneM2M architecture extension to support device abstraction. An external device (e.g., a KNX[10] temperature sensor) is typically registered to an M2M system with their specific attributes and commands so that an M2M application having KNX-specific knowledge can only use all exposed services via the external device. However, with the introduced concept, the external device is first registered to the CSE of an MN as a native M2M application with representation of an information model together with a corresponding abstract device. Instead of communicating with the native device, any M2M Application that understands a common information model used in oneM2M can directly communicate with the abstract device.

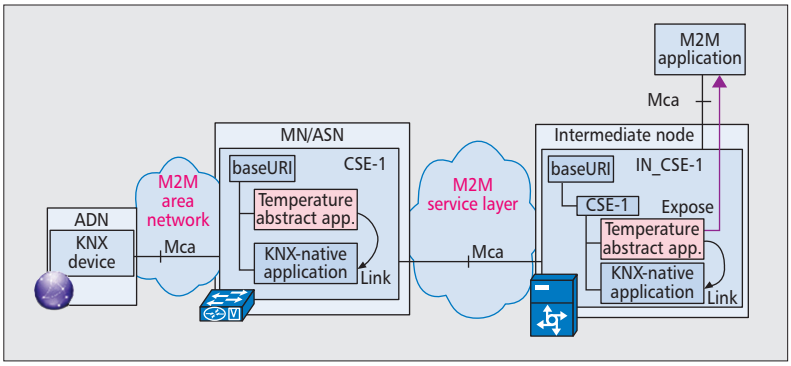

Figure 5. oneM2M architecture for device abstraction.

Discussion and Conclusions

A global standard across various industry verticals is necessary to ensure easier use of M2M technology, data interoperability, and efficient development of M2M systems. To achieve this goal, seven major SDOs formed the oneM2M global initiative for M2M standardization. In this article we review the latest standardization efforts performed in each oneM2M working group. Use cases and requirements are mainly collected from vertical M2M domains and existing specifications of member SDOs. The collected requirements are then used to derive a set of functionalities that the oneM2M functional architecture has to provide. In order to support addressed functions, oneM2M includes support of a RESTful resource-based architecture style and protocols. Standardization activities for other fundamental functions, such as security, device management, and device abstraction and semantics, are also introduced.

The oneM2M Release 1 specifications for realizing a minimum deployable M2M system, mostly described in this article, is scheduled for publishment by the third quarter of 2014. After the initial release, the oneM2M system could be extended in various aspects, such as plug-in of value-added new services, supporting local area network interworking, and adding advanced security functionalities. Another dimension in enhancing the oneM2M system is to support interworking with underlying networks, in particular for 3GPP. In Release 12, 3GPP only supports a device triggering feature for machine type communications (MTC); therefore, oneM2M has only been standardizing the triggering aspect for the interworking with underlying networks [2]. However, a new work item, Service Exposure and Enablement Support (SEES), has been agreed by 3GPP SA1 to support oneM2M interworking for 3GPP. We believe this new work item will enable M2M applications to use various network services exposed by underlying 3GPP networks.

[6] *http://zigbee.org/Standards/ZigBeeSmartEnergy/Overview.aspx.*

[7] *http://www.zibgee.org/.*

[8] *http://www.bluetooth.org/.*

[9] *http://www.bacnet.org/.*

[10] *http://www.knx.org.*

Acknowledgments

Professor Song is supported by grants from the IT R&D program of MKE/KEIT. [10041262, Open IoT Software Platform Development for Internet of Things Services and Global Ecosystem]. All authors made equal contributions to the study and the publication.

References

[1] G. Wu *et al.*, "M2M: From Mobile to Embedded Internet," *IEEE Commun. Mag.*, vol. 49, no. 4, 2011, pp. 36–43.
[2] T. Taleb and A. Kunz, "Machine Type Communications in 3GPP Networks: Potential, Challenges, and Solutions," *IEEE Commun. Mag.*, vol. 50, no. 3, 2012, pp. 178–84.
[3] J. Song *et al.*, "Connecting and Managing M2M Devices in the Future Internet," *Mobile Networks and Applications*, 2013, pp. 1–14.
[4] Internet of Things Architecture project, http://www.iot-a.eu/public/front-page.
[5] oneM2M-TR-0001, "oneM2M Use Cases Collection Technical Report," v. 1.0.0, Oct. 2013.
[6] oneM2M-TR-0004, "Definitions and Acronyms," v. 0.3.0, Oct. 2013.
[7] oneM2M-TS-0002, "oneM2M Requirements Technical Specification," v0.6.2, Oct. 2013, http://www.onem2m.org/.
[8] oneM2M-TS-0001, "oneM2M Functional Architecture Technical Specification," v. 0.2.1, Oct. 2013, http://www.onem2m.org/.
[9] L. Richardson and S. Ruby, *RESTful Web Services*, O'Reilly, May 2007.
[10] oneM2M-TR-0009, "oneM2M Protocol Analysis Technical Report," v. 0.3.3, Oct. 2013.
[11] Z. Shelby, K. Hartke, and C. Bormann, "IETF Internet Draft: Constrained Application Protocol (CoAP)," 2013, http://tools.ietf.org/html/draft-ietf-core-coap-18.
[12] E. Rescorla and N. Modadugu, "IETF RFC 6347: Datagram Transport Layer Security," 2006, http://www.hjp.at/doc/rfc/rfc4347.html
[13] oneM2M-TR-0008, "Analysis of Security Solutions for oneM2M System," v0.2.1, Oct, 2013
[14] OMA Device Management Protocol, v. 1.2, OMA-TS-DM_Protocol-V1_2, Open Mobile Alliance, 9 Feb. 2007
[15] BBF TR-069, CPE WAN Management Protocol v. 1.1, issue 1, Amendment 2, Dec. 2007.
[16] oneM2M-TR-0007, "Study on Abstraction and Semantics Enablement," v. 0.5.0, Oct, 2013.

