Towards Standard M2M APIs for Cloud-based Telco Service Platforms

Asma Elmangoush
Electrical Engineering and Computer Sciences Faculty
Technical University Berlin
Berlin, Germany
asma.a.elmangoush@campus.tu-berlin.de

Adel Al-hezmi, Thomas Magedanz
Fraunhofer FOKUS- NGNI
Berlin, Germany
{adel.al-hezmi,
thomas.magedanz}@fokus.frunhofer.de

ABSTRACT

Machine-to-Machine (M2M) communication technologies is emerging as one of the major trends shaping the development of services in the Future Internet sector. Additionally, cloud computing technologies have been utilized in building large-scale M2M systems, as they could potentially offer more advance solutions for managing data monitoring and service elasticity to support the needs of different consumers. Today, service providers are building a new eco-system with partner vendors to offer new innovative services for Smart Cities. From the users/applications perspective, three concepts aim to deliver Smart service: M2M communication, ubiquitous computing and ambient intelligence. In this paper we identify the requirements for centric service enablers to support the development of desired Smart Cities systems. Also the paper reviews the ongoing work in standardization organizations in this field. Providing standard interfaces to M2M platforms in a network independent way will allow applications deployment by 3rd party developers across variety of end user devices.

Categories and Subject Descriptors

D.2.2 [Design Tools and Techniques]; D.2.10 [Design]: *Representation*; D.2.9 [Management]: *Software configuration management*

General Terms

Management, Design, Economics, Reliability, Standardization.

Keywords

Open API; M2M; Cloud Computing; Smart Service.

1. INTRODUCTION

The concept of "Open innovation" is developing a new ecosystem for open business models. In his book [1], Chesbrough describes the "open Innovative " paradigm which assumes that "firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as the firms look to advance their technology". In comparison to Closed Innovation, Open Innovation offers more potential for operators to leverage their infrastructure and enhances solutions development by involving ideas from third party devolving community. Today it's becoming more convincing for telecom operators and services providers the necessity to reach out to developer community, in order to allow the creation of more innovative services [2]. Thus new eco-system is built with partner vendors to offer toolkits and Open Application Programming Interfaces (APIs) for building new innovative services.

A plethora of new APIs has been released since 2005, rising from approximately 235 publically available Open APIs to more than 9000 in 2013, according to Programmableweb [3]. Many telecom operators and services providers have released open APIs to attract 3rd party developers to introduce their ideas and deploy new services in less time and effort. This extension of the business model:

- Offers new revenue opportunities for existing company assets.
- Provide self-service options for customers.
- Building a community around a brand or campaign.
- Refactoring business processes so they can be delivered as services.
- Exposing core functions to internal and external users in a controlled manner.
- Facilitating a merger or acquisition of another business.

The ITU-T Recommendation of the Internet of Things (IoT) [4] has specified open Web-based service environment as one of the core competencies to support a ubiquitous networking. It is expected that the IoT will incorporate billons of physical and virtual "things" to the communication system and allowing them to seamlessly integrate into a global information space. Machine-to-machine (M2M) is a communication concept (aka Machine Type Communication (MTC)) representing an essential part of the IoT concept, where machines communicate with each other with little or no human interaction. Different from human-to-human (H2H) and human-to-machine (H2M) communication, which mainly involves voice calls, messaging and web browsing, M2M communication provides a new paradigm. This paradigm aims to increase the level of system automation in which the devices and systems can exchange and share data. As current networks are designed to support human-to-human communication, it is essential that the technologies evolve to develop competitive capabilities to support M2M communications. The principle of limiting human interaction in M2M systems and integration billions of devices (with dissimilar computational capabilities) into it, demands a fully network and application agnostic system. Discovering and managing connections between ubiquitous devices and applications are essential capabilities for M2M systems.

In most existing M2M solution implementations the application logic, device and sensors are tightly coupled. However, different standards try to provide a solution for this 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 discover process of connected devices and the usage of each other's resources on demand. Thus the standard defines a horizontal "middleware" for diverse M2M applications on top of heterogonous sensor technologies. On the other hand, the concept of connected "Things" has been introduced by some IoT research activities, 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).

