Designing IoT Architecture(s)

A European Perspective

Srdjan Krčo, Boris Pokrić
Ericsson
Belgrade, Serbia

Francois Carrez
CCSR
University of Surrey
Guildford, UK

Abstract— Internet of Things (IoT) domain has attracted a lot of interest over the last few years, to a large extent due to its applicability across a plethora of application domains. This variety of application domains resulted in a variety of requirements that IoT systems should comply with. Due to the heterogeneity of the domains, the requirements varied significantly, and demanding more or less complex systems with varied performance expectations. This situation affected the architecture design and resulted in a range of IoT architectures with not only varied set of components and functionalities, but also varied terminologies used. It resulted in limited interoperability between the systems which in turn hampered development of the complete domain.
To address these issues, to ensure a common understanding by providing a framework catering for different applications and eventually enable reuse of the existing work across the domains, reference architectures are an appropriate tool. This paper presents an overview of the activities done in Europe towards definition of such a common framework together with how it is being used and a potential outlook for these efforts.

Keywords—Internet of Things, IoT, architecture

I. INTRODUCTION

Over the years, a number of Internet of Things (IoT) related projects have specified their own versions of architectures, basing them on the specific requirements the projects were addressing (IoT-A [4], SENSEI [5], SPITFIRE [7], etc.). Depending on the scope of the project or the problem domain being addressed, architectures were focusing on different aspects or a sub-domain of IoT. Some projects, like SENSEI, were more concerned about the service layer and focused on wireless sensor and actuator networks only, ASPIRE [3] was dealing mainly with the RFID domain, while semantic aspects were addressed for example by the SPITFIRE project. Due to a large heterogeneity of application domains and consequently the requirements, the approaches to the architecture specification differed between the projects thus resulting in more or less different architectures, comprised of a number of components and protocols. This resulted in limited interoperability between the systems which combined with often different terminology used also made discussions between the domains difficult. This situation could be compared to a plethora of remote controllers that we have on our tables and use to control TV, DVD, audio devices etc. All of them look similar, have similar functions, but is often difficult to find the right one and even more to make one of them control all devices.

The initial attempt towards coordination of the efforts done in the context of various projects funded under the FP7 programme, were done in 2011 in the context of the FIA events and the Real-world working group. Leveraging inputs from several contributing projects, an architectural blueprint for a real-world Internet (at that moment, the term IoT was primarily associated with RFID, hence the real-world Internet term was used to highlight that it is about all smart things, not just RFID) is described in [1].
Since the time this paper was published, a number of new IoT related projects, co-funded in the FP7 programme, were initiated. The trend of designing new architectures over and over again has continued. This was recognized as one of the barriers for a faster development of the domain and was a reason for initiation of two projects: IoT-I [9] and IoT-A. The former one dealt with the analysis of different architectures and engaging the community to better understand the needs and requirements of researchers and industry in regard to the IoT architecture. A survey run by the project indicated that the opinion of the IoT community was that a common IoT reference model is required, although 25% of the survey participants did not believe that it was possible to define one. According to the survey, the main purposes of a common model are to enable interoperability between the solutions, promote common understanding of IoT and serve as a basis for development of IoT reference architecture. Among the most important components of an IoT reference model were terminology definition, interface definitions, interaction model, standards, communication model, and security and information models. These activities supported execution of the IoT-A project, a large-scale project focused on design of a comprehensive framework that would facilitate common approach to design of IoT architecture(s).
In parallel with these initiatives and backed by a large industrial support, the FI-WARE project [10] started to work on design of a core platform for the next generation Internet and as a part of that effort produced architecture for the IoT domain of the so called Future Internet.
In addition to these European Commission backed efforts, a global initiative under the ETSI auspices was initiated aiming mainly to define service layer of the Machine to machine (M2M) systems. Over the last two years this initiative evolved into a truly global undertaking (oneM2M) aiming to do for M2M what 3GPP has done for mobile networks.

