Internet-of-Things Architecture

IOT-A

Project Deliverable D3.1 - Initial M2M API Analysis

Project Acronym:	IOT-A
Project Full Title:	Internet-of-Things Architecture
Grant Agreement no.:	257521

Doc. Ref.:	D3.1
Responsible Beneficiary:	CSE
Editors:	Spyridon Tompros (CSE)
List of Contributors:	Dimitris Tsaimos (CSE), Norbert Vicari (Siemens), Werner Liekens (ALU BE), Alexis Olivereau (CEA), Andreas Nettsträter (FHG IML), Michele Rossi (CFR), Pierpaolo Giacomin (HEU)
Reviewers:	Alexandru Serbanati (CATTID)

	Dissemination Level	PU
PU	Public	
~~PP~~	~~Restricted to other programme participants (including the Commission Services)~~	
~~RE~~	~~Restricted to a group specified by the Consortium (including the Commission Services)~~	
~~CO~~	~~Confidential, only for members of the Consortium (including the Commission Services)~~	

Executive Summary

Machine-to-Machine communication is an essential part of the IoT concept, due to the fact that IoT endpoints can be viewed as machines that are able to communicate with remote machines. The actual M2M API exposed by endpoints to indicate their capabilities and services, may serve as an abstraction layer that provides a uniform view of IoT endpoints. Towards this end, the objective of this deliverable is to provide an initial analysis on the M2M API that must be designed for the IoT-A project. Such an analysis is necessary one the one hand to complement the work of WP1, because M2M communication is not visible at architectural point of view. On the other hand, M2M API requirements need to be taken into account - along with requirements originating from other IoT-A WPs - in the design of the protocol stack.

The organization of the deliverable is as follows. Initially, the overall state of the art on M2M communications is presented, identifying the main standardization efforts and international research activities. Furthermore, an overview of currently available protocols and software platforms for realizing M2M communication is provided. Thereafter, target application areas of M2M communication are presented, enlisting commercial products per target application area, as well as prototype implementations. Finally, the document presents the methodology followed to extract the M2M communication requirements. The result of the aforementioned process is an enumeration of M2M communication requirements for the IoT-A project and the effect they have upon the M2M API.

Contents

Contents 1

1 Introduction 6

2 State of the Art on M2M Communication 7
2.1 Standardization Activities . . . 7
2.1.1 ETSI M2M Technical Comittee . . . 7
2.1.2 3GPP TSG Service and System Aspects . . . 8
2.1.3 TIA TR-50 Engineering Committee . . . 8
2.1.4 CCSA TC10 . . . 9
2.1.5 Global ICT Standardization Forum for India . . . 9
2.1.6 International Telecommunication Union . . . 10
2.1.7 Open Mobile Alliance . . . 10
2.1.8 M2M Standardization Task Force . . . 11
2.1.9 Summary . . . 12
2.2 International Research Activities . . . 12
2.2.1 European research projects . . . 12
2.2.2 US research projects . . . 16
2.2.3 Summary . . . 20
2.3 M2M Protocols . . . 20
2.3.1 SCADA . . . 20
2.3.2 UPNP . . . 22
2.3.3 NAT Portmapping Protocol . . . 24
2.3.4 DPWS . . . 24
2.3.5 XML protocols . . . 24
2.3.6 COAP . . . 25
2.3.7 Internet protocols . . . 25
2.3.8 Summary . . . 29
2.4 Software Platforms . . . 29
2.4.1 Open Source Platforms . . . 29
2.4.2 Commercial Software Development Platforms and Operating Environments . . . 34
2.4.3 Summary . . . 35

3 Applications 35
3.1 Commercial Products . . . 36
3.1.1 Home Automation / Smart Home / Commercial Building Automation . . . 36
3.1.2 Medical Applications . . . 39
3.1.3 Logistics Applications . . . 39
3.1.4 Defense Applications . . . 40
3.1.5 Environmental Applications . . . 41
3.1.6 Analysis . . . 41
3.2 M2M Prototypes . . . 41

3.2.1 InterDigital M2M prototype . . . 42
3.2.2 Orange Labs Digital Home . . . 42
3.2.3 EDF R&D Smart Grid Experimental Platform . . . 43
3.2.4 Wuxi (China) Institute of things . . . 43
3.2.5 Trangram . . . 43
3.2.6 HP CeNSE . . . 44
3.2.7 Critical Software EMMON . . . 44
3.2.8 Home Sensor Gateway prototype (Alcatel-Lucent) . . . 46
3.2.9 AIM Gateway prototype . . . 47
3.2.10 Summary . . . 49

4 IoT-A M2M Communication Requirements 49
4.1 Requirements Definition Methodology . . . 49
4.2 List of Communication Requirements from Stakeholders and the ETSI . . . 50
4.2.1 Communication Requirements from Stakeholders in Unified Form . . . 50
4.2.2 Communication Requirements from the ETSI . . . 51
4.3 IoT-A M2M API Requirements . . . 52
4.3.1 Nonfunctional: Adaption to Differences in Applications and Devices . . . 52
4.3.2 Device Control . . . 52
4.3.3 Server and Client Communication Models . . . 53
4.3.4 Device Status Monitoring . . . 53
4.3.5 Communication Failure Notification . . . 53
4.3.6 Information on the Device . . . 53
4.3.7 Device Capabilities . . . 53
4.3.8 Communication Properties - Security, QoS, Confirmation . . . 54
4.4 IoT-A M2M Protocol Stack Requirements . . . 54

5 Conclusions 55

6 Appendix 56
6.1 WPAN standardization activities . . . 56
6.1.1 Institute of Electrical and Electronic Engineers . . . 56
6.1.2 Internet Engineering Task Force . . . 56

List of Figures

1 M2M standardization bodies collaboration links 11
2 The flexWARE system concept 13
3 The usenet concept 15
4 IrisNet architecture, with Sensing Agent and Organizing Agent nodes 18
5 Berkley's Local project architecture 19
6 The AIM Gateway protocol stack 27
7 UPnP operating as a bridge between the appliances and the DVE 27
8 AIM Gateway internal architecture 29
9 OSGi layered model architecture 30
10 Pvbrowser block diagram 34
11 Qees Cloud Architecture. 37
12 Shaspa Bridge. 38
13 Shaspa Cloud Architecture. 39
14 InterDigital prototype architecture (source: InterDigital) 42
15 EMMON prototype architecture (source: Critical Software) 45
16 Home Sensor Gateway layered model 47
17 The AIM Gateway protocol stack 48
18 Requirements Definition Methodology 50

List of Acronyms

3GPP	3rd Generation Partnership Project
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
ADSL	Asymetric Digital Subscriber Line
API	Application Programming Interface
CAN	Controller Area Network
CCSA	China Communications Standards Association
CDC	Connected Device Configuration
CLDC	Connected Limited Device Configuration
CoAP	Constrained Application Protocol
DALI	Digital Addressable Lighting Interface
DPWS	Devices Profile for Web Services
ETSI	European Telecommunication Standards Institute
GISFI	Global ICT Standardization Forum for India
GPRS	General Packet Radio Service
HMI	Human Machine Interface
HTTP	HyperText Transfer Protocol
HVAC	Heating Ventilating and Air Conditioning
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
IPv6	Internet Protocol version 6
ITU	International Telecommunications Union
J2SE	Java2 Standard Edition
JSON	JavaScript Object Notation
JSP	JavaServer Pages
M2M	Machine to Machine
M2MSTF	Machine to Machine Standardization Task Force
MIDP	Mobile Information Device Profile
MTU	Master Terminal Unit
MVC	Model-View-Controller
OASIS	Organization for the Advancement of Structured Information Standards
OLE	Object Linknking and Embedding
OMA	Open Mobile Alliance
OPC	Object linking and embedding for Process Control
OSGi	Open Services Gateway initiative
PLC	Programmable Logic Controller
RFID	Radio Frequency IDentification
RPL	Routing Protocol for Low power
RTU	Remote Terminal Unit
SCADA	Supervisory Control And Data Acquisition
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol

SSDP	Simple Service Discovery Protocol
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TIA	Telecommunications Industry Association
TS	Technical Specification
TSG	Technical Specifications Group
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
UPnP	Universal Plug n Play
USB	Universal Serial Bus
UUID	Universally Unique IDentifier
VPN	Virtual Private Network
VSCP	Very Simple Control Protocol
WADL	Web Application Description Language
WSDL	Web Services Description Language
WSN	Wireless Sensor Networks
XML	eXtensible Markup Language
XMPP	eXtensible Messaging and Presence Protocol

1 Introduction

End points of the Internet of Things will be characterized by varying complexity and capabilities, ranging from RFID tags, to sensors and actuators that can be wired or wirelessly networked and interconnected. In order to deal with such heterogeneity, a central concept of the IoT-A architecture is to virtualize these IoT devices and their capabilities by providing a unifying IoT resource abstraction. The abstraction of IoT resources provides the basis for simplified management, discovery of heterogeneous IoT devices and ensures that interactions between them and the orchestration of these interactions takes place in a uniform manner. From the perspective of business process and end user applications, IoT resources expose their capabilities as service end points that can be easily embedded in emerging service oriented enterprise systems and service delivery platforms.
The interaction between IoT endpoints follows the M2M communication concept. The term M2M refers to systems that enable machines to communicate with back-end information systems and/or directly with other machines, in order to provide real-time data. M2M communication can be event-based, that is triggerred by the occurence of a particular event, and/or polling based, that takes place in predefined time intervals. M2M applications that realize M2M communication include four basic stages

- data collection,
- transmission of specific data over a communication network,
- assessment of the data,
- response to the available information.

Based on the previously mentioned stages, M2M applications can be used to orchestrate and deliver services in remote endpoints without the need for human intervention. The logic used to process the collected data can implement complex decision making, thus enabling the provision of services.
A key enabler in accessing IoT resources as service end points is the Machine-to-Machine Application Programming Interface. The M2M API provides the means for the device to expose its capabilities and the services it may offer, so that remote machines may utilize them. Consequently, such an API is necessary to enable proactive and transparent communication of devices, in order to invoke actions in IoT devices and receive the relating responses. A solid, well-designed M2M API provides the basis for the simplified management of resources. Additionally, the main advantage of such an API is that it provides the abstraction layer necessary to realize interactions between IoT devices uniformly.
Last but not least, due to the fact that the data exchanged between machines travels over a communication network, M2M applications are inherently related to the protocol stack used in the communication network. Typically, M2M applications utilize open protocols that are standardized, in order to be widely deployable. Depending on the needs of the M2M application, different protocols may be utilized; i.e. if the application requires reliable message delivery the Transmission Control Protocol can be used instead of the User Datagram Protocol.

The intention of the document is to provide an initial analysis of the requirements for an M2M API for IoT endpoints, which will serve as a starting point for defining the actual services the IoT endpoints should expose via this M2M API. Towards this direction, the current state-of-the-art in M2M communications, in terms of standardization bodies, research projects, protocols, software development platforms and already deployed prototypes was taken into consideration. The requirements identified, in conjuction with requirements imposed by stakeholders involved in the IoT-A project, form the basis upon which the M2M API will be built.

2 State of the Art on M2M Communication

In order to derive requirements for M2M communication in the IoT, an overview of the state-of-the-art in M2M communications is presented. The aim of this overview is to identify current trends and practices applied in M2M communication, in order to use them as input to the IoT-A project M2M API requirements definition methodology. Towards this end, an overview of current standardization bodies and their ongoing work is presented. Furthermore, the work of various research projects related to M2M communication is also described. Finally, an overview of protocols and software platforms used to realize M2M communication is provided.

2.1 Standardization Activities

The realization of the IoT concept, heavily relies on M2M technologies and the standardization efforts on M2M systems. Towards this direction, and with the M2M solutions slowly becoming mainstream, a number of standardization bodies have been formed. The scope of the standardization bodies is to provide a unified, end-to-end M2M system architecture which will be the key enabler for substituting existing proprietary vertical applications.

2.1.1 ETSI M2M Technical Comittee

The European Telecommunications Standards Institute has created a dedicated Technical Comittee for developing standards on M2M communications [1] . This committee aims at developing and maintaining an end-to-end architecture for M2M systems, as well as addressing various M2M communication considerations, such as naming, addressing, location, QoS, security, charging, management, application interfaces and hardware interfaces. Additionally, a major concern of the committee is to integrate sensor networks in the envisagaged architecture.
In order to derive service capabilities and requirements for the overall M2M system architecture, the ETSI M2M technical committee has defined a number of application areas and a number of use cases for each application area. The actual application areas accounted for are eHealth [2], Connected Consumer [3], City Automation [4] and Automotive Applications [5]. The Smart Metering application, driven by the European Commision Standardization Mandate M/441, is of particular importance [6]. The general objective of the mandate

- and consequently of the use case - is to create European standards that will enable interoperability of utility meters (water, gas, electricity, heat), which will be used to improve customers' awareness of the actual consumption. Thus, the consumers will be enabled to allow consumption adaptation to their demands, while improving the energy end-use efficiency.

2.1.2 3GPP TSG Service and System Aspects

The 3rd Generation Partnership Project maintains and develops technical specifications and reports for mobile communication systems. Mobile networks are also concerned with the integration and support of M2M communications, as the nature of M2M systems is substantially differentiated than that of Human-to- Human services, i.e. plain telephone calls, which mobile networks originally addressed. Therefore, the 3GPP Technical Specifications Group dealing with Service and System Aspects [7], has issued a number of specifications dealing with requirements that M2M services and M2M communication imposes on the mobile network.
Specifically, the TS 22.368 [8] specifies the service requirements for M2M communications, by identifying and specifying general requirements, as well as network improvements that are needed to account for the particular features of M2M communications. The TS 22.868 [9] identifies potential requirements targeted at improving M2M communications and the more efficient use of radio and network resources. The particular Techical Specification gives special consideration to optimisations for charging mechanisms, addressing, handling large numbers of subscriptions and subscriber data within the network. Furthermore, it addresses the issue of handling large number of M2M subscriptions for the user of M2M services. Finally, the TS 23.888 [10] studies and evaluates architectural aspects of the M2M-specific system improvements specified in TS 22.368, mainly the architectural enhancements to support a large number of M2M devices in the network and architectural enhancements to fulfil M2M service requirements.

2.1.3 TIA TR-50 Engineering Committee

The Telecommunications Industry Association is the United States counterpart to ETSI with respect to telecommunications standards, developing industry standards for a wide variety of telecommunication products. The standardization activities are assigned to separate Engineering Committees. The TR-50 Engineering Committee Smart Device Communications [11], has been assigned the task to develop and maintain physical-medium-agnostic interface standards, that will enable the monitoring and bi-directional communication of events and information between smart devices and other devices, applications or networks. TR-50 will develop a Smart Device Communications framework that can operate over different types of underlying transport networks (wireless, wired, etc.) and can be adapted to a given transport network by means of an adaptation/convergence layer. The TR-50 framework will make its functionality available to applications through a well-defined Application Programming Interface that is agnostic to the vertical application domain (eHealth, Smart Grid, Industrial Automation, etc.)