Today service providers are building an eco-system with partner vendors to offer new innovative services. This paper explores the benefits of having a service creation platform to aid in developing new M2M services for Smart Cities. It outlines the elements of a Software Development Kit (SDK), which eases development process and hides complexity. Thus, SDKs allows a shorten development cycle, helps in maximizing productivity and enables developers to focus on creativity. The remainder of this paper is organized as follows: Section 2 includes a literature review of related work in APIs development and service provision middleware. Section 3 contains an analysis of Open API standardization activities for telecom service provision. Section 4 presents an analysis of the requirements and architecture to support Smart Communications and Service Enablers. Section 5 provides an overview of proposed SDK for supporting Smart Service Creation. Finally Section 6 concludes the paper.

2. LITERATURE REVIEW

2.1 Application Programming Interface (API)

API is becoming a common expressing in software engineering, referring to any set of related operations in a specific domain that allow one software component to access programmatically another component. Software components, libraries and frameworks usually provide APIs that expose their functionality to other software elements. Generally, the use of APIs facilities the collaborative software development through:

- Hiding implementation details of software modules, which decreases the dependency between them, and in consequence simplify the coordination of collaborative software development [5].
- Enabling the reusability of the set of actions or algorithms, which can be performed by a special set of APIs.
- Enhancing the extensibility of features, as different stakeholders are able to focus on their own competencies and add value to the end product.

Using APIs is becoming a larger part of programming. The numbers of availably APIs are growing rapidly, aiming to mediate the application delivery platform, and allowing a worldwide developer community to contribute in the process of providing value-added applications rapidly. This paradigm helps operators and service providers to increase profits and reduce time of offering new innovative services.

2.2 Service Provision Technologies

Conventionally, services in telecom were built within the network based on proprietary hardware/software solutions. Using Intelligent Network (IN) technology, the concept of service independent platforms was introduced. Later, object orientation and distributed middleware took off and APIs were introduced to allow for flexible service creation and make service 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 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 (AS) 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].

It is expected that billions of nun-human objects will be connected to the Internet by the end of this decade [7], which will increase the opportunities of developing novel machine-oriented services. Different from Human-to-Human (H2H) and Human-to-Machine (H2M) communication, which mainly involves voice calls, messaging and web browsing, M2M communication provides a new paradigm. This paradigm aims to increase the level of system automation in which the devices and systems can exchange and share data. Newer platforms are essential to overcome the limitations of current frameworks that ware designed to support H2H communication, and develop competitive capabilities to support M2M communications. Smart Cities frameworks are expected to employ multiple networking technologies for the best service provisioning; the presence of different actors and disparate transport technologies has the effect of defining complex scenarios with important challenges for interconnection, interworking and interoperation between network technologies and telecommunication operators. A Smart City is an integrated system of multiple systems forming a closed loop and characterized by functions: sensing, information management, analytics and modelling, and influencing outcomes. Each system produces its own information and consumes others' information in a well-defined urban planning. Although the phrase Smart City have been used frequently, there is no clear definition of it [8], however defining a Smart City 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” [9] can be well adopted in the context of this research.

2.3 Cloud Computing

Recently, the paradigm of shifting the location of computing infrastructure to the network gained more momentum. This concept referred to as “Cloud Computing”, aims to present computing resources (i.e. software, platform infrastructure) on the Internet as a service that can be used on demand bases. Cloud computing is a general term for anything that involves delivering hosted services over the Internet. The term was inspired from the cloud symbol frequently used to represent the Internet in diagrams as an abstraction of the underlying infrastructure it represents. According to NIST (National Institute of Standards and Technology) Cloud Computing is: “*A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*“ [10]. What differentiate cloud services from traditional hosting is its on-demand availability to consumers over broad access networks, flexibility and scalability in provisioning required recourses on the fly, and highly automation avoiding the expensive personnel support. Cloud computing is rapidly increasing in popularity as different stakeholders, business organizations and government agencies are outsourcing hardware and maintenance, and instead focus on software. The concept brings several benefits including [11]:

- Reducing the management costs of hardware and software resources.
- High ability to automatically scale services and infrastructures.
- Increased service availability and reliability resulting from the replication of service components and rapid deployment of new service instances.
- Pay-per-use models that let users pay only for actual resource consumption.
- Improved time-to-market for services owing to reduced development and infrastructure deployment times.
- Cloud interoperability, which lets users deploy a service on multiple clouds, thus providing unlimited scalability and optimized service performance.