978-1-4799-3459-1/14/$31.00 ©2014 IEEE

The rest of the paper is organized as follows. The following sections provide an overview of the activities undertaken in the ETSI M2M, FI-WARE and IoT-A projects. These sections are followed by a section describing architecture of the IoT6 project which relied on the IoT-A ARM and is one the first projects that adopted the ARM and used it to design own architecture. The final section provides concluding remarks.

II. ETSI M2M AND ONEM2M

In January 2009, ETSI M2M technical committee was established with the aim to develop and maintain an end-to-end high level architecture for M2M. The final release has been created in 2013 [11] and the corresponding architecture is shown in Figure 1.

The architecture consists of two distinct domains: the *device and gateway* domain and the *network* domain. IoT/M2M gateways enable communication of M2M devices with other parts of the system via access networks, i.e. wide area network. There can be one or more M2M devices connected to a M2M gateway. In principle, M2M devices connected via gateway do not implement the so called M2M applications and M2M service capability functionality. However, if a M2M device implements the M2M applications and M2M service capability functionality, then this device can be connected directly to the access network and interact with the rest of the system. The network domain consist of the wide area communication networks (access and core networks), M2M service capabilities and M2M applications functions. M2M management and network management functions are also components of the network domain.

In 2012, intensive efforts on synchronization of M2M standardization activities were undertaken, resulting in of the oneM2M Global Initiative [12]. The overall objective of oneM2M is to develop globally agreed technical specifications which address a common M2M Service Layer that can be readily embedded within various hardware and software, and relied upon to connect the myriad of devices in the field, promoting interoperability across vertical industries and networks. The following technical specifications and reports are in the initial scope of this organization:

- Use cases and requirements for a common set of Service Layer capabilities;
- Service Layer aspects with high level and detailed service architecture, in light of an access independent view of end-to-end services;
- Protocols/APIs/standard objects based on this architecture (open interfaces & protocols);
- Security and privacy aspects (authentication, encryption, integrity verification);
- Reachability and discovery of applications;
- Interoperability, including test and conformance specifications;
- Collection of data for charging records (to be used for billing and statistical purposes);
- Identification and naming of devices and applications;
- Information models and data management (including store and subscribe/notify functionality);
- Management aspects (including remote management of entities); and
- Common use cases, terminal/module aspects, including Service Layer interfaces/APIs between:
  - Application and Service Layers;
  - Service Layer and communication functions.

III. FI-WARE

The overall vision of FI-WARE is to build Core Platform of the Future Internet.

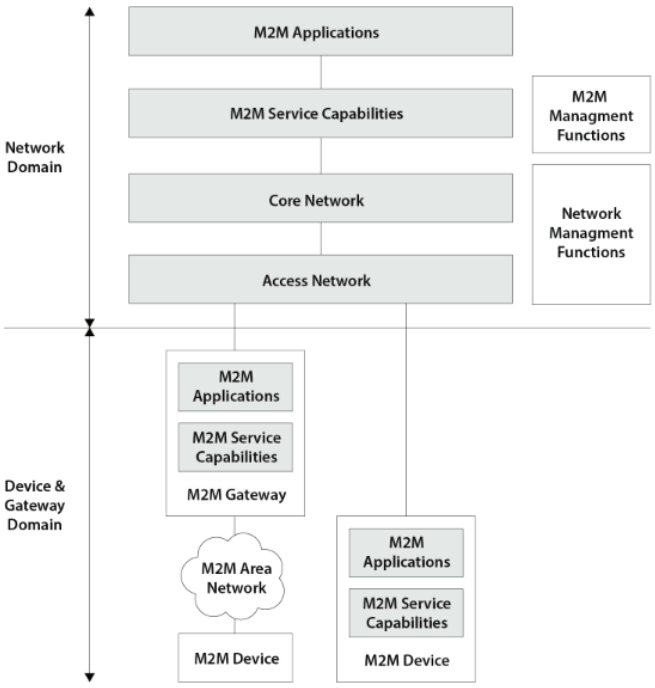

Figure 1: ETSI M2M top-level architecture

This platform will be open, based upon components referred to as Generic Enablers (GE) which offer reusable and commonly shared functions serving a multiplicity of Usage Areas across various sectors [10].