Practically, the scope of TR-50 is to identify the requirements of M2M communications, and specify the overall system architecture along with the data models necessary. Ultimately, a universal protocol for communicating with devices used in industries like manufacturing, semiconductor, communication, medical, and building automation will be developed. The target uses of the protocol include monitoring, as well as changing of device state remotely. Types of information/events that TR-50 will accommodate are

- equipment fault events,
- preventative maintenance events,
- energy metrics,
- network metrics,
- equipment data,
- configuration management.

2.1.4 CCSA TC10

The China Communications Standards Association [12] is the organization responsible for carrying out standardization activities in the field of Information and Communication Technologies across China. For a particular area of ICT, a Technical Committee is responsible for developing and maintaining the standards. The M2M communication area is tackled by the 10th technical committee of CCSA, which addresses ubiquituous networks standards in general.
TC10 is organized into four work groups, with WG1 being responsible for the design, architecture and common requirements of ubiquitous networks, WG2 complementing requirements with the ones steaming from industry applications, WG3 identifying network improvement to support the applications and WG4 identifying the necessary extensions to telecom networks in order to support sensor networks as well. As the CCSA TC10 was established in early 2010, the first specifications are expected to be published by late 2011.

2.1.5 Global ICT Standardization Forum for India

The Global ICT Standardization Forum for India (GISFI) [13] is an Indian standardization body active in the area of Information and Communication Technologies and related application areas, such as energy, telemedicine, wireless robotics and biotechnology. GISFI addresses the research and product development of ICT in India, so as to provide a bridge towards the globalization of the Indian achievements; the issues of technology, governance, and development. The working groups organized in GISFI will draw knowledge from academia, business, civil society, and Government/policy-making circles.
GISFI indirectly addresses the issue of M2M communications via its Internet Of Things working group. The scope of the IOT working group is to

- study relevant ongoing standardization activities around the world,

- study specific requirements from Indian perspective and gap analysis with respect to international standardization,
- perform intial study on interoperability, security, management, identity and application development areas in order to identify issues affecting the IOT concept,
- investigate different industry vertical specific use cases, such as healthcare, manufacturing, supply chain, transportation, manufacturing and process.

As the GISFI forum has been recently established, it has not published any specifications and/or reports in the aforementioned areas; therefore, it has not provided any input to the global M2M communications community.

2.1.6 International Telecommunication Union

The International Telecommunication Union is a specialized agency of the United Nations, responsible for information and communication technologies. The Telecommunications Standardization Sector (ITU-T), addresses the issue of M2M communication via the Ubiquitous Sensor Networks-related groups [14], which address the area of networked intelligent sensors. Study groups within ITU-T delaing with Ubiquitous Sensor Networks standardization activities are Study Group 13, which is concerned with functional requirements and architectures, Study Group 16, relating to multimedia service descriptions and requirements aspects and Study Group 17, dealing with security and object identifier aspects.

2.1.7 Open Mobile Alliance

The scope of Open Mobile Alliance (OMA) [15], is to develop mobile service enabler specifications, in order to support the creation of interoperable end-to-end mobile services. Towards this direction, OMA drives service enabler architectures and open enabler interfaces that are independent of the underlying wireless networks and platforms. OMA's data service enablers are intented to work across devices, service providers, operators, networks, and geographies.
As there are several OMA standards that map into the ETSI M2M framework, a link has been established between the two standardization bodies in order to provide associations between ETSI M2M Service Capabilities and OMA Supporting Enablers [16]. Specifically, the expertise of OMA in abstract, protocol-independent APIs creation, as well as the creation of APIs protocol bindings (i.e. REST, SOAP) and especially the expertise of OMA in RESTful APIs is expected to complement the standardization activities of ETSI in the field of M2M communications. Additionally, OMA has identified areas where further standardization will enhance support for generic M2M implementations, i.e.

- device management,
- network APIs addressing M2M service capabilities,
- location services for mobile M2M applications,

- messaging of M2M devices that are sleeping.

The overall aim of the collaboration between ETSI and OMA is twofold. On the one hand, it must be ensured that the APIs defined by OMA to describe service capabilities map into the ETSI M2M framework. On the other hand, there must be a mapping of OMA service enablers to the ETSI M2M framework.

2.1.8 M2M Standardization Task Force

As seen from the above, there is a quite large number of organizations involved in the standardization of M2M communications. In order to enable the development of globally compatible standards, it must be ensured that coordination and collaboration exists between standards organizations. The Global Standards Collaboration (GSC) organization [17] is responsible for organizing the coordination and collaboration activities. GSC brings together top standard officials from the USA, Canada, the EU, China, Japan, Korea, Australia and the International Telecommunication Union to discuss their standards work programs, thus identifying areas of collaboration and identifying ways to accelerate global standards for the indu

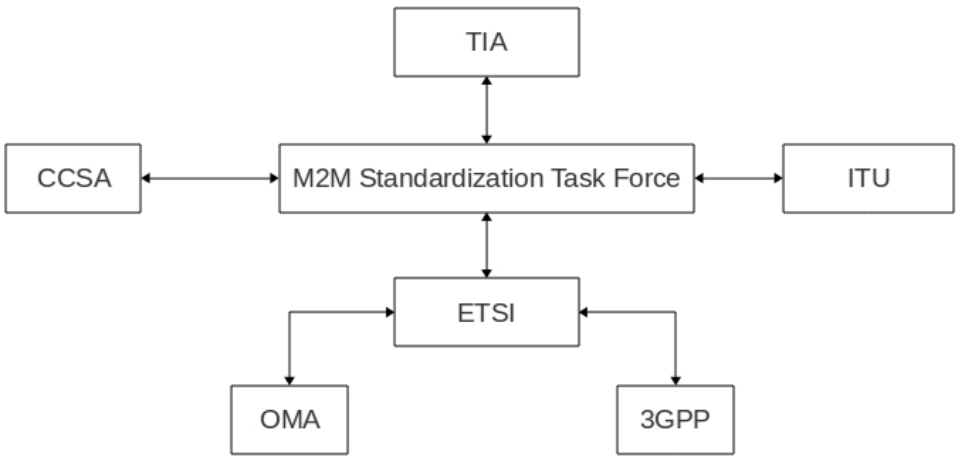

Figure 1: M2M standardization bodies collaboration links

Towards this end, during a Global Standards Collaboration meeting in Beijing in September 2010, the M2M Standardization Task Force (M2MSTF) was created. The M2MSTF scope is to facilitate global coordination and harmonization of M2M standardization work. Due to the fact that the M2MSTF is still in its cradle, its exact organization and coordination efforts could not be investigated - the M2MSTF does not seem to have a working internet site. A high level view of the collaboration links established between the standardization bodies is depicted in Figure 1.

2.1.9 Summary

The growth of the M2M communications industry has lead to the need for interoperability between M2M solutions. Towards this direction, all major standardization bodies have established working groups dedicated to M2M communications. However, all M2M standardization bodies have been recently formed. As a result of that, the overall specifications are under development. Few specifications have been published, mainly addressing the overall system requirements and architecture and not delving into the specifics of M2M APIs. From the point of view of the IoT-A project, the ongoing activities in M2M standardization bodies are of particular importance because they will be setting the standards for M2M communication. Consequently, the standardization bodies will also set requirements and specifications for the functionality that should be implemented by M2M APIs. Therefore, the updates in the work of these bodies must be monitored, in order to ensure the compatibility of the solutions developed by the IoT-A project with the solutions proposed by standardization committees and possibly complement these solutions with additional functionality.

2.2 International Research Activities

2.2.1 European research projects

2.2.1.1 flexWARE, IST FP6, 2008-2010

The project flexWARE [18] (Flexible Wireless Automation in Real-Time Environments) dealt with real-time communication based on different wireless technologies and finished in 2010. The aim was to develop a secure middleware between the physical communication and the application with the focus on security, flexibility and mobility. The idea was that wireless nodes, which are able to move seamless between different access points of the system combined with localization services will enable easy and reconfigurable solutions for production and logistic facilities. Towards this end, the project goal was to establish a wireless infrastructure independent from actual communication technologies used, by developing a middleware which would cope with dynamic reconfiguration of the network topology due to nodes travelling between different access points in the system. The middleware designed should be as secure as possible yet ensuring real-time, QoS (Quality of Service) aware behaviour.
The overall flexWARE system supports:

- Flexible production paths,
- Roaming between wireless domains,
- Network-based control.

Flexible productions paths provide changeable factory automation with varying paths of production goods, that are on the one hand not clearly defined and on the other hand not fully predictable in advance. Production paths need to be re-arranged on short notice, e.g. in case of production machine failures. This has to be done in a flexible way and without extensive reconfiguration. Furthermore wireless real-time networks have to be capable of

hosting multiple domains, allowing all mobile clients to select those providing the best services. Consequently, this requires the network infrastructure to transparently switch between access points. To implement sufficient flexibility, the communication controllers have to interact in order to organise this roaming between domains. Finally, another aspect taken into consideration are network based control and servo loops of an automated production systems.

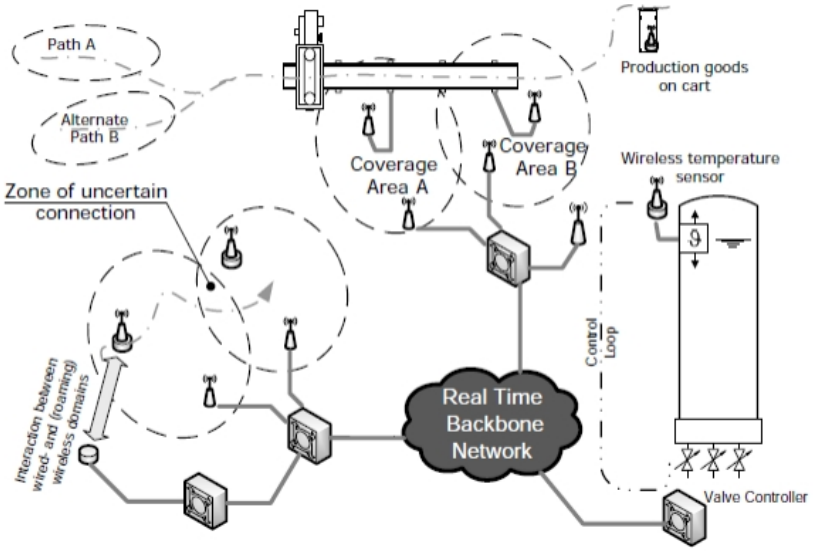

Figure 2: The flexWARE system concept

The overall flexWARE system concept is illustrated in Figure 2. Boxes connected to the Real-Time Network are considered as real-time controllers which are enabled to do the traffic scheduling in both, the wired and the wireless domain. The developed solution offers open interfaces, allowing the connection to already existing systems.

2.2.1.2 SIRENA, ITEA, 2004-2006

The EUREKA ITEA software Cluster SIRENA project [19] dealt with the applicability of SOA in industrial automation. SIRENA laid the foundation for the development and the later standardization of the Device Profile for Web Services (DPWS) technology. The SIRENA framework allows the creation of distributed control and command systems with network-based cooperation between different devices. Out of SIRENA two follow-up projects dealing with DPWS formed: SODA, also an ITEA project, and SOCRADES, an EU FP6 project.

2.2.1.3 SODA, ITEA, 2007-2009

The EUREKA ITEA software Cluster SODA project [20] has created a service oriented approach for communication between computer systems and embedded devices. One of the objectives was on the communication between and with lowcost devices as an key-enabler of the internet of things. The approach is useful for a wide range of industrial scenarios from industrial automation, automotive electronics, home and building automation, telecommunications up to health care. The concept was standardized by the Organization for the Advancement of Structured Information Standards (OASIS) as the Devices Profile for Web Services (DPWS) standard in the middle of 2009. This was an important contribution to the web services protocols that allows linking the properties of the underlying objects with the functions of user services. Today web service protocols are the most commonly used protocols for web communications.

2.2.1.4 Usenet, ITEA, 2006-2009

Information technology applications usually operate as separate M2M solutions that are unaware of each other. As a result of this approach, interoperation between devices and their enabled applications in wired and wireless systems is limited; most M2M solutions provided are vertical and depend on the equipment provided by the supplier. M2M systems usually require the integration of components coming from various stakeholders in the value chain: M2M service providers, M2M operators, M2M device manufacturers, software houses and M2M system integrators. The referred components need to be interoperable in order to establish sensible business operations. Traditionally, M2M solutions have applied vertical architecture and closed solutions.The scope of the Usenet projects was to develop a horizontal architecture for solving the above interoperability problems.
Usenet [21] was focusing on the R&D of generic horizontal technologies related to service communication infrastructures for Machine-to-Machine systems. The project applies and evaluates the developed technologies in different M2M application domains such as remote metering and control of ubiquitous built environment, maintenance and control of (mobile) machines, ubiquitous mobile client for consumer devices, wired and wireless (mobile) telematics systems. Main project objectives are:

- Enable interopreable M2M applications in the M2M service solution,
- Specify generic service infrastructures for M2M application domains,
- Enable communication over the heterogeneous M2M networks,
- Develop advanced M2M endpoints,
- Make use of the M2M services smooth and convenient for mobile users,
- Provide end user services for smart usages,
- Clarify and enhance the stakeholders roles in the domain of M2M,

- Usenet aims at developing a one-size-fit-all M2M architcture based on the server-client communication paradigm, where the M2M communication is hosted centrally by gateways and the transportation protocol is the IP.