Fortunately, cloud solution architectures involve technology components from different fields and many of challenges have been already addressed to a certain degree by various research communities, mostly virtualization and service elasticity. These challenges will play a critical role in defining the technology roadmap for developing future Platform as a Service (PaaS) cloud architecture. Several cloud-based platforms have been implemented to support monitoring and data sharing of various activities such as energy consumption, such as AlertMe [12] and Xively [13]. While these systems can manage different types of data and enable different applications to exchange and analyze data, they do not provide a generic framework for managing complex relationships between monitoring data and monitored objects.

The OpenMTC Platform [14], developed by FOKUS fraunhofer and the Technical University of Berlin, aims to provide a cloud-based standard-oriented middleware for M2M applications and services, enabling research and development of M2M systems. The OpenMTC platform implemented features are aligned with ETSI M2M Rel.1 specifications [15], also it supports automatic elasticity scaling for application servers and database.

3. Open API Standardization Activities

Generally, the creation of standards APIs are crucial to the development of Future Internet platforms, and with consideration of the huge heterogeneity nature of networks and devices in M2M and Smart Cities platforms, the need for standard interfaces is more clear. Making the M2M platform transparent to the application layer will encourage developers to build applications for these platforms without the need to learn deep details of the operator middleware organization. However some highly recognised achievements in the standardization of open API for telecom applications are available:

- The Parlay/OSA API standards are the first standardization efforts of telecom Open APIs, established by 3GPP and ETSI. Its most significant feature is the independence of the telecommunication technology. Thus it is mostly used to publish and deploy 3rd party services. Parlay Group specifies the Parlay Web services SOAP-based API, known as Parlay X API. It abstracts the service interface from Parlay/OSA API, and wraps it using web service technology, shielding the primitiveness and complexity of the telecom protocol and control logic. Later OMA released the Next Generation Service Interfaces (NGSIs), which expand services for seamless integration of fixed and mobile networks. NGSI focuses on new requirements and API extensions for server-to-server based third party services [16]. The final version was approved and released in May 2012 [17].
- The Java Community Process (JCP) [18] provides a service level APIs to developers, including Java specification request (JSR) 281 IMS Services API and JSR 289 Session Initiation Protocol (SIP) Servlet v1.1. These specifications provide a good development environment for IP Multimedia Subsystem (IMS) networks but only through Java technology.
- OneAPI form GSMA [19], is a set of network API specifications, developed in public and based on existing Web standards. It doesn't focus on the technological innovation, but aims at setting the business rules for the 3rd party application’s developer.
- The European Telecommunications Standards Institute (ETSI) created a Technical Committee (TC) whose standardization work is mainly focusing on the service middleware layer for M2M platforms [20]. The first Release of M2M ETSI standards were finalized in 2012, and published in three parts: requirement, architecture, and protocol. The specification includes definition for a set of resource-oriented APIs, accessed via HTTP and

uses XML-based or JSON-based representations for information interchange between M2M applications and a Service Capability Layer (SCL) included in the M2M platform and gateways.

- OMA has established a cooperation work with ETSI in order to map OMA Enablers into ETSI M2M Service Capabilities [21]. The OMA Device Management (DM) protocol is considered as a stable starting point to provide the desired associations between ETSI M2M standards and OMA supporting Enablers. OMA DM has been designed to provide remote device configuration-related functions such as: configuration management, performance management, and fault management. Additionally, OMA is currently working on extending a lightweight DM protocols for M2M gateways and devices. The lightweight M2M (LWM2M) focuses on service enablement of particular Resource Constrained devices, i.e. devices that consumes low power and is limited in its CPU, memory, I/O for processing requests.

4. Smart Communication and Open Communication Server

The future structure of the innovative service industry will be a new situation that heterogeneous and widely distributed common services will mainly compose the application. The reuse of an extensible set of existing service components to create rapidly new market driven applications is a key aspect of telecommunications platforms.