The generic enablers are classified into six major groups providing architecture reference model for the specific features addressed within the chapters:

- Cloud Hosting – computation, storage and network resources, upon which services are provisioned and managed.
- Data/Context Management – accessing, processing, and analysing massive volume of data, transforming them into valuable knowledge available to applications.
- Applications/Services Ecosystem and Delivery Framework – the infrastructure to create, publish, manage and consume FI services across their life cycle, addressing all technical and business aspects.
- Internet of Things (IoT) Services Enablement – the bridge whereby FI services interface and leverage the ubiquity of heterogeneous, resource-constrained devices in the Internet of Things.

- Interface to Networks and Devices (I2ND) – open interfaces to networks and devices, providing the connectivity needs of services delivered across the platform.
- Security – the mechanisms which ensure that the delivery and usage of services is trustworthy and meets security and privacy requirements.

Most relevant chapter in the context of this paper is the IoT chapter. Figure 2 shows the IoT architecture as defined by the FI-WARE project. This architecture has already taken into account the ETSI M2M specification and has extended it to incorporate OMA NGSI activities [13], [14]. The following large functional blocks can be identified in this architecture: backend, gateway, IoT devices and legacy devices. The deployment of the architecture of the IoT Service Enablement chapter is typically distributed across a large number of Devices, several Gateways and the Backend. The Generic Enablers shown in Figure 2 implement the functionalities distributed across these elements.

The BackEnd functional block acts as the main component providing the functionality to access the IoT devices both in the terms of the information they produce and in the way to control them. This component provides both REST and NGSI interfaces for interaction with the users as well as appropriate features such as things and resources management and publish/subscribe functionality and connectivity management. BackEnd consists of three main GEs, namely IoT Broker, Configuration Management and Backend Device Management. The IoT Broker GE is responsible for for retrieving and aggregating information from the devices. The Configuration Manager GE (ConfMan GE for short) is responsible for context availability registration. The Backend Device Management GE is the central component which provides the resource-level management of remote assets (devices with sensors and/or actuators) as well as core communication capabilities such as basic IP connectivity and management of disconnected devices.

The gateway provides similar functionality, but on the local level, i.e. it provides functions like things and resource management for the IoT and legacy devices connected to the gateway. It consists of three GEs, namely Data Handling, Gateway Device Management and Protocol Adapter. The Gateway Device Management GE is responsible for the communication with the Backend and IoT and non-IoT devices. The Gateway Device Management GE includes the functional components to handle the registration/connection phases towards the Backend/Platform, to translate the incoming data or messages in an internal format and to send the outgoing data or messages in the ETSI M2M format (marshal/unmarshal). The Data Handling GE addresses the need for filtering, aggregating and merging real-time data from different sources. The Protocol Adapter GE deals with the incoming and outgoing traffic and messages between the Gateway and Devices registered, to be served by the gateway towards the Gateway Device Management GE or the Data Handling GE. The Protocol Adapter GE translates device specific protocols into a uniform internal API.

IoT devices can be connected to a gateway (e.g. IPv4-based devices with private addresses) or directly to the backend (e.g. IPv6-based devices with public addresses). Legacy devices are always connected via a gateway.