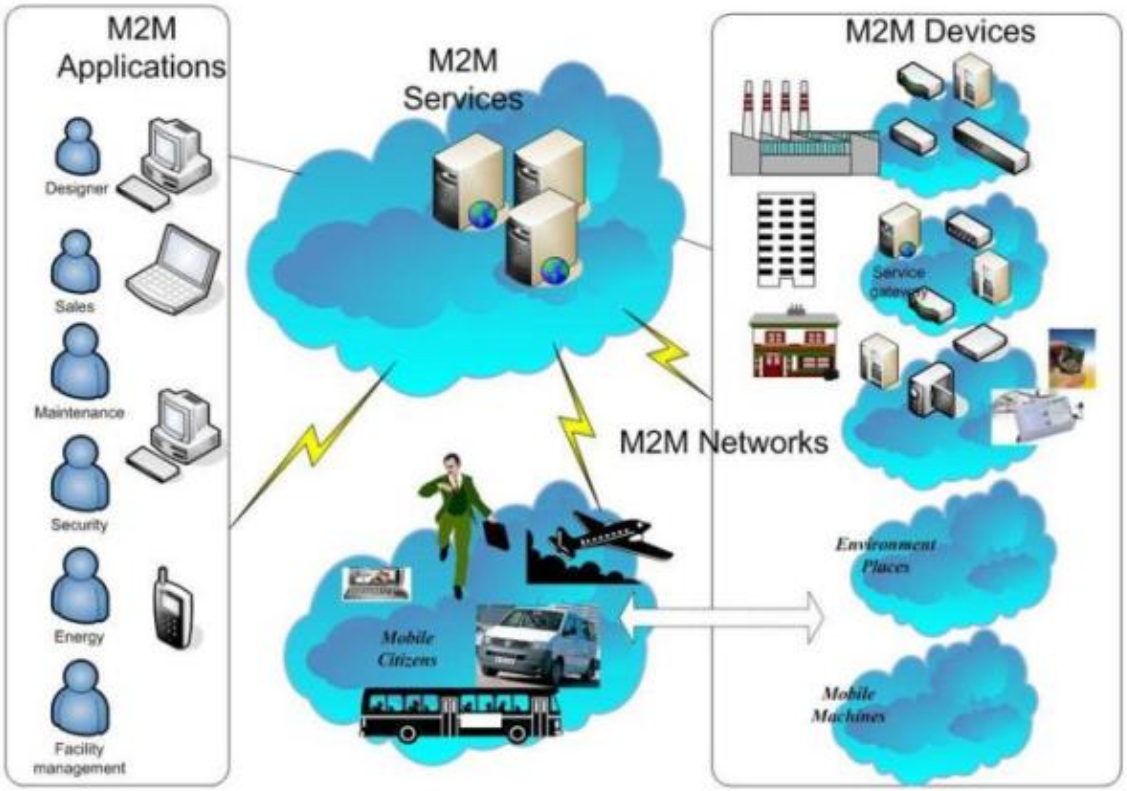

Figure 3: The usenet concept

2.2.1.5 EXALTED, FP7, 2010-2013

The project EXALTED [22] (EXpAnding LTE for Devices), which started in september 2010, aims to develop a scalable network architecture providing secure, energy efficient, and cost effective Machine-to-Machine (M2M) communications suitable for low end devices. The focus is on M2M communication using 3GPP Long Term Evolution (LTE) infrastructure. One area of research deals with the creation of an LTE-M system extending the LTE specifications towards M2M communication. LTE is the sucessor of 3G UMTS, is IPv6 enabled and supportes considerably higher throughput. M2M technology migration into the field of LTE is today strongly supported by 3GPP, the standardisation body of UMTS. A number of issues have been identified as needed for enabling M2M operation on mobile networks:

1. new identiies and addressing schemes to cope with the plethora of connected devices,

2. new communication paradigms, supporting opportunistic communication,
3. security and privacy,
4. new protocols that support small message communication.

2.2.1.6 AIM, FP7, 2008-2010

The AIM project [23] (A novel architecture for modelling, virtualising and managing the energy consumption of household appliances) was about the development of a unified technology for the monitoring, profiling and management of energy consumption in private households. To cover the most of the common household devices AIM was applied on white goods, like refrigerators, communication devices, like telephone or personal computers, and entertainment equipment, like TVs or set-top boxes. The evaluation of the concept and development phases included real-world test beds trying to verify the real energy savings. AIM has worked out a M2M communication concept based on the server-client communication paradigm, where all agents (conected devices-sensors, appliances, etc) communicate with each other through a home gateway that hosts an IP-based M2M protocol layer. The main characteristic of this layer is that it allows management of the connected devices without demanding user involvement, whereas it exhibits a single API towards user services. In next sections, the AIM prototypes are analyzed in detail.

2.2.1.7 e-SENSE, FP6, 2006-2007

The e-SENSE project [24] proposed a context capturing framework enabling the convergence of many input modalities, e.g. wireless sensor networks with many sensors, mobile or static ones, active in different networks. The main contributions of e-SENSE consisted of providing some of the key enabling technologies for Ambient Intelligent Systems. Specifically, the intent has been that of capturing ambient intelligence through low power, distributed and highly efficient WSNs. e-SENSE workplan pivots around capturing of devices information as building elements for ambient intelligence user applications. To this end the project does not work on machine-to-machine communications.

2.2.2 US research projects

2.2.2.1 Berkeley's OpenWSN

The OpenWSN (Open Wireless Sensor Networks) project [25] from the Berkley University is a repository for an open-source implementation of a protocol stack for different applications of the Internet of Things. The idea is to use open standards and a wide variety of different hardware and software platforms. The actual protocol stack proposed by OpenWSN consists of the following technologies (based on [25]) and has implementations for different wireless sensor platforms:

- application: openADR [26], HTTP, sensor.network [27],

- transport: TCP, UDP,
- IP/routing: IETF RPL,
- adaptation: IETF 6LoWPAN,
- medium access: IEEE 802.15.4e,
- physical: IEEE 802.15.4-2006.

2.2.2.2 Berkeley's WEBS

The WEBS (Wireless Embedded Systems) project [28] from the Berkely University is focused on sensors and actuators which communicate wirelessly with each other. The main challenges of those systems which often contain highly embedded and networked devices are the interaction with people and the physical environment, and the limited computation power and energy. Under this project many other well-know projects are consolidated, like sMAP or tinyos.

2.2.2.3 Intel IrisNet, 2003

IrisNet (Internet-scale Resource-Intensive Sensor Network Service) [29] deals with the efficient collection from high-data rate sensors such as cameras. The project provides the data collection infrastructure to let users query globally distributed and high-bit rate sensors. Authoring and deploying sensing services requires addressing a number of challenges related to data acquisition and processing, placement and replication of the sensor data for availability and performance, query processing on the widely distributed sensor data, data integrity and privacy etc. The objective of IrisNet is that of providing these functionalities as well as a simple programming interface such that the service programmer can easily write their new sensing services.
IRIS is composed of a collection of Sensing Agents (SAs) and Organizing Agents (OAs). Any Internet connected, PC-class device can play the role of an OA. Less capable PDA-class devices can act as SAs. Sensor-based services are deployed by orchestrating a group of OAs dedicated to the service. These OAs are responsible for collecting and organizing the sensor data in a fashion that allows for a particular class of queries to be answered (e.g., queries about parking spaces). The OAs index, archive, aggregate, mine and cache data from the SAs to build a system-wide distributed database for that service. In contrast, SAs are shared by all services. An SA collects raw sensor data from a number of sensors. The focus of the Iris design is on sensors, such as webcams, that produce large volumes of data and require sophisticated processing. Key features of IRIS include:

- Providing simple APIs for orchestrating the SAs and OAs to collect, collaboratively process and archive sensor data while minimizing network data transfers.
- Handling issues of service discovery, query routing, semantic caching of responses and load balancing for all services.

- Handling the demanding requirements of processing and querying live and historical high-volume sensor feeds such as webcams.

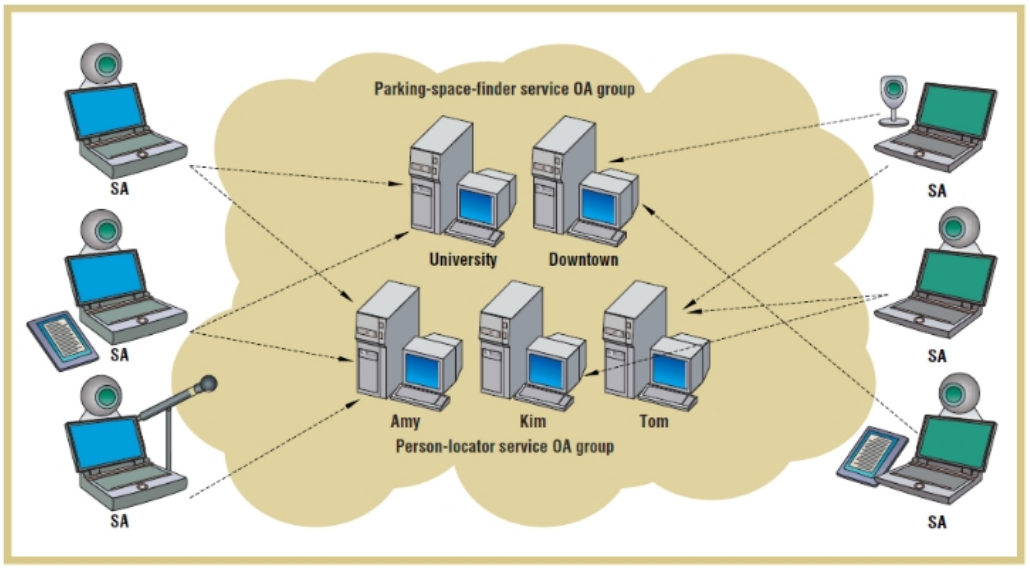

Figure 4: IrisNet architecture, with Sensing Agent and Organizing Agent nodes [30]

IrisNet takes a database centric approach in its design. Service authors specify the new service through a set of XML documents. IrisNet implements service-specific distributed XML databases of the sensor data and provides high level APIs to query and process the sensor data. For more details, please see our publications. IrisNet is mostly written in Java and has been proven useful to a number of real systems, source code and documentation can be downloaded from the project site.

2.2.2.4 Berkeley's Local, 2009-2011

The design of a scalable, flexible and resilient electric power infrastructure is the focus of the LoCal project [31] (A Network Architecture for Localized Electrical Energy Reduction, Generation and Sharing) from the University of Berkley. The idea of the project is to investigate how the design principles of the Internet can be used to develop a future energy infrastructure. The overall architecture proposed by Local is illustrated in Figure 5.

The LoCal Energy Network is a cyber overlay on the energy distribution system for all various physical manifestations, integrating information exchange everywhere that power is transferred. The LoCal Energy Network is a cyber overlay on the energy distribution system in

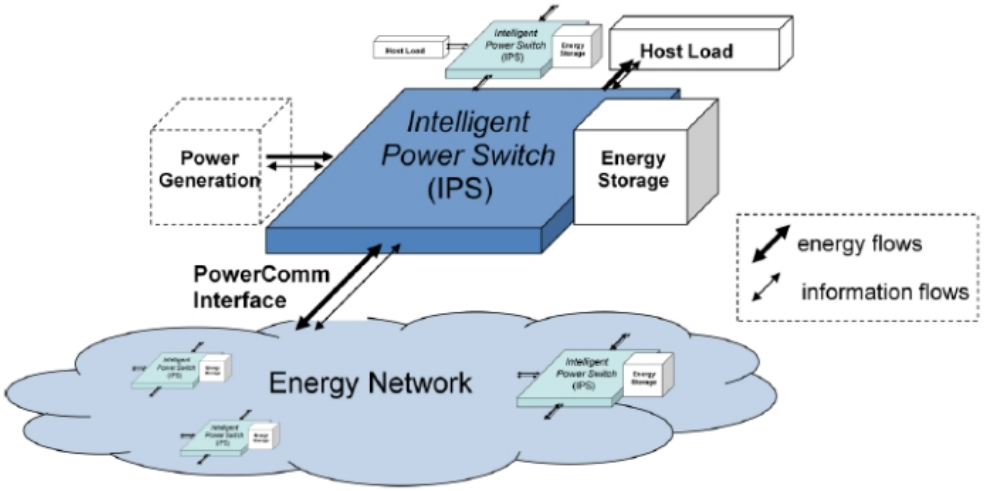

Figure 5: Berkley's Local project architecture [32]

its various physical manifestations, e.g., machine rooms, buildings, neighborhoods, isolated generation islands and regional grids. Pervasive information exchange will enable a more efficient scalable energy system with improved resilience and quality of delivered power. The key contribution of the Local project is to bring together:

1. pervasive information about energy availability and use,
2. interactive load/supply negotiation protocols,
3. controllable loads and sources, and
4. logically packetized energy, buffered and forwarded over a physical energy network.

At its core, a LoCal grid is a connected group of loads, energy sources, and energy storage that intelligently manages its own power needs and interfaces to external power systems in a well-behaved manner. A LoCal grid is designed to operate either independently, connected to the electric grid, and/or to other LoCal grids without affecting its ability to provide power to its users. The networking concepts of LoCal grids are based on ideas derived from the design principles of the Internet. Intelligence and management are shifted to the ends of the system, allowing rapid expansion and high levels of complexity without the increasingly difficult and costly limitations of a centrally controlled system. The unrestricted diversity of the underlying implementation of LoCal nodes is encapsulated by well-defined communications and power interfaces between nodes. The LoCal network is fundamentally a peer-to-peer technology, rather than a client-server technology as is the case in the traditional electric grid.
An essential part of the LoCal network is a communications system that exists in a parallel layer with the power system. Communication between nodes is necessary to coordinate

transfers and to meet aggregate energy needs of an ensemble network. A robust peer-to-peer communications network is also necessary to convey market signals effectively to facilitate the economics of transactions between nodes. Wired, wireless, and power line communications may all be used to suit particular applications.
Even though the project does not explicitly target Internet of Things, the actual application scenario is very much connected to the Internet of Things domains; as a matter of fact, the Local project is talking about a distributed network of sensors and actuators that should communicate their data and interact among themselves and ultimately with the users to achieve the foreseen benefits in terms of energy consumption and distribution. Thus, the project has a lot to do with specializing the Internet of Things concept to Smart Grid networks.

2.2.3 Summary

The collection of international activities on M2M technology has shown very few research activities going on or concluded quite recently. All of them have been carried out in Europe. In the US research, the field of Internet of Things is concentrated on particular issues, such as information capturing technologies (Auto-ID, Iris-NET and WEBS) and energy distribution & management). In the area of Asia there have been no relevant projects identified. In paticular, Europe has already been working for 2-3 years on M2M technologies addressing the following issues:

1. object information profiling in web based communications (SODA),
2. M2M protocol stack (AIM, USENET),
3. enabling M2M support on mobile networks (EXALTED).

A more detailed analysis of these projects' deliverables has led to the following observations

- Supporting compatibility of M2M functionality with web services protocols is a very important requirement to enable M2M portability within user applications all of which are today realized using web based applications.
- It is important to consolidate a common, IP-compatible, protocol stack to enable abstraction between the user applications and the underlying objects by means of accommodating functions brokering on the M2M communication layer.
- Support of M2M communication creates new implications to the underlying networks, with most impostant ones those of security & privacy, object-level addressing, IPv6 support, opportunistic communication.

2.3 M2M Protocols

2.3.1 SCADA

SCADA (Supervisory Control And Data Acquisition) systems can be considered wired ancestors of WSN. Nowadays, SCADA systems rely on long distance communications, so their SCADA transition to the Internet is more than a trend.

SCADA systems constist of a central host or master (usually called a master station, master terminal unit or MTU), one or more field data gathering and control units or remotes (usually called remote stations, remote terminal units, or RTU's) and a collection of standard and/or custom software used to monitor and control remotely located field data elements.
Coming to application fields, SCADA, generally, refers to industrial control systems:

- Manufacturing, production, power generation, fabrication, and refining, and may run in continuous, batch, repetitive, or discrete modes.
- Infrastructure processes may be public or private, and include water treatment and distribution, wastewater collection and treatment, oil and gas pipelines, electrical power transmission and distribution, Wind farms, civil defense siren systems, and large communication systems.
- Facility processes occur both in public facilities and private ones, including buildings, airports, ships, and space stations. They monitor and control HVAC, access, and energy consumption.