Smart services gained strong momentum through the advances of technology in distributed computing, networking and sensor networks. Smart Communication middleware aims to enable seamless combination of services through providing a single point of access that support various application layer technologies for communication services and networks

The process of designing APIs is critical because it is intended to be written once and used many times and for various use cases which can influences the design. Latter changes or update may impact users due to compatibility issues. We present here a taxonomy for centric enablers that will aid in defining requirements of Smart Communication and Future Internet:

4.1 Human-to-Human (H2H) communication

Where both the participants of the session are humankind using a Smart devices (e.g. mobile, PC), and can exchange voice, video or data during the time of the session. Applications targeting H2H communications require service enablers that could value-add the communication session between participations, such as: Address book, presence and file transfer enables.

Table 1. Service enablers required for some application domains in Smart City

Enabling Services		Business / Collaboration	Utility Metering	Smart Buildings	E-Health	Utilities	Facility Management	E-Energy	Logistics
Machine-2-Machine	Push/Pull content		x	x	x	x	x	x	x
	Control devices			x	x	x	x	x	x
	Subscription/notification			x	x		x		x
	Discovery	x		x			x		x
	Location			x	x				x
Human-2-Human	A/V Call	x			x	x	x		
	A/V conference	x			x	x	x		
	Messaging / File transfer	x	x		x	x	x		
	Presence	x					x		x
	Location	x	x		x	x	x	x	x
	Address Book	x					x		
Overarching enablers	QoS	x			x	x	x		
	Device/entity mgmt	x	x		x	x	x	x	x
	Security	x	x		x	x	x	x	x

4.2 Machine-to-Machine (M2M) communication

Refers to the paradigm of communication that enables machines to communicate autonomously with each other with limited or no human interaction. It leverage the independent communication between large number of objects, devices and service platforms, which in turn presumes the transmission of large amount of data of heterogeneous types and sizes over the network. Service enablers for M2M should be designed with keeping in mind the fact that devices are not under direct human control, while the current networks are designed to support H2H communication.

4.3 Overarching Services

This category include all types of management and control services that are required in association with H2H and M2M communications, such as security and Quality of Service (QoS) controlling.

Table 1 shows an analysis of required service enablers from each category in building some application domains. The main aim of this work is to contribute to the definition of service enablers enabling the combination of seamless H2H and M2M services, and to propose a set of APIs. Smart City platforms should be able to support various service verticals and provide required enablers to implement value added services for different application scenarios. The application domains selected in our work reflects the key affected industries by the IoT paradigm. Figure 1, illustrated that intelligent building and utilities will be the most significant industry verticals for the IoT by 2022. As illustrated in Figure 2, current Future Internet architecture includes multiple telecom horizontal middleware platforms, beside Over-The-Top (OTT) networks. A Generic framework for federation between these heterogeneous platforms is one of the challenges. We propose three services enablers for H2H, M2M and Overarching services forming an Open Communication Server (OCS), that act as a convergence and orchestration point for networks, services and data, supporting Smart Cities framework. OSC is supposed to break the gap between classical telecommunication services and Internet services. For achieving this aim the platform enables easy service integration and the composition of those services. This approach allows providers and even 3rd party developers to create services that can be exposed via several standard interfaces.

Generally, the creation of standards APIs are crucial to the development of NGN platforms, and with consideration of the huge heterogeneity nature of networks and devices in the M2M platform, the need to standard interfaces to M2M platforms is more clear. Making the M2M platform transparent to the application layer will encourage developers to build applications for these platforms without the need to learn deep details of the operator middleware organization.

5. Complete Development Environment

Developing new services is becoming a complex process involving various programming languages, frameworks, and paradigms. Furthermore, billions of nun-human objects will be

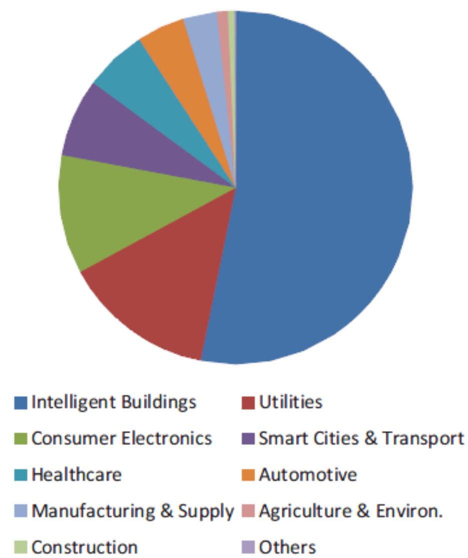

Figure 1. IoT opportunity by Sectors, 2022 [24]**.**

connected to the Internet [7], recording everyday activities and interact with each other in an seamless integrated communication scenarios, enhancing the traditional way of providing communication service in several domains, including:

- Transportation and logistics domain.
- Healthcare domain.
- Smart environment (home, office, plant) domain.
- Personal and social domain.