Biographies

JÖRG SWETINA (joerg.swetina@neclab.eu) studied chemistry and mathematics at the University of Vienna, Austria, and later conducted research in theoretical chemistry. Moving from academia to industry, he led a development team dealing with GSM call processing and software testing at Siemens Austria. Since the early days of 3GPP he represented Siemens and later Nokia Siemens Networks in standardization bodies including ETSI SMG, 3GPP, OMA, and others. In 2008 he moved to NEC Europe in Heidelberg, Germany, continuing his work on standards. In addition to 3GPP, his current field of interest covers machine-to-machine communication. He was active in ETSI TC M2M and is currently acting as Vice Chair of the Requirements Group of the oneM2M Global Initiative organization.

GUANG LU (guang.lu@interdigital.com) is currently a senior engineering manager working on the M2M and Internet of Things (IoT) at InterDigital. She joined the R&D department of InterDigital in 2001 and has worked on system architecture and solutions on various technologies, including M2M and Internet of Things, LTE, UMTS, IEEE 802.11, IEEE 802.21, and media mobility. She has contributed to various standard organizations including oneM2M, ETSI TC M2M, IETF, IEEE, and 3GPP. Prior to joining InterDigital, she worked on broadband wireless access solutions at Nortel and information systems at Unisys.

PHILIP JACOBS (phjacobs@cisco.com) holds a B.Eng. (honors) in electrical engineering and an M.Sc. in engineering management. He has contributed to and led product development at several network and computing companies, and delivered chip design services since 1979. He has been issued 14 patents, plus 2 allowed and 8 filed. He has contributed to and served on several standards and industry bodies including HomePlug (BoD, WG chair, and contributor), IPSphereForum (WG co-chair, BoD alternate, contributor), ITU-T (NGN-FG contributor, SG16/13 co-rapporteur/contributor/editor of series of five H.741 standards, IPTV-GSI contributor), ATIS (M2M-FG contributor, eHealth-LT/FG contributor, SDN-LT/FG co-chair and contributor, SDN/NFV-FG co-chair and contributor), ETSI M2M TC contributor, and oneM2M (WG1 rapporteur and contributor, WG2 contributor, WG3 vice-chair, rapporteur, and contributor).

FRANCOIS ENNESSER (Francois.Ennesser@gemalto.com) has been involved in embedded secure applications development and standardization activities since 1999, first with Bull CP8 and then with Schlumberger Systems, Axalto, and Gemalto. He participated to the definition of secure over-the-air protocols for remote management of secure elements and contributed to the definition of an enhanced communication interface for smart cards. He initiated and chaired the initial definition of the Inter-Chip USB low-power embedded adaptation of the USB interface within the USB Implementers Forum. He has been focusing on machine-to-machine standardization since 2008 and is currently chairing the Security Working Group of the oneM2M Partnership Project. He contributes to EU-level initiatives on smart grids with a focus on critical infrastructure protection and privacy preservation. He holds a French Engineer's degree from ESIEE and an M.Sc. in electrical engineering from the University of Southern California.

JAESEUNG SONG (jssong@sejong.ac.kr) is an assistant professor in the Computer and Information Security Department of Sejong University. His research areas include IoT/M2M platforms, big data analytics, and the reliability and security of networked software systems. Prior to his current position, he worked for NEC Europe Ltd. from 2012 to 2013 as a leading standard senior researcher working on R&D projects and IoT/M2M standardizations. He also worked for LG Electronics from 2002 to 2008 as a senior researcher. He occupied leadership positions in the 3GPP and oneM2M standard groups as a rapporteur and contributor. He received a Ph.D. from Imperial College London in the Department of Computing, United Kingdom, in 2012. He holds B.S. and M.S. in computer science from Sogang University in 2000 and 2002, respectively.