Some basic characteristics of SCADA systems are:

- In some cases system data should be time-tagged to the nearest millisecond, therefore Real Time distributed clocks synchronization is a seminal requirement (RTU);
- They have a frame-based network communication technology that is not available in Ethernet and TCP/IP based protocols and is needed in particular M2M applications for message determination, synchronization, protocol selection, environment suitability;
- SCADA systems are converging with standard networking technologies. Ethernet and TCP/IP based protocols are replacing the older proprietary standards. The vast majority of markets have accepted Ethernet networks at least for Human Machine Interface for SCADA.
- SCADA systems are becoming increasingly ubiquitous. Thin clients, web portals, and web based products are gaining popularity with most major vendors. The increased convenience of end users viewing their processes remotely introduces security considerations. While these considerations are already considered solved in other sectors of Internet services, not all entities responsible for deploying SCADA systems have understood the changes in accessibility and threat scope implicit in connecting a system to the Internet.

Nowadays, vendors have begun offering application-specific SCADA systems hosted on remote platforms over the Internet. This reduces the total cost of ownership and removes the need of full system deployment in the user facilities, thus enforcing convergence towards standard TCP/IP protocols. With this kind of deployment, security, internet connection reliability and latency are consistent concerns, often reliyng on common Internet technologies such as VPNs and SSL. SCADA has been made available also for Cellular Networks [42] [43].

2.3.1.1 Modbus

Modbus [40] [41] is a serial communications protocol extensively used in SCADA systems to estabilsh a communication between RTU and devices. Many of the data types are named from its use in driving relays: a single-bit physical output is called a coil, and a single-bit physical input is called a discrete input or a contact. Frames, both on query message from master and response message from slave, carry the following data: device address, function code, data bytes and error check.

2.3.2 UPNP

Universal Plug and Play (UPnP) [45] [49] is a set of networking protocols, mainly designed for residential networks, that enables networked devices, such as personal computers, printers, Internet gateways, Wi-Fi access points and mobile devices to seamlessly discover each other's presence on the network and to establish network services for entertainment, data sharing, and communications. The concept of UPnP is an extension of plug-and-play, a technology for dynamically attaching devices directly to a computer, although UPnP is not directly related to the earlier plug-and-play technology. UPnP devices are "plug-and-play" because when connected to a network they automatically (zero configuration) "collaborate" with other devices.
UPnP is relevant to M2M, telling apart the presentation step. UPnP put a lot of focus on video streaming, which is not so relevant to IoT, but all the mechanics involved are valid.

2.3.2.1 Addressing

Addressing, is step 0 in UPnP, relies on dhcp or Auto-IP. UPnP Addressing mode is not directly portable outside IPv4 or IPv6, but it gives a good argument for using IP as the underlying network protocol for M2M communications.

2.3.2.2 Discovery

Discovery, is step 1 in UPnP. It allows a device added to a network to advertise itself and its services within the network, mainly on the control point. Also the opposite works, enabling a new control point added to a network to search for services and devices within the network. Discovery heavily relies on multicast, rendering UPnP unsuited for networks without multicast capabilities. In order to reduce network congestion multicast UPnP discovery packets have limited Time-To-Live (usually 1 or 2). SSDP (Simple Service Discovery Protocol) is the protocol used by UPnP discovery, an application protocol, using HTTP 1.1 headers format, which is not HTTP 1.1 itself using UDP as underlying protocol in order to be able to exploit multicast packets. SSDP messages introduce the crucial concept of UUID (Universally Unique Identifier), an identifier fixed over time, unique for each device, heavily used in the following UPnP steps.

2.3.2.3 Description

Description, is step 2 in UPnP. After the discovery of a new device or set of devices within the network, the control point knows only information present into the SSDP message: the device UPnP type, the UUID and the URL where to get the description. For the control point to learn more about the device and its capabilities, or to interact with the device, the control point must retrieve a description of the device and its capabilities from the URL provided by the device in the discovery message. The description is an XML listing expliciting a lot more information. In this XML listing elements are mainly specified from the UPnP Architecture, but there is room also for vendor specific elements, while values are specified for a given set of devices by a given UPnP forum Working Committee or by the UPnP vendor. The transfer of this XML listing is done over HTTP.

2.3.2.4 Control

Control is step 3 in UPnP. Given knowledge of a device and its services, a control point can ask those services to invoke actions and receive responses indicating the result of the action. Invoking actions is a kind of remote procedure call; a control point sends the action to the device's service, and when the action has completed (or failed), the service returns any results or errors. Control is done through SOAP over HTTP.

2.3.2.5 Eventing

Eventing is step 4 in UPnP. This step is closely coupled with Control. Eventing can be both Unicast and Multicast. Unicast eventing relies on GENA (General Event Notification Architecture) over HTTP, following the Publisher/Subscriber model, where Publisher is the source of events and the Subscriber is the destination of the events, being, mainly, a control point. Multicast eventing relies on XML with HTTP headers over UDP, in order to overcome TCP lack of Multicast capabilities.

2.3.2.6 Presentation

Presentation is step 5 in UPnP. This layer focus on Human Machine Interfaces (HMI), therefore it is outside the scope of this document.

2.3.2.7 IGD

Internet Gateway Device (IGD) Standardized Device Control Protocol is supported by some NAT routers. It is a common method of automatically configuring port forwarding, but is not an Internet Engineering Task Force document. Applications using peer-to-peer networks, multiplayer gaming, and remote assistance programs need a way to communicate through home and business gateways. Without IGD one has to manually configure the gateway to allow traffic through, a process which is time consuming and can be difficult. Universal Plug and Play (UPnP) comes with a solution for network address translation traversal. IGD makes it easy to do the following:

- Learn the public (external) IP address
- Enumerate existing port mappings
- Add and remove port mappings
- Assign lease times to mappings

2.3.3 NAT Portmapping Protocol

NAT Port Mapping Protocol (NAT-PMP) [46] is an Internet Engineering Task Force Internet Draft, introduced by Apple Computer as an alternative to the more common Internet Gateway Device (IGD) Standardized Device Control Protocol implemented in many network address translation (NAT) routers. It was introduced in June 2005. NAT-PMP allows a computer in a private network (behind a NAT router) to automatically configure the router to allow parties outside the private network to contact itself. NAT-PMP runs over UDP. It essentially automates the process of port forwarding.
Included in the protocol is a method for retrieving the public IP address of a NAT gateway, thus allowing a client to make this public IP address and port number known to peers that may wish to communicate with it.

2.3.4 DPWS

The Devices Profile for Web Services (DPWS) [47] [48] defines a minimal set of implementation constraints to enable secure Web Service messaging, discovery, description, and eventing on resource-constrained devices.
Its objectives are similar to those of Universal Plug and Play (UPnP) but, in addition, DPWS is fully aligned with Web Services technology and includes numerous extension points allowing for seamless integration of device-provided services in enterprise-wide application scenarios.
DPWS builds on the following core Web Services standards: WSDL 1.1, XML Schema, SOAP 1.2, WS-Addressing, and further comprises WS-MetadataExchange, WS-Transfer, WS-Policy, WS-Security, WS-Discovery and WS-Eventing.
DPWS overlaps uPNP.

2.3.5 XML protocols

The BiTXml [50] communication protocol has been designed to implement a presentation level of the (OSI-based) communication stack reference, with the main goal to standardize the way commands and control information are exchanged for the specific target of M2M communication demands (i.e. communication with generic devices with or without processing power on board - like sensors, actuators, as well as air conditioning systems, lifts, .. - or a combination of them). The current protocol specification defines

- The abstraction of a BiTXml gateway

- The abstraction of a BiTXml controller
- The syntax and the semantic of a set of values used to exchange data between two communicating parties
- The syntax and semantic of a set of primary commands used to drive the devices connected to the BiTXml controller

The protocol syntax will be meta-coded in Xml Schema language. The protocol semantic will be defined in natural language (no formalization will be provided).

The purpose of the M2MXML [51] project is to develop an open-standard XML based protocol for Machine-To-Machine (M2M) communications. The primary design philosophy of M2MXML is simplicity. Other attempts to develop protocols for M2M communications have resulted in protocols that are difficult to understand and too verbose to be used in small devices with limited communications bandwidth. Currently, most M2M applications are a custom undertaking from end-to-end, including in many cases the development of custom communication protocols. One of the goals of M2MXML is to establish an open-standard that can be adopted by device manufacturers and M2M application developers, this allowing some interoperability that does not exist today. The M2MXML project will include the development of open-source APIs and code libraries to facilitate the use of the protocol by M2M application developers.

2.3.6 COAP

The Constrained Application Protocol (CoAP) [44] is a specialized web transfer protocol for use with constrained networks and nodes for machine-to-machine applications such as smart energy and building automation. These constrained nodes often have 8-bit microcontrollers with small amounts of ROM and RAM, while networks such as 6LoWPAN often have high packet error rates and a typical throughput of 10s of kbit/s. CoAP provides a method/response interaction model between application end-points, supports built-in resource discovery, and includes key web concepts such as URIs and content-types. CoAP easily translates to HTTP for integration with the web while meeting specialized requirements such as multicast support, very low overhead and simplicity for constrained environments.

2.3.7 Internet protocols

As has been identified in section 2.2:"International Research Activities", research efforts in the domain of M2M communications and technology converges towards the use of Internet protocols for the following reasons:

1. Internet protocols are the industry standard for user applications.
2. If used ensure solution portability to a vast range of network components (embedded computers, PCs, SCADA devices, commercial communication solutions, etc), user terminals (mobile, wireless and fixed).

3. Offer instant integration with user applications.

The term Internet protocols is used to identify a number of plug n'play, IP compatible protocols, which today are very popoular, such as the HTTP/HTTPS, the UPnP described above, the Simple Object Access Protocol (SOAP), the General Event Notification Architecture (GENA).
To explain how M2M functionality can be provided as part of the Internet protocols, in this section, we make reference to the AIM project which has been also analyzed in section 2.2 as part of the International research activities. The AIM project gives a fine expample of how M2M functions can be accommodated within the IP-based protocol stack of the Internet protocols to support a wide range of applications for several application domains, such as health, energy management, home environment control, etc.
The AIM project itself has targeted the appication domain of energy control in residential environments by offering a M2M-enabled technology for managing in real time the energy consumption of domestic appliances. The project uses M2M functions to collect appliance consumption information over the home network and furnishes a virtualisation layer on to of them to enable information exploitation in the frame of energy management services for the residential users.
The key component of the AIM architecture is the AIM Gateway. The AIM gateway is a communication component that has the ability to host user services, while serving communication with user terminals over the indoor and outdoor networks and implementing control of the household appliances by employing special communication interfaces, tailored to the communication interfaces of the household appliances - i.e. KNX, Zigbee.
The following text is an extract from the AIM public deliverable 4.2 "Integration logic with the home network" that illustrates an example of how M2M communication can be realized in a protocol stack made up of internet protocols. In the following extract the following definitions apply:

1. EMD is the Energy Management Device, which is the part of the energy management responsible for implementing the lower-layer, appliance-specific functions.

2. DVE is the Device Virtualization Environment, which is practically a web page through which AIM system users are able to access, monitor and configure energy monitoring and management functions.

The main part of the protocol stack outline in Figure 6, pertains to the main functionality of the AIM logic that consists of the Residential (or Home) Gateway, the EMD and the Operator platform (service logic), where users services can be optionally accommodated if they are not hosted on the home gateway. Automated identification of the connected appliances regarding their function type (fridge, washing machine, TV, dryer, kitchen, etc) and supported functions (programes) is performed using the UPnP protocol. UPnP allows encoding appliance infomation in particular information fields, which are visisble from the overlay service protocols (SOAP) and thus enable automated integration within the service logic.

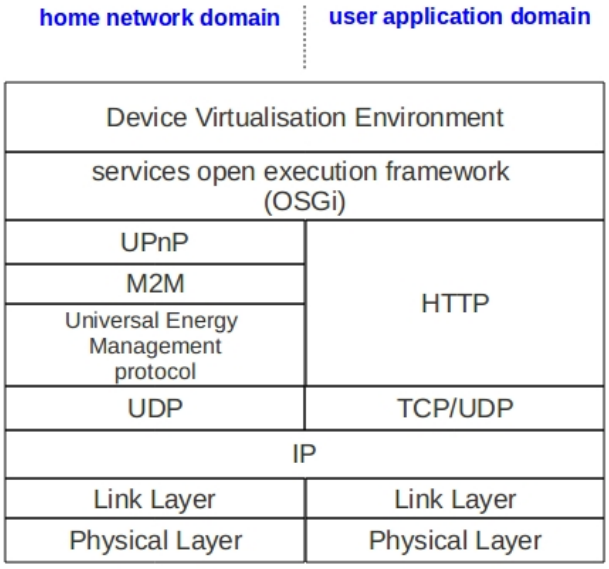

Figure 6: The AIM Gateway protocol stack

To cope with the various specificities of the appliances concerning UPnP support, an intermediate bridging function was defined, which plays the role of the message "translator" between requests coming from the UPnP protocol (UPnP Control Point) and appliance responses, which are modified so that to be compatible with the UPnP protocol. The overall communication scheme is illustrated in detail in the following figure.

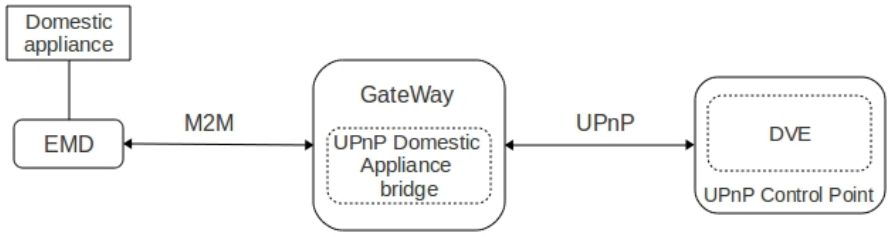

Figure 7: UPnP operating as a bridge between the appliances and the DVE

According to this, the gateway exploits the internal M2M layer to provide a single API towards the bridging UPnP function of the gateway, which responds to the standard requests of users' UPnP protocol using the standard UPnP protocol functions. Here it is worthy to note that the UPnP bridging function is designed so that to server simultaneously all connected appliances acting as concentrator and responder of requests coming from the Device Virtualisation Environment (DVE).

2.3.7.1 Internet Protocols

Communication of the home AIM system with the user terminals is realized using standard protocols, widely used in Internet communications. The enabler of these protocols is actually the IP and its UDP and TCP extensions. With the use of the HTTP protocol the AIM system is able to use industry standard software on the user terminal to activate, manage and use the services offered by the AIM system, such as web browsers, operator applications or any other third party software. This way, the services of the AIM system remain accessible for all IP users, whereas they become independent from the user terminal type.
System configuration is performed by means of a local web server that contains configuration pages for:

- create, update, modify AIM-specific protocols
- configure the IP address of the system,
- add,remove and maintain the physical interfaces (wireless, ZigBEE, powerline, etc).
- creation, modify and manage energy management services within the Device Virtualisation Environment (DVE).