Software development tools combined with availability of next generation networks will create new opportunities for the creation of the desired services. There has been no doubt that M2M communication plays an important role in realizing the next-generation Smart City system. Possessing a standard based service creation platform facilitates the creation of innovative services. In this paper, we highlight the requirements and challenges of developing a unified SDK to support 3rd party developers in code, build, and debug their applications in a single place. The main characteristics expected to be presented in a feature rich SDK are:

- Emulator of target devices, network and target build environment for allowing applications testing.
- User-friendly GUI to code, build and test.
- Support for multiple host operating systems.
- Full documentation of integrated API, including example code and build instructions.

In the vision of the IoT, M2M applications are not only responsible to monitoring the status of remote equipment, but also responsible to gather real-time data from millions of heterogeneous connected machines and translate it into meaningful information, that shall support quick decisions, automated actions, and strategic analytics. While these capabilities can increase the value of M2M applications for the enterprise, they can also increase complexity for developers. Taking into account the variety of technologies and protocols integrated in the system, this may lead to long and costly development cycles. Generally, the requirements of deploying M2M applications in large-scale systems are:

- Supporting services deployed and hosted by 3rd party developer.
- Delivering services rapidly to the market.
- Co-existing and collaborating with current core networks and new deployment.
- Delivering and handling different kind of content and supporting variant bit rates.
- Being compliant with standards based services.

Based on the requirements previously described, we developed the M2M enablers and integrated it to our SDK for OpenMTC platform. The main concerns in the approach of developing M2M APIs was: the compatibility with main Standards in the M2M area, such as ETSI and OMA NGSI [22]; and independency from underlying technologies. Although the adaptation of APIs in developing Web services increases the portability, it does not ensure the interoperability. We need standards to unify the communication mechanisms between vertical services.

OpenMTC supports a client/server based RESTful architecture with a hierarchical resource tree defined by ETSI. This style governs how M2M applications exchanging data with each other, and the platform. Adopting the RESTful style facilitates the development of M2M applications, due to its simplicity in comparison with most SOA technologies. Figure 3 depicts an overview of the M2M resources tree for M2M enablers in line