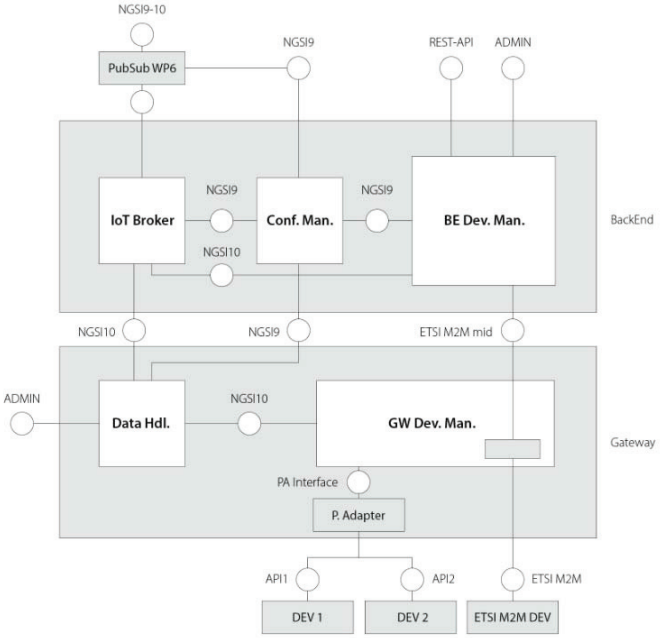

Figure 2: FI-WARE IoT architecture

IV. IoT-A

The main objective of the IoT-A project is to provide as generic as possible Architectural Reference Model (ARM) that can be used to derive concrete IoT architectures. In other words, IoT-A is not focused on defining THE architecture for the IoT, but on the contrary, on providing a number of means (models, views, perspectives, best practices, etc.) that can be used to derive an IoT architecture. In this context, two different architects focusing on two specific IoT applications would use and share the same Reference Architecture as a tool, but would eventually come up with different architectures at the end of their architecting process, but not "any architecture" as we explain in the following.

The motto of IoT-A is to build "Internet of things" not "*intra*-net of things", i.e. to offer IoT architects a common technical grounding in order to optimize interoperability. In that case, IoT applications would not be any longer built as stand-alone silo applications, but as inter-operable vertical applications still having a common "horizontal" grounding – the ARM (compliant components, protocol suites, etc.).

The ARM consists of three interconnected parts and it also takes from based on the best practices in software engineering as introduced by Rozanski & Woods [19]. Those three parts are:

- The IoT Reference Model (RM);
- The IoT Reference Architecture (RA);
- A set of Guidance (also called best practice).

The RM provides a set of models that are used to define certain aspects of the architectural views. One of the most important models is IoT Domain Model [20]. It defines taxonomy of IoT concepts (e.g. Physical, Virtual and Augmented Entities, Devices, Resources and Services) and a set of relationships between those concepts. It defines the IoT domain in general, a customization of this generic model w.r.t. a specific IoT application allows to generate a common understanding of that domain (like identifying the entities of interest for that application, identifying the resources, e.g. sensors, actuators etc.). The RM also provides: 1/an Information Model (IM) which is a meta-model used to describe information as handled within the system, 2/ a Communication Model, 3/ a Functional Model (FM) used as the foundation of the Functional View and finally 4/models for Security, Trust and Privacy.

Based on the RM models, the RA consists of a set of Views (used to represent certain structural aspects of the system) and Perspectives (that focus on quality of the system that spans different views, e.g. Security, Resilience).

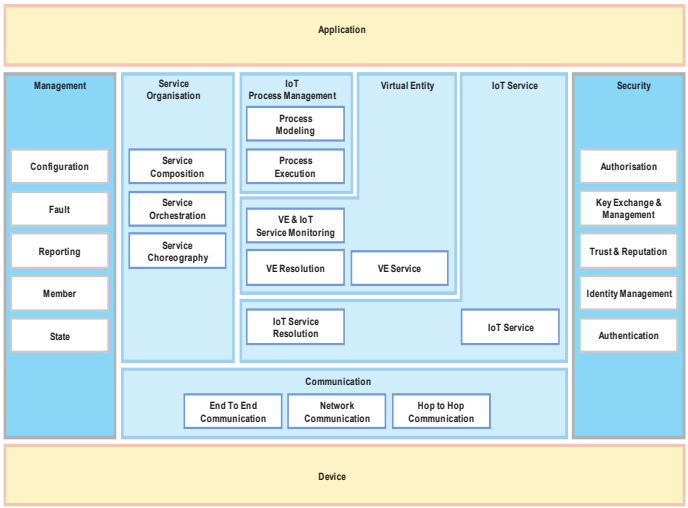

Figure 3 IoT-A ARM functional view

The *Functional View* (Figure 3) proposes a layered model of Functional Groups which maps to most of the concepts introduced in the DM, together with a set of essential Functional Components (and associated interfaces) that an IoT system should provide. It is worth mentioning that the FV is not exhaustively developed (see the conclusion section). The Information View, based on the IM, complements the FV and provides a more detailed view about how information is to be handled in the system (including details about the components where the information is handled) and how it flows within the system. The perspectives are mainly derived from non-functional requirements and consist of activities and related tactics.