2.3.7.2 Machine-to-Machine interfaces

The Machine-to-Machine interface (M2M API), illustrated in the figure above, has the task of implementing interoperability between the different functions supported by the connected appliances. The M2M API is physically implemented as a bundle so that it can be modified when needed, for example when a new appliance is added to the gateway. The M2M API is actually composed of a low-level interface, providing connectivity with the drivers of the connected appliances, and a high-level one that enables integration of appliance functions within the frame of user protocols.
In the AIM system, the M2M API has been implemented in the form of bundle as part of the OSGi (Open Services Gateway Initiative) environment. The OSGi is an open, license-free software environment that enables easy and rapid implementation of user services for home applications. It has been engineered as an overlay software environment that is accommodated within the operating system and provides a virtual abstraction layer between the hardware communication interfaces and the user services. Among the most important advantages of OSGi is the freedom that it gives to programmers to add new protocol implementations above the IP stack, hence enabling creation of application-centric solutions. OSGi is available for Linux OS versions, which is the industry standard OS for the impementation of home gateways with embedded CPUs.

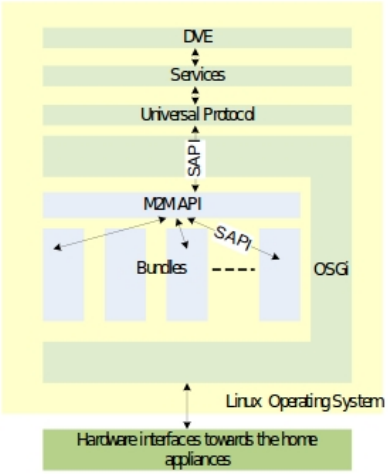

Figure 8: AIM Gateway internal architecture

2.3.8 Summary

Summarizing the protocols information outlined in this section it can be pointed out that the tendency for M2M functionality implementation is clearly towards integration within the protocol stack of Internet protocols. This way it is ensured maximum compatibility of M2M communication with underlying networking technologies, such as those addressed in this section (SCADA, NAT, COAP), as the IP is the industry standard in nearly every communication domain, while on the other hand solution portability to existing user applications and terminals is possible in every Operating System without additional programming effort.

2.4 Software Platforms

2.4.1 Open Source Platforms

There has been significant activity in the open source community relating to the development of platforms and operating environments upon which M2M applications can be built. This section provides an overview of the most important open-source platforms used to realize M2M communication.

2.4.1.1 OSGi

The OSGi technology [58] is a set of specifications that define a dynamic module system for the Java platform. These specifications enable a development model where applications are (dynamically) composed of many different (reusable) components. OSGi technology is based on a service-oriented architecture, enabling components to dynamically discover each other and collaborate. A service within the context of OSGi, is specified as a Java interface implemented by one or more bundles. Lookups can be used to track services from other bundles using a query language. A bundle is practically a java .jar file, that realizes

the deliverable application and registers the services the application is offering. When a bundle is stopped, the services it has registered are removed along with references to other services. Bundles can be notified when a service they depend on is unregistered and class path dependencies are managed. This model allows long running applications with dynamic software updates.
The OSGi architecture adopts a layered model depicted in Figure 9. The Bundles are the

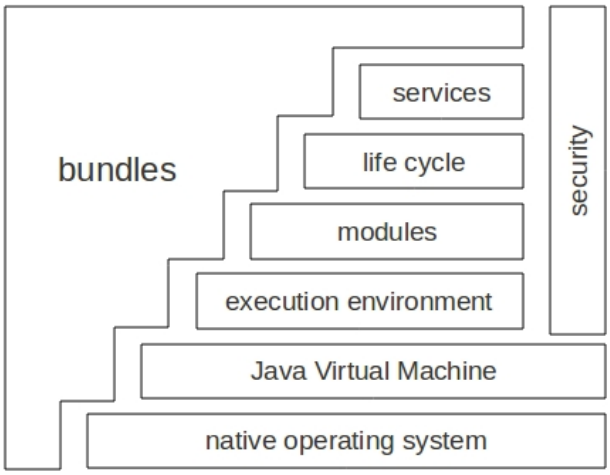

Figure 9: OSGi layered model architecture

OSGi components made by the developers. The services layer is responsible for connecting bundles in a dynamic way by offering a publish-find-bind model for plain old Java objects. The service layer provides an OSGi service registry, where the interfaces implemented by objects of the bundles are registered. Consequently, other bundles may retrieve to the service registry and list all objects that are registered under a particular interface or class. In order to disambiguate bundle objects registered under the same interface or class, each service registration is accompanied by standard and custom properties. The Life-Cycle layer provides the APIs to install, start, stop, update and uninstall bundles.
The modules layer defines how a bundle can import and export code. This layer is required due to the fact that OSGi does not allow any code sharing by default, in contrast to standard Java. While in standard Java the contents of a .jar file is completely visible to other jars, OSGi hides everything in that .jar file unless explicitly exported. Furthermore, a bundle that wishes to use another jar must explicilty import the parts it needs. OSGi security is based on Java and the Java2 security model, utilizing the Java built-in security features. Finally, the execution environment is a specification of the Java environment. Java2 configurations and profiles such as J2SE, CDC, CLDC and MIDP are all valid execution environments.
One of the most essential features of OSGi, is that bundles may be updated dynamically without the need to shut down the framework. Such updates may be done by remote access to the framework, thus allowing the provision of dynamic APIs, i.e. a new bundle

provides new functions, which may be discovered and accessed by other modules without prior definition of a higher layer API. Another key concept of OSGi that is particularly suited to M2M communications, is the usage of OSGi bundles as proxies for services, which are not contained on the OSGi platform itself, but hosted on other devices in a network. Such applications may be used by OSGi bundles in the very same way as bundles which are installed locally. Therefore, the use of OSGi bundles which are native to other technologies (such as Universal Plug and Play, Bluetooth etc) may be used to offer functions contained in the existing OSGi platform - and may be offered (or published) to other devices as native functions, although they are not actually native.
There are currently a number of open-source OSGi framework implementations, which have been adopted by the community of OSGi developers. Widely-used OSGi frameworks include the Apache Felix [59], Makewave Knopflerfish [60], and the Eclipse Equinox [61].

2.4.1.2 Mango

Mango [52] is a web-based, AJAX-enabled M2M software, which enables users to access and control electronic sensors, devices and machines over multiple protocols simultaneously. It provides a database to store the data collected, a Human-Machine Interface to provide graphs, diagnostic data and management information. According to Mango philosophy, end devices are treated as data "sources", which are polled in user-defined time interval for data collection. Depending on conditions configured by the Mango user the data values may lead to events triggering, which are acted upon by the corresponding handler - i.e. sending an e-mail or setting the value of a device.
The Mango M2M application development environment can be used for manual monitoring, control, and data logging, but the automation power of Mango is in the event subsystem. Events are split into two parts: the detection and the handling. These parts can be arbitrarily paired in a one-to-many fashion, and can also be chained. An event detector can be something like a high limit (for analog data) or a binary/multi-state state. For example, you can define conditions such as "when the humidity exceeded 60% for more than 1 hour". (Conditions depend on the data type of the point, so a somewhat different list of options will be displayed for different points.)
Mango is practically a server application which requires a Java Servlet/JSP container to run. As a Java-enabled application,it is operating-system independent. From an implementation point of view, Mango uses the Spring Web MVC framework, Direct Web Monitoring and the Dojo JavaScript toolkit.

2.4.1.3 Very Simple Control Protocol & Friends

The Very Simple Control Protocol [53] is a protocol developed for use on low end microcontrollers. Its target area is remote measurement and control, as well as home automation. In order to provide a complete solution, VSCP does not come stand-alone, but is instead collected in a package reffered to as VSCP & Friends. The package includes

- the VSCP protocol specification and software,

- web-related tools for realising home automation on top of VSCP,
- the low level driver interface used by VSCP.

VSCP uses an event format and supports global unique identifiers for nodes, thus making a node identifiable no matter where it is installed in the world. Furthermore, it includes a register model in order to provide a flexible common interface for node configuration and a model for controlling the functionality of each node. VSCP does not make any assumptions regarding the the lower level system used to realize physical interconnection with the node, therefore it works with different transport mechanism such as Ethernet, TCP/IP, Wireless, Zigbee, Bluetooth, CAN, GPRS, RS-232, USB.
VSCP is event-based. Every time an event occurs, it is broadcasted to all other nodes on the network. From there on, each node will decide on its own if the event received needs to be processed or not. The final decision depends on the node's decision matrix. The decision matrix is made up of a number of if <condition> then <action> lines, where the <condition> is evaluated based on fields present in the VSCP datagram broadcasted to the network.

2.4.1.4 AllJoyn

AllJoyn [54] is a peer-to-peer framework, which enables ad-hoc, proximity based, device-to-device communication thus eliminating the need of an intermediary server. Practically, AllJoyn defines a device-to-device communication protocol which enables mobile devices to support peer-to-peer applications. It is designed as a backwards-compatible extension of DBus, a standard protocol for inter-application communication in the Linux desktop environment. It is platform and operating system agnostic, whereas its API provides support for both C++ and Java. AllJoyn addresses the issues of discovery and network complexity management in peer-to-peer networks, in order to enable nearby devices communicate directly with one another over Wi-Fi or Bluetooth without the need to connect to cellular networks. Towards this end, it provides application developers with a simple API for enabling ad-hoc networks from within their applications. The AllJoyn protocol focuses on solving communication barriers related to peer-to-peer communication, such as

- transparently managing device and service discovery,
- managing network and message routing,
- providing a security framework for message authentication and encryption,
- designed to have minimal requirements on the host operating system, and be hardware and radio technology agnostic.

Furthermore, AllJoyn is optimized for the mobile embedded environment, providing features such as low latency, low bandwidth by means of header compression, supporting both reliable and unreliable transport as well as point-to-multipoint communications.

2.4.1.5 OpenSCADA

OpenSCADA [55] is an open source Supervisory Control And Data Acquisition System, built using the OSGi framework. It does not provide an out of the box solution, but a set of tools that can be combined to create SCADA applications. Therefore, it provides development libraries, interface applications, mass configuration tools, as well as front-end and back-end applications.
The various functionality subsets are provided by different subprojects within OpenSCADA, with the main ones being ATLANTIS and UTGART. The former is the main SCADA component; it contains the implementation of the OpenSCADA interfaces in Java, and provides modules for interfacing with external systems like S7 PLC, OPC, SNMP, relational databases, etc. Data acquired by these interface modules can be processed, monitored and archived by the components of ATLANTIS. Furthermore, OpenSCADA provides components for building custom client and server applications. UTGART on the other hand is a vendor-independent Java OPC Client API, that can be also used independently from other OpenSCADA projects. The scope of UTGART is to provide functions that enable connection to an OPC server. UTGART is used in conjuction with ATLANTIS to enable connection and communication with third party systems via OPC. As OpenSCADA is implemented entirely in Java it is platform-independent.

2.4.1.6 Proview

Proview [56] is a Process Control System, containing functions required for sequential control, data acquisition, communication, supervision, etc. The main concept of Proview is based on a soft-PLC solution, which runs on standard computers running the Linux operating system.
Proview is a distributed system, therefore the overall process control system may consist of several computers connected via a network, where the network of preference is Ethernet. The typical Proview system consists of one process control system and one or more operator stations. Programming Proview is possible both via a graphical PLC editor and with high-level programming languages such as C, C++, or Java. The common I/O system used in Proview is Profibus, a robust and well-tested field bus. However, there is also support for other I/O systems such as Modbus TCP, whereas there is the capability to implement other I/O systems with available drivers or develop new ones using high-level languages. Finally, Proview supports communications with other process control equipment over the OPC XML/DA protocol.

2.4.1.7 Pvbrowser

Pvbrowser [57] is a SCADA application framework, which provides a specialized browser for the client computer, and an integrated development environment for creating servers that implement the visualization presented on the client. Additionally, it provides data acquisition programs for a variety of protocols that the server uses to communicate with the real world. The data acquisition programs are realized in the form of daemons, where a separate daemon is used for each interface. The daemon reads the interface and writes the result in

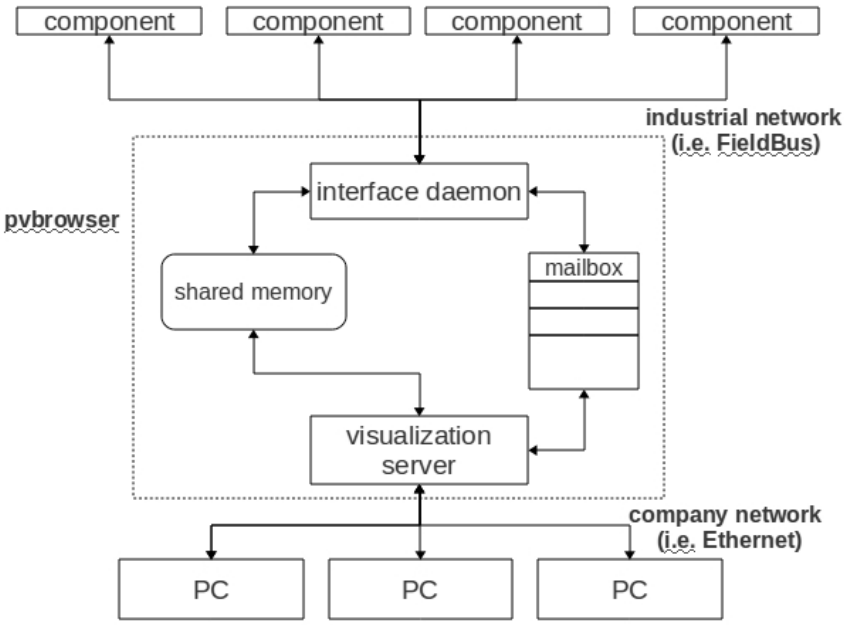

Figure 10: Pvbrowser block diagram

a shared memory. The visualization server may read the shared memory and display its content, thus revealing the interface information to the end user. In the reverse direction, the interface daemon has a mailbox, which is written by the virtualization server to output signals to the interface. The overall Pvbrowser architecture is summarized in Figure 10.

2.4.2 Commercial Software Development Platforms and Operating Environments

The overall approach of commercial software development platforms is to provide functional blocks that cover common application requirements such as security, notifications and alarms generation, network monitoring and from there on enable the applications to communicate with end devices using a customer specific API. The details of the actual protocols and interfaces required to realize communication with end devices are handled internally by the platform. Furthermore, a device-specific adaptor is integrated into the platform in order to communicate with each device.

Examples of the aforementioned approach include the Apprize M2M communications platform [62], the SensorLogic Service Delivery Platform [63], as well as the SeeControl Device Cloud [64] M2M applications platform. It should be noted that the aforementioned platforms target at the communication of a centralized server with remote devices for data collection, and do not seem to truly address the issue of direct M2M communication. An exception to the rule is the deviceWise [65] product suite of ILS technologies, which apart from including a generic software framework for realizing M2M communications, also includes a gateway. The deviceWise gateway is an integration appliance that maps data between different de-

vices without requiring any configuration changes in either device. The particular gateway includes USB, Ethernet and serial interfaces for realizing connectivity with end devices.