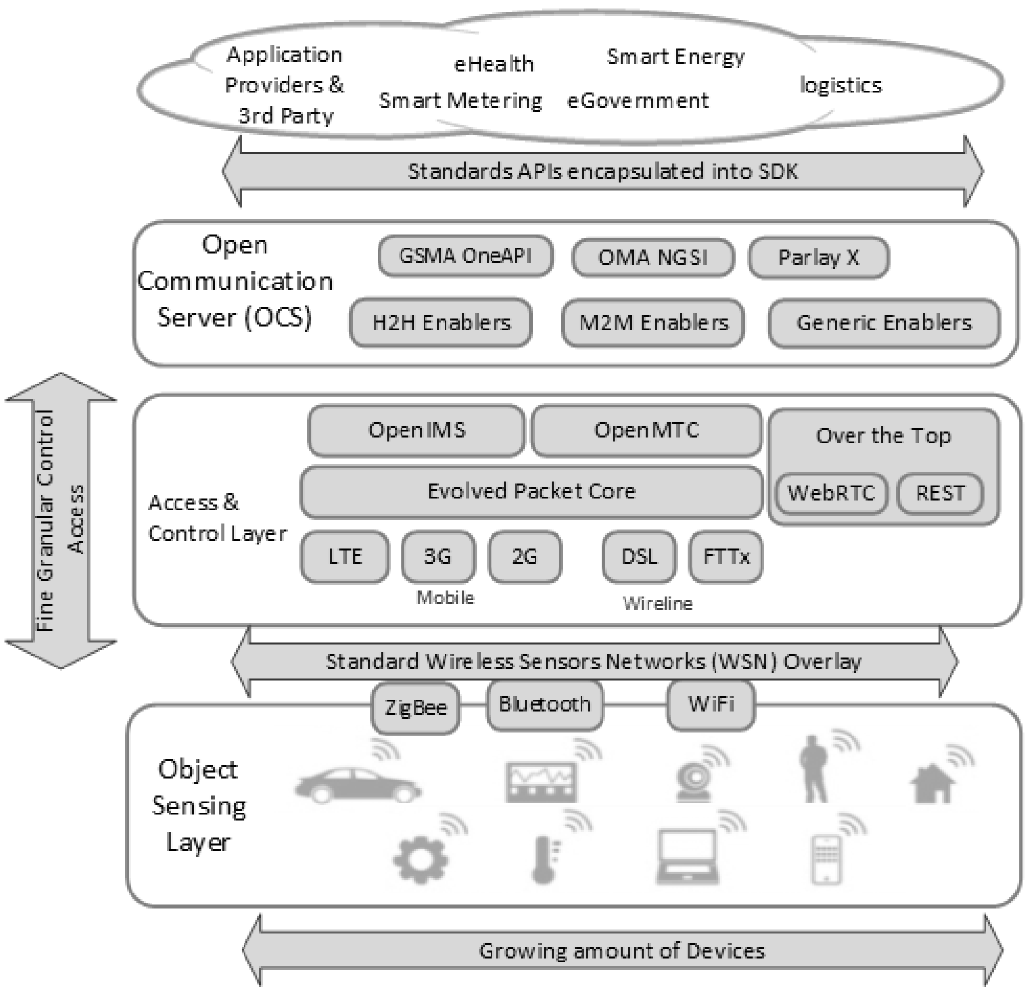

Figure 2. Smart City layer Architecture

with ETSI and OMA NGSI standard operations.

The Future Internet Core Platform (FI-WARE) project [23] defines a reference architecture for the Future Internet, that consists of a set of Generic Enablers (GE) offer reusable and commonly shared functions serving multiple areas of use across various sectors. The FI-WARE architecture can be instantiated into a concrete architecture by means of selecting and integrating products implementing the corresponding FI-WARE GEs. The project specified standard based APIs and interoperable Protocols supported by GE. The OpenMTC is included in the Fi-ware Internet of Things (IoT) Services Enablement chapter as a device management GE for IoT.

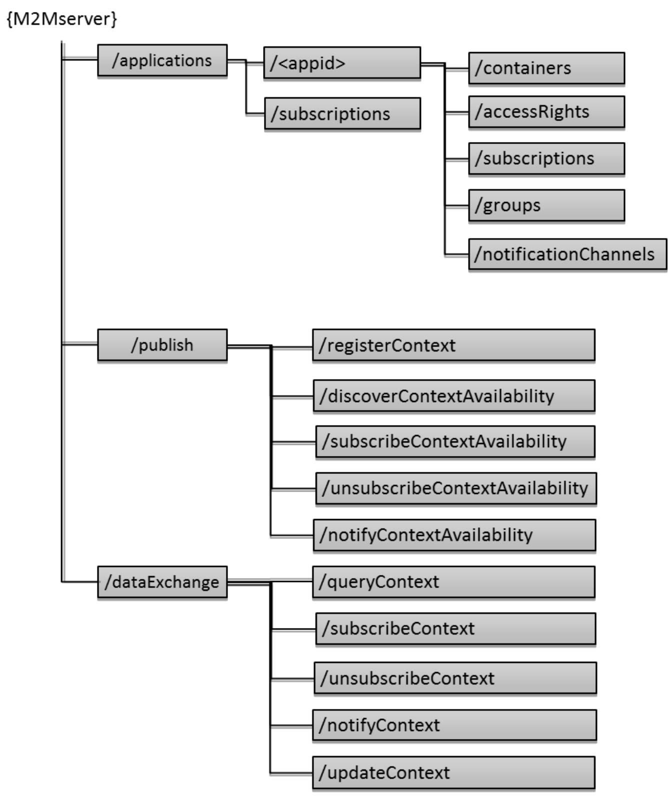

Figure 3. M2M Enabler resource tree.