Last, but not the least there is the Guidance part. It defines the process that based on the RA and RM will lead to the generation of the concrete IoT architectures. In particular it defines the requirement process, introduces additional views (i.e. Physical View and Context View) that are not part of the ARM as they are extremely application dependent and explain in general how and in which order the set of architectural views (which constitute a concrete architecture according to Rozanski and Woods) should be generated. It also gives a large (but not exhaustive) list of design choices that can be used as recommendations to achieve certain system qualities (see perspectives above).

V. IoT6

From the very beginning of the project, the approach to IoT6 architecture design was to reuse to the furthest extent possible the outcomes of other projects, most notably IoT-A, ETSI M2M and FI-WARE, and to adapt them and enhance with IoT6 specific features and components, mainly coming from leveraging various IPv6 functionality. The aim is to utilise properties of this protocol and re-use them within the architecture model, possibly replacing some of the standard components. For example, parts of the service and resource discovery functionality are replaced with the DNS-SD [15] and mDNS [16] based approaches. Looking at the ARM functional model (Fig. 3), the main focus of IoT6 contribution is on the Communication, Service organization, IoT service and Security components.

The project followed architecture design methodology as outlined by the ARM (Figure 4). Based on the analyzed scenarios and the derived requirements, with the support of the ARM and the associated best practices, and influenced by a set of design choices, the IoT6 architecture was designed. On the device level, IoT6 defines two big groups: devices supporting IPv6 and legacy devices that do not support it. IPv6 based devices can be organized in small or large clusters. Legacy devices can support a range of protocols, like KNX, ZigBee, Bluetooth as well as IPv4. An additional cluster is dedicated to EPCglobal compliant RFID system. On the communication level, IoT6 utilizes IPv6. Devices are connected via the so-called half gateways (convert legacy protocols to IPv6) or directly in the case of IPv6 devices. This setup can be directly mapped to the ARM communication channel model [5]; ARM's constrained networks are mapped to one or the other group of devices as defined above, while IoT6's half-gateways represent ARM's gateways.

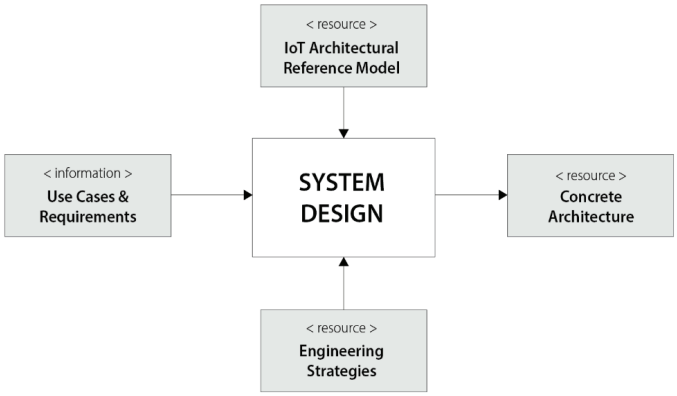

Figure 4 Architecture design methodology

On top of the IPv6, CoAP is selected as the preferred protocol with different encoding techniques (JSON, XML). For a specific case of building automation, oBix protocol is included. On the IoT service level, several solutions are

supported. In the case of small IPv6 clusters, mDNS is used for service registration and discovery (inside the cluster). In the case of large clusters, DNS-SD is used for internal cluster service registration and discovery. For the EPCIS cluster, an adaptation of the Digcovery solution is envisioned. On the global level, two solutions are supported: Digcovery [17] and CoAP RD [18]. When it comes to the service organization level, the project relies on the cloud based workflow and process management services which interact with the rest of the system using CoAP.

Following the ARM approach, the project was able to streamline architecture design and basing it on a common terminology make it easier for other researchers to compare, reuse and expand.