2.4.3 Summary

This section provided an overview of the most important open-source and commercial platforms used to realize M2M communication. The bottom line of this overview is that the vast majority of M2M platforms, both commercial and open-source do not seem to truly address M2M communication, as they require the presence of a centralized point where information is maintained and decisions are made. The typical pattern they follow is having a centralized server with a Human-Machine Interface, where the end user is able to poll devices, gather information about their status and perform actions necessary. Nevertheless, there are exceptions, meanly the VSCP approach which utilizes event broadcasting as well as AllJoyn, which addresses peer-to-peer communication of mobile devices. In the commercial products landscape, there is the deviseWise gateway, which incorporates a variety of physical interfaces to enable the direct interfacing of heterogeneous devices. However, all of the aforementioned solutions are vertical and do not provide a generalized framework for realizing M2M communications.

3 Applications

This section gives an overview of applications enabled by M2M communication. According to recent publications [88, 89] M2M communication is used for example in building automation, industrial automation, security, transportation, healthcare, smart energy, smart grid, supply and provisioning, logistics, and city automation. In the following we will focus on some typical examples of those application areas to illustrate typical use cases and scenarios.
In general, a distinction is made between wired, cellular and capillary M2M applications [89]. The terms wired (e.g. Ethernet), cellular (e.g. GPRS, 3G) and capilary (802.15.4, 802.11) refer in this context to the physical transmissuion media of the M2M communication. While wired M2M communication offers the best reliability and highest data rates, it is expensive in the roll-out and not scalable. Cellular solutions with a dedicated cellular link offer a good coverage but are not energy efficient and expensive in rollout and maintenance. Capilary solutions make use of local wireless links, such as 802.11 or 802.15.4. These solutions are cheap, scalable and have acceptable energy consumption, but show weakness in coverage, data rates and have weaker security than the other solutions. Hybrid solutions using capilary technology to reach aggregating devices or gateways and continuing the commuication from there with cellular or wired links will probably be able to combine the strength and overcome the weakness of the single technology solutions. This combination of technologies needs to be reflected in the M2M API.
The smart metering / smart grid application area deals with the control, management and usage of enerygy ressources. Smart metering measures the energy consumption of household appliances and collects the data at a central smart meter. The collected information may be accessed by the owner and made available remotely for billing of the energy provider. The demand-response use case combines the price- and availability information of the energy

provider and the distributed energy ressources (e.g. solar panels) to control the household appliances.
The building automation domain encloses the different areas HVAC, light, security, fire. For a heating application, the communication will be among temperature sensors, actuators controlling the heating devices, calendar objects and other inputs controlling the temperature according schedules or room usage. Additionally, the application is supervised from a (potentially remote) control centre. For commissioning and restart, the devices are typically implemented as an object model that allows for combination and discovery of services.
The eHealth application scenario consists of the data collection of body/health sensors (e.g. blood pressure, ecg, hart rate) and movement sensors (pedometer). The data is potentially made available also remotely and in critical situations there may be a need for 'real time' survailence of health data. While the communication among the body sensors is likely to be of capillary nature, the remote access will require wired or cellular M2M communications.

3.1 Commercial Products

3.1.1 Home Automation / Smart Home / Commercial Building Automation

3.1.1.1 DESIGO commercial building automation systems

The SIEMENS [66] product line DESIGO for commercial building automation systems provides a scalable solution for HVAC(Heating, Ventilation, and Air Conditioning) and shading control for commercial for commercial buildings, such as airports, hotels and shopping malls. It integrates over all automation levels, from basic sensor IO to the management level. The basic protocol and the APIs are based on the BACnet standard. It supports the integration of LonWorks and KNX based third party products.

3.1.1.2 Home Automation Europe

Home Automation Europe [68] provides solutions for home security, monitoring systems, power consumption measurement, home control. Details on the M2M techonology and the used APIs are not supplied.

3.1.1.3 EnergyHub

Energyhub [69] provides solutions for energy control of the home and energy data to the utilitys companies. The energyhub is stationed in the home and controls the data from the sensor metering network. Sensor data is sent to the back-end application in the nework using web services. Energyhub communicates with M2M wireless technology Zigbee.

3.1.1.4 QEES

QEES [67] is a global supplier of intelligent solution or smart home solutions for homes, offices, hotels, schools, etc. The Green Living Solution of Qees enables energy management and smart home solutions. The M2M architecture is based on Qees Open Software

Platform, a cloud based solution. The smart gate offers connection to M2M wireless technologys like ZWave, Zigbee, WiFi, 3G and Bluetooth. M2M communication between Qees Management Software, Utilities and Broadband Providers in the cloud and the qees smart gate is through XMPP protocol.

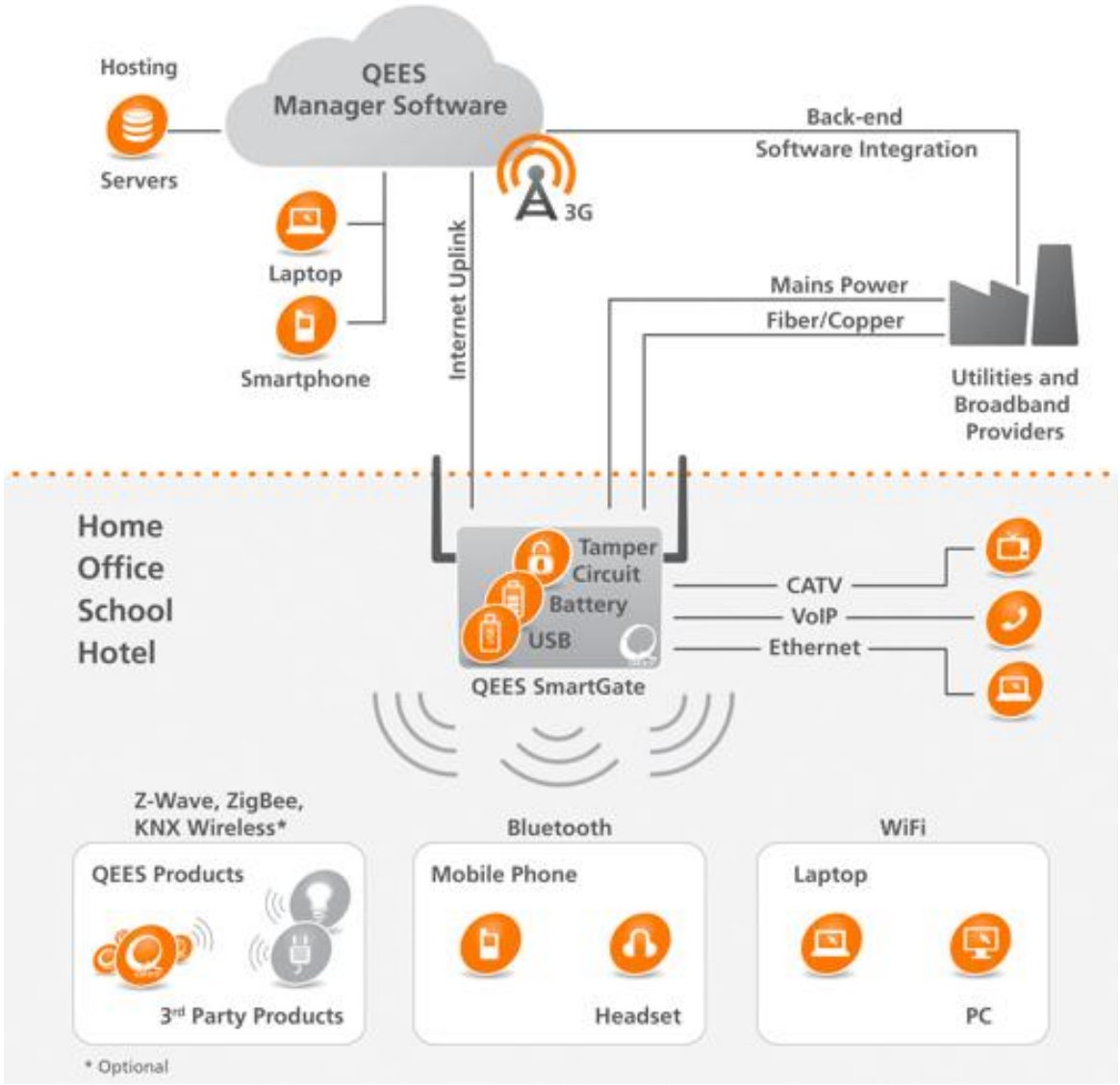

Figure 11: Qees Cloud Architecture.

3.1.1.5 Openpeak

Openpeak [70] provides solutions for home energy control, media and entertainment as home security and monitoring. The supporting sensor technology's are Zigbee, ERT, UPnP. Tablets, iPads, settop boxes communicate over ip with applications deployed into the cloud.

3.1.1.6 Prosyst

Prosyst [71] provides solutions for smart home, mobile, automotive telematics and M2M in general with an open standard technology OSGi which is java based. The supporting M2M sensor technology's are ZWave, Zigbee, UPnP, KNX, X10, WebCams and UPnP. Software API's to the applications is JSON for backend or OSGi services for applications on the M2M gateway.

3.1.1.7 Fifthplay

Fifthplay [75] provides solutions for home, safety control and for home energy management for residential homes. Fifthplay uses Zigbee as M2M wireless technology.

3.1.1.8 Shaspa

Shaspa(IBM) [72] provides solutions for smart home solutions converting entertainment, domotica, energy management, assisted living to one solution. The Shaspa Bridge is the basic building block that enables smart technology, connecting and controlling building automation systems, household appliances and mobile environments.Depending on the environment deployed the Shaspa Bridge can operate in two modes: as Standalon; or as Building Service Gateway (or 'Cloudlet' if used in combination with the Service Framework). In Gateway mode the Shaspa Bridge acts as local extension to the Cloud based Service Delivery Framework. The Shaspa Bridge communicates with the protocols (Enocean, KNX, Modbus, CANOpen, Zigbee, ZWave, DALI, MBus, SNMP, MPBus..) .

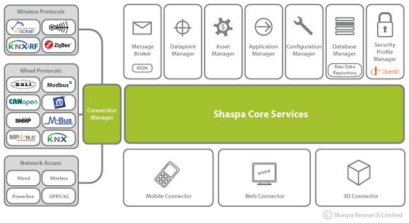

Figure 12: Shaspa Bridge.

The Shaspa Bridge connects to the Shaspa Service Delivery Framework. The Framework collects, stores, analyses data and automates an workflow based on information from the Shaspa Bridge and other sources.

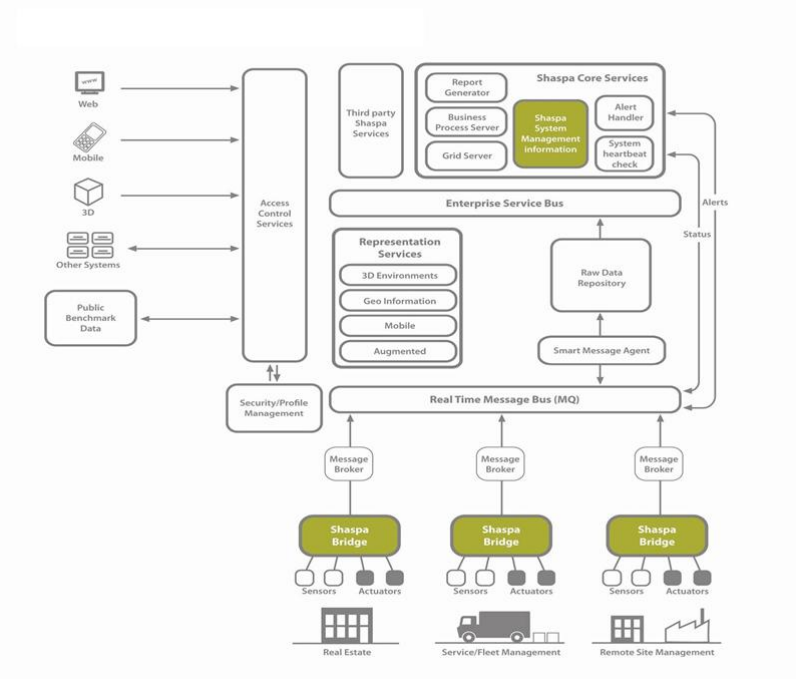

Figure 13: Shaspa Cloud Architecture.

3.1.2 Medical Applications

3.1.2.1 Alcatel-Lucent TeleHealth Manager

Alcatel-Lucent TeleHealth Manager [73] offers health care providers a combination of health monitoring, medical alerts and secure access to collected data. The TeleHealth Manager communicates with the sensor device using Bluetooth.

3.1.2.2 Google Health

Google [74] health provides an application to organize, track, monitor, and act on your health information. The supporting M2M sensor technology is unspecified. Software API to the application is Rest interface(WADL).

3.1.2.3 Fifthplay Health Monitor

Fifthplay [75] health monitor provides a completely integrated internet-enabled remote personal health diagnostics solution. The supporting M2M sensor technology is unspecified as the API to the applications.

3.1.3 Logistics Applications

The area of logistics and production applications is characterized by a diverse field of systems mainly built out of individual special solutions. The amount of standard end-user ready

products using M2M communication is very limited. But there are several standards used for the M2M communication in industrial systems.
OPC (OLE for process control) and OPC UA (Unified Architecture) are two specifications from the OPC Foundation [76]. OPC is a standardized interface for the data exchange between different systems and vendors in the automation industry. The first specification of OPC was released in 1996, the first version of OPC Unified Architecture was released ten years later. The difference between OPC and OPC UA is among others in the communication model. OPC based on COM/DCOM ((Distributed) Component Object Model) communication, coming from the Microsoft world, while OPC UA is based today on a cross-platform service-oriented architecture based on the one hand on an efficient binary protocol and on the other hand on web services.
Fieldbus protocols (IEC 61158) are a collection of network protocols allowing (real-time and) distributed communications with all kind of devices, like sensors, actuators or embedded PCs. Communication via fieldbuses can be found in nearly any time and safety critical industrial application, like manufacturing and production systems up to automotive in-car communication. Some examples for common fieldbus standards are Interbus, Industrial Ethernet (EtherCat, ProfiNet), ASI-Bus and CAN-Bus.

3.1.4 Defense Applications

In general M2M applications in the defense industry are centered around monitoring and controlling assets in remote locations, emergency management, search and recovery and disaster recovery situations. The players in the particular area do not offer specific products, but provide generic M2M application development platforms, which can be tailored to the target application requirements. The functionality offered by these platforms is largely based on wireless sensor networks for monitoring the application-related conditions.

3.1.4.1 SeeControl

SeeControl offers a generic platform for the development of M2M applications, along with customization services of the platform to suit the target application area. Amongst others, the SeeControl M2M platform has also been used in the Defense industry [77]. Details on the exact M2M techonology and APIs utilized are not supplied.

3.1.4.2 Sedona Strategies

Sedona Strategies [78] provides multi-sensor network solutions based on proprietary fusion algorithms in order to enhance M2M based sensor networks. The company offers M2M development services and consulting services, in order to provide optimized wireless sensor networks. The defense industry is addressed by the company‘s services; however no further information on neither the fusion algorithms nor the M2M technologies utilized by the company are provided on the site.