6. CONCLUSION

Although significant work has been done recently in the standardisation of open APIs for telecom services, however, application developers are still creating their software based on tools provided by equipment vendor and network operators, which are mostly not compatible to related standards. This restricts developers to write applications for a wide user base. The challenge is to develop a methodology for the interaction between remote sensors and the user application that hides the complexity of the communication layer and carries out tasks automatically without the user's intervention. Additionally it is necessary to introduce new innovative applications that improve the Quality of Life (QoL), and increase awareness of the surrounded environment. Building and marketing apps for smartphones has become an essential part of Smartphone manufactures business, and it is expiated that an IoT ecosystem will be established in the same manner for M2M/IoT platforms. However, the IoT environment includes heterogeneity of protocols in both device and application domain. Therefore it is important to adapt unified standard interfaces in order to insure interoperability, and thus enable the development of Smart cities.

This paper highlights the work in designing Service enablers for M2M applications. Using APIs support the process of build systems from semi-independent components in a scalable way. Our SDK provides a unified API to M2M client applications while hiding heterogeneity of end-customer premises equipment (i.e. Smart meter, and sensors), and the communication links between customer premises and M2M service center.

7. REFERENCES

[1] Henry William Chesbrough, *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Harvard Business School Press, 2003.

[2] A. Manzalini, R. Minerva, and C. Moiso, "If the Web is the platform, then what is the SDP?," *2009 13th Int. Conf. Intell. Next Gener. Networks*, pp. 1–6, Oct. 2009.

[3] "9,000 APIs: Mobile Gets Serious." [Online]. Available: http://blog.programmableweb.com/2013/04/30/9000-apis-mobile-gets-serious/.

[4] I. Y.2060, "Overview of the Internet of things." 2012.

[5] C. de Souza, D. Redmiles, L. C. David, J. Patterson, and D. Millen, "Sometimes You Need to See Through Walls — A Field Study of Application Programming Interfaces," in *the 2004 ACM conference on Computer supported cooperative work*, 2004, pp. 63–71.

[6] N. D. Tselikas, G. S. Tselikis, and N. 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*, 2010, vol. 33–37, pp. 33–37.

[7] Y. Chen, "Challenges and Opportunities of Internet of Things," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2012, pp. 383–388.

[8] H. Chourabi, T. Nam, S. Walker, J. R. Gil-Garcia, S. Mellouli, K. Nahon, T. a. Pardo, and H. J. Scholl, "Understanding Smart Cities: An Integrative Framework," in *2012 45th Hawaii International Conference on System Sciences*, 2012, pp. 2289–2297.

[9] C. Harrison, B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak, and P. Williams, "Foundations for Smarter Cities," *IBM J. Res. Dev.*, vol. 54, no. 4, pp. 1–16, Jul. 2010.

[10] "NIST Guide to Supervisory and Data Acquisition-SCADA and Industrial Control Systems Security (2007).pdf (application/pdf Object)." [Online]. Available: http://www.cyber.st.dhs.gov/docs/NIST Guide to

Supervisory and Data Acquisition-SCADA and Industrial Control Systems Security (2007).pdf. [Accessed: 28-Sep-2012].

[11] I. M. Moreno-Vozmediano, Rafael; Montero, Ruben S.; Llorente, “Key Challenges in Cloud Computing,” *IEEE Internet Comput.*, vol. 17, no. 4, pp. 18–25, 2013.

[12] “Alertme.” [Online]. Available: http://alertme.com.

[13] “Xively.” [Online]. Available: https://xively.com/.

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

[15] M. Corici, H. Coskun, A. Elmangoush, A. Kurniawan, T. Mao, T. Magedanz, and S. 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*, 2012, pp. 1860–1865.

[16] G. Branca and L. Atzori, “A Survey of SOA Technologies in NGN Network Architectures,” *IEEE Commun. Surv. TUTORIALS*, vol. 14, no. 3, pp. 644–661, 2012.

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

[18] “Java Community Process.” [Online]. Available: http://www.jcp.org.

[19] “GSMA OneAPI.” [Online]. Available: http://oneapi.gsma.com/.

[20] ETSI TS 102.690, ““Machine-to-Machine communications (M2M); Functional architecture.” 2011.

[21] “Open Mobile Alliance (OMA).” [Online]. Available: http://openmobilealliance.org/.

[22] 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.

[23] “Fi-Ware Project.” [Online]. Available: http://www.fi-ware.eu/.

[24] J. Morrish, “The Emergence of M2M / IoT Application Platforms,” no. September. Machina Research, 2013.