VI. CONCLUSIONS

Definition of the scenarios and use cases, followed by technical requirements identification and then architecture design is the usual approach towards organization of projects, including those run under the EU co-funded FP7 programme. In the case of projects in the IoT domain, mainly due to a large range of potential application domains and the corresponding requirements, this resulted in a range of architectures, with varied similarity, protocols, interfaces and functionalities. This was identified as one of the stumbling blocks for rapid development of the IoT technology and even more for rapid adoption of solutions across the application domains as it was hard to replicate solutions in different usage areas or to reuse components between the solutions.

The European Commission supported several initiatives in order to deal with this issue. IoT-i dealt with the community engagement and initiated establishment of the International IoT Forum aiming among other things to facilitate discussions on the IoT architecture and to help in creating a common IoT architecture framework. IoT-A project has done a great job by systemizing the IoT architecture design and providing IoT architecture reference model together with a set of best practices to help system designers with the design of concrete IoT systems. This was also noted by the FI-WARE project and resulted in a series of meetings aiming at synchronizing the approaches and the terminology, thus facilitating a common understanding and a more rapid development of the domain going forward.

These discussions were particularly supported by the IoT European Research Cluster which also served as a venue for coordination and collaboration of FP7 IoT projects.

Last but, not the least, the International IoT Forum, officially founded as a separate entity during the IoT week 2013 in Helsinki has pledged to continue to support the IoT-A ARM and further extend it and enhance it after the end of the project (November 2013). The Architecture working group of the IoT Forum will actively work on this topic, particularly aiming to improve and complement several aspects of the ARM (e.g. RM improvement, Views definition, expending the list of design choices) as well as to define appropriate ARM profiles in greater details (e.g. security and semantic interoperability profiles). The ultimate objective is boosting the ARM usability and adoption within the IoT community. To facilitate this, the IoT Forum will define an ARM-compliant "label" and provide procedures for certification and maintenance of an eco-system of reusable ARM-compliant certified components.