3.1.5 Environmental Applications

As is the case for the defense industry, environmental-related M2M products are generic M2M development platforms, which can be customized as needed for the target application. In general, M2M applications in this area incorporate monitoring environmental conditions via wireless sensor networks and reacting to user-specified conditions according to the user-defined actions.

3.1.5.1 MeshVista

The MeshVista M2M platform is developed by MeshSystems in order to provide M2M solutions in various idustries. Amongst these, environmental applications of the MeshVista platform include [79] pollution/contamination detection, soil, air and water quality monitoring, climate change measurement (greenhouse gas emissions and carbon sequestration), management and control of hazardous chemicals, detection and eradication of pests and vermin, as well as cold chain and other environmental conditions monitoring. Details on the specifics of M2M APIs used in the platform are not specified.

3.1.6 Analysis

M2M communications have infiltrated a large number of application areas and have enabled the provision of new services. The main target applications that use M2M communications are home/building automation, eHealth, logistics, defence and environmental conditions monitoring. The overview of currently available products that integrate M2M communications has lead to the following conclusions

- the vast majority uses web-based technologies to realize communication between machines,
- the IP protocol is the dominant protocol for the network layer,
- an interworking gateway is a crucial part of the overall architecture. As there is a variety of technologies used in local domains, the gateway is the point for implementing the interworking function of the domain technologies with the external public data network that utilizes IP and web-services-based communication.

3.2 M2M Prototypes

A prototype is considered in this section as a working model used to assess the risk associated with deployment and uses of new technologies and identify potential development paths. The prototypes listed below are developed by both commercial companies and educational organisations, as proofs of concept. Depending on the entity that developed it, different kind of information will be provided regarding the actual configuration/features/capabilities of the entities constituting the prototype.

3.2.1 InterDigital M2M prototype

InterDigital is prototyping [80] an M2M system based on the ETSI M2M standard. The produced architecture consists of M2M sensor nodes, managed by M2M coordinators. M2M coordinators are themselves connected to an M2M gateway. This latter is connected to the (simulated) Internet through a 3G femtocell. Internet connectivity allows a remote M2M server and emulated M2M core network to interact with the M2M gateway.
The scenarios that are demonstrated by this prototype regard security and home automation. In the home automation demo, there are temperature measurement sensors as well as actuators that allow to setup the target temperature of a thermostat.
The M2M sensor nodes are TinyOS platforms, that use an IP stack with 6LoWPAN and 802.15.4 radio technology to communicate. The CoAP protocol is used by the M2M gateway for retrieval of a sensed data at a sensor (pull model).

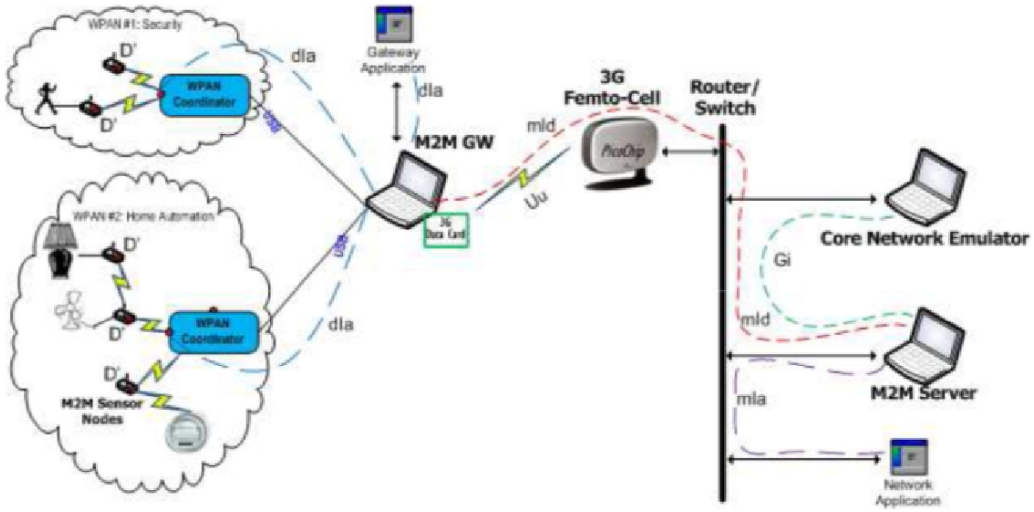

Figure 14: InterDigital prototype architecture (source: InterDigital)

The prototype was still incomplete at the time of information retrieval. The next planned steps include: additional use cases (home automation, home security, ETSI compliant registration of a device application), integrity validation, over-the-air updates, additional CoAP features (POST, PUT, DELETE), caching/proxying and support of low-duty cycle sleeping nodes.

3.2.2 Orange Labs Digital Home

Within the Digital Home program [81], Orange Labs is exploring in a "connected home" project how machine-to-machine communications can be used in an energy saving scenario, to interconnect energy-aware devices with an M2M gateway. This M2M gateway is the ADSL box installed at customer‘s home in order to also provide him Internet connectivity.
Through a web interface, used for measurement and control, the user is able to check sensors‘ status and trigger actuators‘ behavior. Smart context-aware chaining mechanisms are

demonstrated. They consist in intelligent sequences that perform command operations that optimally correspond to a given context, so as to make the control process much simpler to the user.
Another prototype developed at Orange Labs is "Lockster", a SIM-card enabled object tracking device that can be used, for example, as a means to prevent theft of a vehicle: the device is able to detect movement (corresponding to various pre-programmed patterns) and to send an SMS warning to the owner of the tracked device, including its current location.

3.2.3 EDF R&D Smart Grid Experimental Platform

The EDF R&D Smart Grid experimental platform is an M2M communications platform that consists of a multi-technology vehicular router connecting over Wi-Fi to nearby devices. The vehicular router is equipped with various egress interfaces using multiple uplink technologies, which it uses to connect to a remote management system. The uplink technologies that the mobile router is able to use are the following: PLC, GPRS, 3G, Wi-Fi, Digital Mobile Radio and Wimax. Mobility management of the mobile router, which also allows to hide mobility from the end devices performing Wi-Fi connection to the mobile router‘s Wi-Fi ingress interface, is ensured by the use of the NEMO protocol.
Thanks to these technologies, the mobile vehicle can provide to neighboring workers an optimal connectivity while hiding its own mobility to them. It could be used for automated meter reading, especially with respect to "isolated" meters [82].

3.2.4 Wuxi (China) Institute of things

As a key project in the field of the Internet of things, China is building around the city of Wuxi a large (10 sq km) area for demonstrating the benefits of wide-scale objects interconnection. Within eight months from August 2009 to the present multiple structures have been established one after another: Wuxi Research Institute of Internet of Things Industry, State Sensor Information Center, China Research and Development Center for Internet of Things, China Pilot Zone for Innovations and R&D in Sensor Network Industry, China R&D Center for Innovations in Sensor Network, and China Alliance of Internet of Things Technology and Industry. Meanwhile, airport intrusion protection system based on Internet of Things application has been put into operation [83].
The Internet of things architecture deployed in Wuxi bases on a wide-scale deployment of sensors (surrounding vibration, sound, magnetic, microwave) and IP cameras, able to communicate with each other.

3.2.5 Trangram

The Japanese Trangram system [84] provides a simple way of controlling electrical equipment in a home environment. A prototype shows how an iPhone can be used to quickly control a variety of electrical devices in a "smart home". The emphasis is specially put on the planning phase: interconnection of various electrical devices is achieved through a user friendly web-based configuration interface. In a nutshell, each device is represented as a

box on the web applications. Boxes can then be combined using logical interrupters and cause/consequence relationships, so as to enable complex system behaviors (turn on this device and this one if this other device or this one are turned off, etc.) with a limited (and funny) interaction with the user.

3.2.6 HP CeNSE

The HP CeNSE (Central Nervous System for the Earth) [86] project aims at gathering information about various parameters related to global health of the Earth through sensors and have this data processed by cloud computing technology. In this project, it is especially relevant to consider a large amount of sensors, in order to obtain the most precise data. Sensors are also able to capture a large variety of information about the environment: vibration, tilt, air flow, light, biological, rotation, navigation, temperature, chemical, humidity, sound, pressure.
Technology developed for CeNSE project is not restricted to environmental use. For example, a sensitive accelerometer sensor coupled to an advanced signal processing software has been demonstrated as a system for monitoring essential health parameters for elderly people. Breadth and heartbeat rhythms analysis can also be achieved, which can lead to disease/incident diagnosis.
CeNSE will also be used in a more profit-oriented scenario: in partnership with Shell, it is planned to deploy HP CeNSE sensors in prospected areas in order to allow for underground oil detection: high sensitivity of CeNSE sensors is expected to be able to "listen for sound waves bounced through miles of sub-strata and hopefully pockets of oil".

3.2.7 Critical Software EMMON

Realized within the Artemis-Emmon EU project, the EMMON (EMbedded MONitoring) prototype is claimed [85] to be the largest wireless sensor network in Europe, with no less than 303 tiny sensors in a single room, gathering detailed, real time information on temperature, humidity and ambient light. It demonstrates the integration of a number of core components, from embedded wireless sensors to the control station. The prototype itself is named DEMMON1 and has been demonstrated in December 2010. It exhibited the ability to quickly report perceived changes in the environment (light and temperature), even when a large number (100+) of sensors were affected simultaneously by the change.
DEMMON 1 bases on a hierarchical architecture in which sensors are grouped within clusters under a cluster head which, in turn is grouped under a serving gateway. A Command and Control (C&C) server stands on top of the overall sensor networks, and is the interface offered to various command and control clients. Zigbee connectivity is used to interconnect sensor nodes, cluster heads and gateways; in the sensor network, only gateway nodes are IP-capable. An advanced geographical networking layer has been developped on top of all network layers of EMMON nodes (Zigbee for sensors and cluster heads, IP for gateways). This makes it possible to issue geographical requests, e.g. asking a specific portion of the network about the status of some physical parameters.
The networking stack used in the DEMMON1 prototype is depicted below.

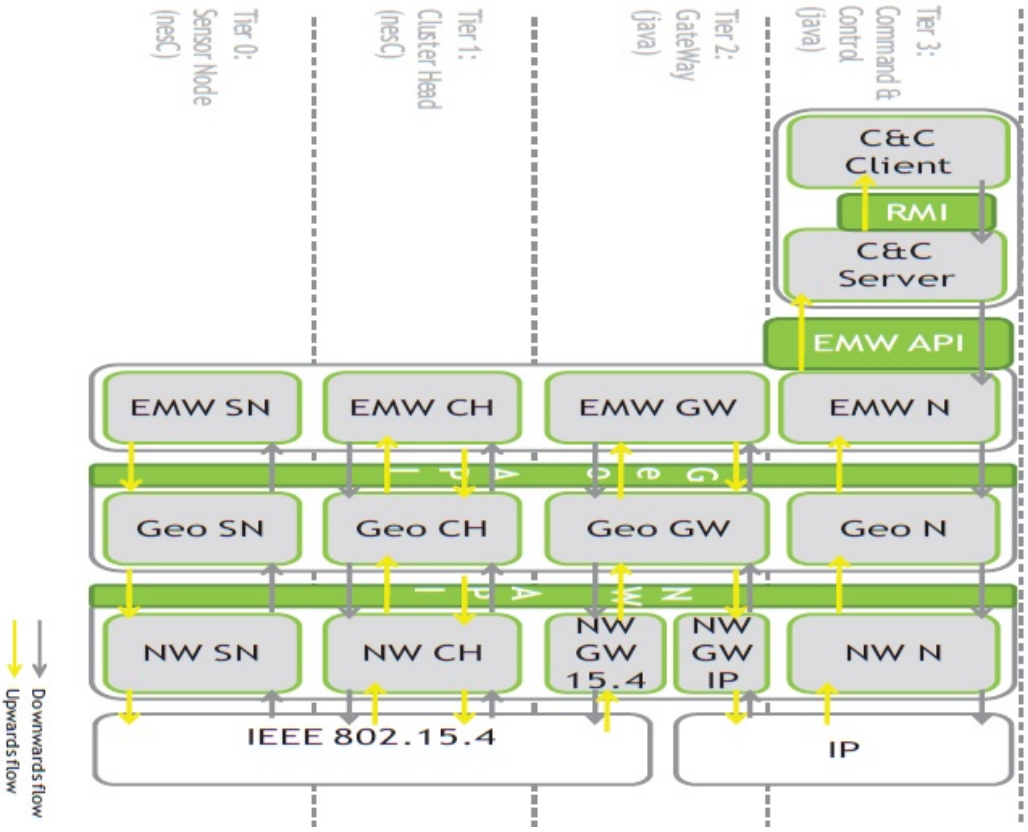

Figure 15: EMMON prototype architecture (source: Critical Software)

3.2.8 Home Sensor Gateway prototype (Alcatel-Lucent)

Current state-of-the-art solutions are vertically oriented: sensor application vendors are in complete control from application down to the sensor hardware, each providing their own ecosystem of software, sensor devices and gateways. The Alcatel-Lucent prototype of HSG(Home Sensor Gateway) promotes a horizontal model that offers an open service platform, such that 3rd party application providers can deploy and run their services with the necessary service assurance. The HSG performs M2M functions such as automatic activation, control commands, accessing configuration settings, data collection.
In order to realize this model, a HSG prototype has been developed. The HSG is a software framework that runs on the service gateway and that offers sensor device and network abstraction in a scalable and extendible telecom compliant way:

1. The framework models every sensor in an easy parameter/value data model that discriminates data plane parameters and management plane parameters.

2. The sensor abstraction layer hides technology communication details from the application and management plane.

3. To access the sensor parameters, an asynchronous REST-full transaction protocol is offered that is handled by a priority queuing mechanism.

4. The framework is also based on a device driver SDK that uses a template based development approach.

The HSG consists of 3 layers:

1. Management Layer. The Management layer provides management of the sensor devices (e.g. auto-configuration, monitoring, etc) and configures and controls the HSG.

2. Sensor Abstraction Layer. The Sensor Abstraction Layer provides a sensor device abstraction API to the applications and a device driver API to the device vendors in order to hide the sensor technology details from the applications.

3. Protocol Layer. The Protocol Layer provides fair and reliable low layer access to sensor technology and triggers the management layer for new device and technology detection. Integerated protocols are RFID, Bluetooth, Zigbee, (ZWave), sunspots and under development COAP.