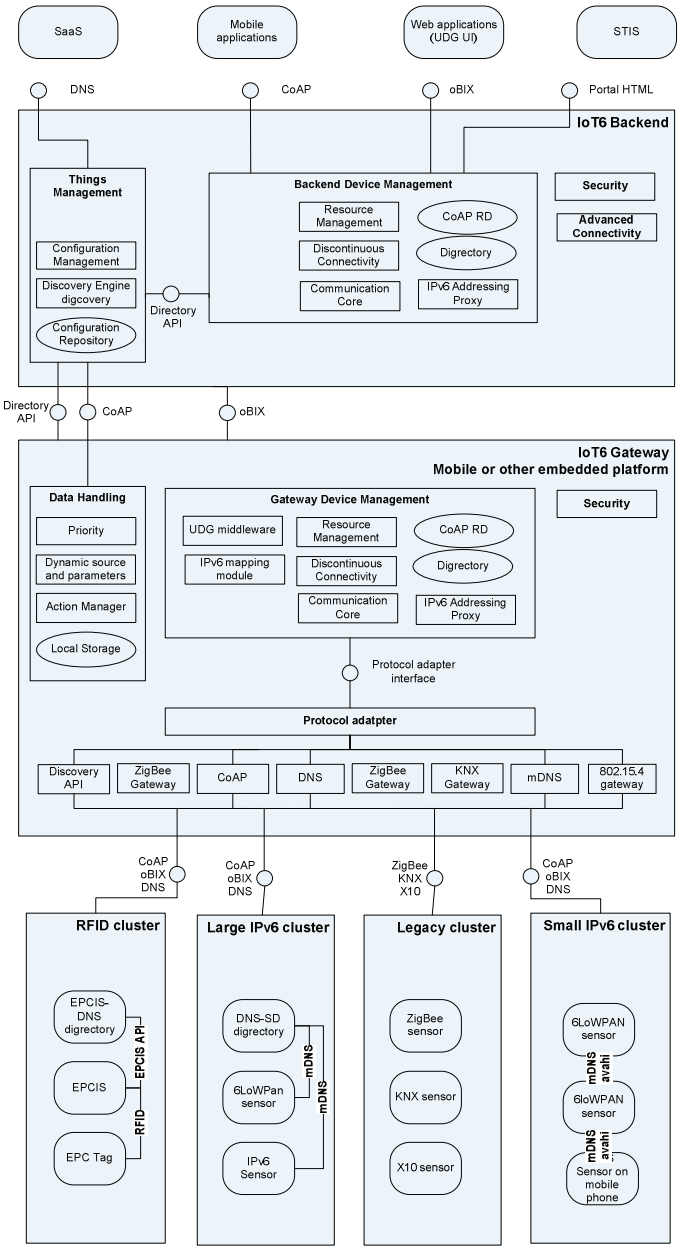

Figure 5 IoT6 architecture

The aforementioned activities will be of significant benefit to the new wave of the FP7 IoT projects which are increasingly adopting the IoT-A ARM as the starting point of their architecture design activities. This should greatly foster the alignment of the architectures and enable simpler reuse of the results, functionalities and components.

ACKNOWLEDGMENT

The work on this paper was done in the context of IoT6 (http://www.iot6.eu) and IoT-A (http://www.iot-a.eu) projects that received funding from the EC's 7th Framework Programme, grant agreements n°288445 and n°257521.

REFERENCES

[1] A. Gluhak et al., "An architectural blueprint for a Real-World Internet", The Future Internet, Lecture Notes in Computer Science Volume 6656, 2011, pp 67-80

[2] ETSI TS 102 690, Machine-to-Machine communications (M2M), Functional architecture, 2011

[3] Advanced Sensors and lightweight Programmable middleware for Innovative RFID Enterprise applications, FP7, http://www.fp7-aspire.eu/, last accessed 08/10/2013

[4] EU FP7 Internet of Things Architecture project, http://www.iot-a.eu/public, last accessed 08/10/2013

[5] F. Carrez *et al.*, "IoT-A Deliverable D1.5 – Final Architectural Reference Model for the IoT v3.0", www.iot-a.eu/public/public-documents/

[6] Integrating the Physical with the Digital World of the Network of the Future, FP7, http://www.ict-sensei.org, last accessed 08/10/2013

[7] Semantic-Service Provisioning for the Internet of Things using Future Internet Research by Experimentation, FP7, http://spitfire-project.eu/, last accessed 08/10/2013

[8] Zorzi, M., Gluhak, A., Lange, S., Bassi, A.: From Today's INTRAnet of Things to a Future INTERnet of Things: A Wireless- and Mobility-Related View. IEEE Wireless Communications 17(6) (2010)

[9] S. Haler et al., IoT-i deliverable D1.5: IoT Reference Model White Paper, September 2012

[10] FI-PPP FI-WARE project, http://www.fi-ware.eu, last accessed 08/10/2013

[11] ETSI M2M latest technical specifications, http://docbox.etsi.org/M2M/Open/Latest_Drafts/

[12] OneM2M global initiative, http://www.onem2m.org/

[13] OMA Next Generation Services Interface V1.0, http://technical.openmobilealliance.org/Technical/release_program/ngsi_v1_0.aspx

[14] NGSI-9/NGSI-10 information model, http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/NGSI-9/NGSI-10_information_model

[15] DNS Service Discovery (DNS-SD), http://www.dns-sd.org/

[16] Multicast DNS, http://www.multicastdns.org/

[17] Antonio J. Jara et. al., "Mobile digcovery: discovering and interacting with the world through the Internet of things", Personal and Ubiquitous Computing, March 2013, DOI: 10.1007/s00779-013-0648-0

[18] CoRE Resource Directory IETF specification, http://tools.ietf.org/html/draft-ietf-core-resource-directory-00

[19] N. Rozanski and E. Woods, "Applying Viewpoints and Views to Software Architecture" , Addison Wesley, 2011

[20] S. Haller *et al*., "A Domain Model for the Internet of Things", in iTHINGS'2013 proceeding, Beijing, China (see also IEEE eXplore)