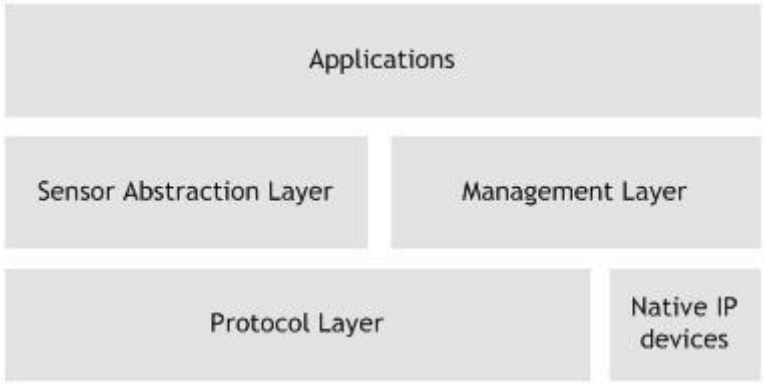

Figure 16: Home Sensor Gateway layered model

3.2.9 AIM Gateway prototype

The main concept of the AIM project [87] is to offer a harmonised technology for managing in real time the energy consumption of appliances at home, interworking this information to communication devices over the home network and virtualising it with the final aim of making it available to users through home communication networks in the form of standalone or network operator services.
The key component of the AIM architecture is the AIM Gateway. The AIM gateway is a communication component that has the ability to host user services, while serving communication with user terminals over the indoor and outdoor networks and implementing control of the household appliances by employing special communication interfaces, tailored to the communication interfaces of the household appliances - i.e. KNX, Zigbee. Furthermore it has the role of bridging the functionality of the home network with the user applications residing in the wide area network (outdoor networks) and providing harmonisation of communications between the users and the involved network components over the IP protocol. Apart from providing bridging logic, the AIM gateway implements the basic substrate for

- the deployment of user services,
- the implementation of interoperability between network components, such as sensors, household appliances, specialised network components, etc, and
- the accommodation of service creation and execution environments.

For scalability, upgradeability and openness reasons, the gateway is implemented as an open, standardised architecture, based on the open services execution framework of OSGi. Inside AIM, appliances provide one or more machine accessible interfaces. Utilising these interfaces, the status of a device can be examined and its operation can be controlled. However, different vendors provide their own products addressing the need for a specific functionality. For example, there is a multitude of vendors that produce washing machines. From the point of view of AIM, what makes these products belong in the same category is the existence of a common set of functions or operations that each and every one performs.

For instance, all washing machines need to be “programmed”, started and monitored for current status/ wash phase.
In order to provide a unified view of the household appliance operations, the AIM Gateway provides an abstraction layer (the M2M) interface which can readily use and be adapted to standardized sets of functions for the supported device types. The role of the M2M interface is to provide a unified interface from where devices can be controlled, regardless of the brand or the“protocol” they use. The AIM M2M can be viewed as the abstraction layer between the real-world physical devices’ exposed interfaces and the software world that integrates and enhances the device functionality based on those interfaces. The necessity of this abstraction layer is imposed by the fact that, most of the time, and for various reasons, vendors do not expose the same machine interfaces format and technology, even though the functionality they provide is almost the same. So, an intermediate layer is needed between the “user” and the device, which will implement a more abstract interface, focusing on the functionality, rather than the device specifics. The M2M interface implemented in the AIM Gateway uses standard UPnP mechanisms for communicating with household appliances.

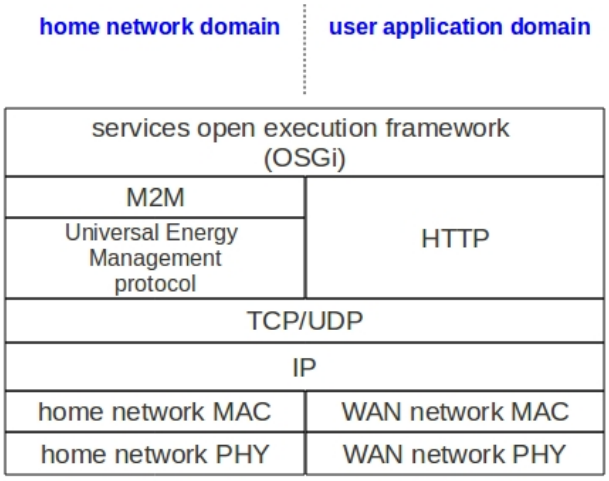

Figure 17: The AIM Gateway protocol stack

The management of the services of the gateway by the home user is realized via Web service calls, using HTTP. The user interface has the form of Web pages. On the other hand, for the communication of the gateway with the connected household appliances, the M2M API is utilized, which is employed by the gateway services for the implementation of appliance program control and status monitoring. Additionally, a new protocol called Universal Energy Management protocol was designed, to provide a unified way of implementing control and status commands exchange between the AIM Gateway and the energy controlled appliances. The overall AIM Gateway protocol stack is summarized in Figure 17.

3.2.10 Summary

M2M prototypes assessed in this section cover multiple domains in home and industry areas, and usually involve hundreds of nodes. These prototypes can roughly be split into two categories depending on whether they target the validation of a single functionality, as proof of concept, or aim at defining global frameworks. The first category focuses on scenario-specific elements. While very various and innovative physical sensors can be found in this group of solutions, these generally rely on a single protocol family and consider interoperability, completeness and to some extent security as secondary aspects. The second category (Alcatel-Lucent and AIM solutions) rely on the same well-known radio technologies and protocols, but also design overlays in order to hide the diversity of the multiple supported protocols to the user and applications. In order to define an efficient IoT framework, the two aspects of universality and closeness to physical world are to be taken into account.

4 IoT-A M2M Communication Requirements

This section discusses the the initial M2M API requirements that will be used for the design of the protocol stack in the IoT-A project.
After a presentation of the applied methodology in section 4.1, the section 4.2 lists the requirements used as basis for the chapter. In the sections 4.3 we discuss the requirements for a M2M API that are applicable to the general IoT Architecture and present examples for the application specific meaning of those requirements. The section 4.4 concludes this chapter with an overview of requirements that might be addressed by the communication protocol.

4.1 Requirements Definition Methodology

The results from the review of the state of the art of M2M communication in section 2 and the evaluation of existing commercial M2M applications and M2M prototypes in section 3 are considered as precondition and focus when deriving the requirements. These sections provide and extend a M2M communication viepoint on content that is contained in a more generic pespective in deliverable D1.1 [92].
While the recent start of M2M standardization in many countries indicates the need for unified M2M communication protocols, only very limited results are available yet. As one of the available results, the requirements from ETSI [90] are included in the analysis. A general trend in protocols and applications is the integration of the Internet Protocol (IP) in the communication layer. Nevertheless, existing protocols (section 2.3) are mainly application specific and the platforms (section 2.4) aim at vertical markets. The need for a more generalized approach is confirmed by the weight given to interworking gateways as seen in the analysis of the commercial M2M products (section 3.1). The range of functionality needed for a generalized M2M communication approach can be seen from the analysis of M2M prototypes (section 3.2)- those are divided by focus on universality versus technology specific grade of detail. A future solution needs to by universally usable while being able to reflect very specific applications.

The initial M2M API requirements are derived from requirements of the overall IoT-A project and requirements to M2M services taken from the ETSI. As shown in Figure 18 the general IoT-A and ETSI requirements are filtered for communication requirements and then analyzed to derive a set of requirements applicable to a M2M API, while the results from the state of the art are used as side condition.

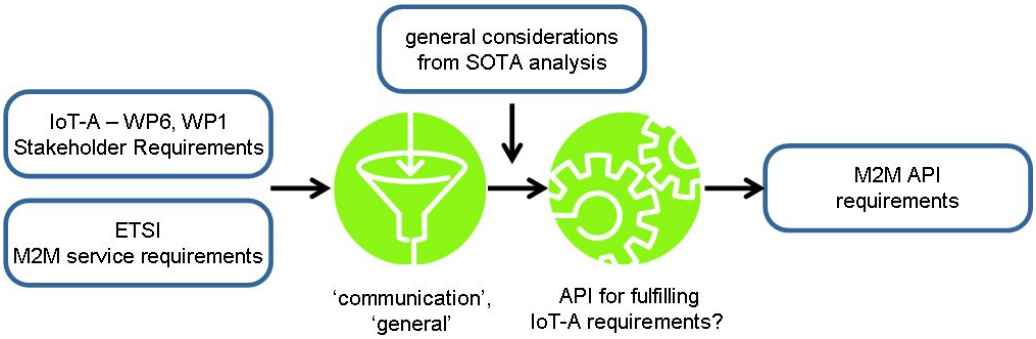

Figure 18: Requirements Definition Methodology

4.2 List of Communication Requirements from Stakeholders and the ETSI

WP1 and WP6 analyzed the requirements provided by the stakeholders in the WP6 Requirements List [91] and derived a set of unified requirements [94]. These unified requirements are classified by the perspectives and viewpoints as defined in the Initial Reference Architecture [93]. The following subsection 4.2.1 lists the requirements related to the 'communication' vision, since those can be used to derive M2M API requirements.
A further source for requirements is the ETSI M2M service requirements documents [90]. The general requirements of this document are listed in subsection 4.2.2.

4.2.1 Communication Requirements from Stakeholders in Unified Form

The following Table 1 presents the stakeholders requirements related to the viewpoint 'communication'.

ID	Unified Requirement	Priority
UNI.15	Devices shall be remotely controlled and configured	Low
UNI.17	The system shall support event-based, periodic, and/or autonomous communication	High
UNI.18	The system shall support data processing (filtering, aggregation/fusion, ...) on different IoT-system levels (for instance EoI level)	High

UNI.37	The system shall provide location information associated to the EoI	High
UNI.20	The system shall support the real-time monitoring of the radio activity of devices and gateways	Medium
UNI.21	The system shall support the management of the radio transmitting devices in real-time	Medium
UNI.26	The system shall support time critical message handling and delivery	High
UNI.27	The system shall support priorization of services	High
UNI.28	The system shall support some mechanism of messages priorization	High
UNI.22	The system shall support secure communications through secure messaging tool	High
UNI.23	The system shall provide access to external information sources, e.g. health databases	High
UNI.24	The system shall provide reliable communication, e.g. for health information	High
UNI.29	The system shall provide a support for routing of data based on content	Medium
UNI.89	The system shall support secure time synchronization	Medium

Table 1: IoT-A communication viewpoint system requirements from WP6 and WP1

4.2.2 Communication Requirements from the ETSI

The following Table 2 lists the general requirements given in [90]. The requirements are identified by 'ET-' and the section numbering. If needed, additional explanation is given in parenthesis after the requirement.

ID	ETSI Requirement
ET-4.1	M2M Application communication principles (abstract underlying network)
ET-4.2	Message Delivery for sleeping devices
ET-4.3	Delivery modes: anycast, unicast, multicast and broadcast
ET-4.4	Message transmission scheduling (should be supported)
ET-4.5	Message communication path selection
ET-4.6	Communication with devices behind a M2M gateway
ET-4.7	Communication failure notification
ET-4.8	Scalability

ET-4.9	Abstraction of technologies heterogeneity
ET-4.10	M2M Service Capabilities discovery and registration
ET-4.11	M2M Trusted Application
ET-4.12	Mobility (if supported by underlying network)
ET-4.13	Communications integrity
ET-4.14	Device/Gateway integrity check
ET-4.15	Continuous connectivity
ET-4.16	Confirm (of messages)
ET-4.17	Priority (of the services and communications services)
ET-4.18	Logging
ET-4.19	Anonymity
ET-4.20	Time Stamp
ET-4.21	Device/Gateway failure robustness
ET-4.22	Radio transmission activity indication and control
ET-4.23	Operator telco capabilities exposure
ET-4.24	Location reporting support
ET-4.25	Support of multiple M2M Applications

Table 2: ETSI general M2M service requirements

4.3 IoT-A M2M API Requirements

This subsection presents the results of the discussions in WP3 of the M2M API requirements for M2M communication in the IoT. The approach as defined in section 4.1 was used.

4.3.1 Nonfunctional: Adaption to Differences in Applications and Devices

Given the heterogeneous nature of the applications and device categories targeted in the IoT, the M2M API concept needs to be adaptable to the capabilities and requirements of the specific use case. It is obvious, that restricted devices will implement less functionality in the API then a more capable device (e.g. smartphone) will do. A basic common API that can be extended in a well defined way will allow to fulfill more complicated requirements, while offering the simplicity needed in other cases. This API requirement is also addressed by the ETSI general requirements ET-4.8 (scalability) and ET4.1, ET-4.9 (abstraction of technologies and communication heterogeneity).

4.3.2 Device Control

The M2M API should provide an interface for controlling the device. This includes the configuration of the device as well as support for remotely activating, deactivating or updating the device.

The requirement UNI.15 reflects the need for a device control API directly. The stakeholder application for this example is the reconfiguration of patient monitoring equipment. In other cases software on IoT devices might be updated to fix bugs or to add new features, while the device will remain at its installation location. This will be a use case affecting some millions of deployed meters when upgrading the systems to a new standard, as it might happen during the transition from ZigBee Smart Energy Profile 1.0 to Smart Energy Profile 2.

4.3.3 Server and Client Communication Models

The API should provide support for operating a device as sever and client. By changing the server / client role, event-based, autonomous (unsolicited) and periodic communication, as well as request-response communication behaviors are possible. While this requirement - derived from UNI.17 - seems to be quite natural in the Internet, the strict polling pattern of many automation devices hinders integration and efficient resource usage.

4.3.4 Device Status Monitoring

The status (battery, memory, radio usage) of a device should be accessible. Usage of this kind of information may be found when selecting the communication path (ET-4.5) or needing information or control on radio usage in sensitive health environments (UNI.20, UNI.21).

4.3.5 Communication Failure Notification

The M2M API should provide an interface to inform about failures of the communication module (c.f ET-4.7). This information provides a handle for mechanisms ensuring communication connectivity (ET-4.15), device failure robustness (ET-4.21) and reliable communication (UNI24).

4.3.6 Information on the Device

The API should provide access to the application specific information on the device. This might be results/data captured on different levels (UNI.18) or location information (UNI.37, ET-4.24). The mechanism for data processing or aggregation as well as determination of the location information is out of scope of the API.

4.3.7 Device Capabilities

The API should provide information of the capabilities of a device. In terms of information offered by a device, this feature allows to access devices based on content (UNI.29) or to select the best communication path (ET-4.5). Further, information about a devices communication capabilities will help with prioritization requirements (UNI.26, UNI.27, Uni.28).

4.3.8 Communication Properties - Security, QoS, Confirmation

An API accessing the communication properties allows to configure the security (confidentiality, integrity) or QoS of the communication. This API feature is important for fulfilling requirements for prioritization (UNI.26, UNI.27, Uni.28) or confidentially, integrity and reliability (UNI.22, UNI.24).

4.4 IoT-A M2M Protocol Stack Requirements

The M2M API uses the underlying protocol stack to realize communication between end devices. As M2M APIs expose the device capabilities and will also support remote device management and configuration, it is evident that security becomes a dominant issue. Furthermore, the M2M protocol stack must provide the mechanisms necessary for discovering devices that join the network, along with their capabilities. In order for the services offered by a device to be accessible, the device must primarily be identified. Additionally, the services it offers must be published so that they are accessible by other devices. Another important aspect of the protocol stack, is its capability to differentiate message flows and prioritize datagrams. Thus, it will be possible to have a M2M system that will support different categories and types of services, enabling service providers to enforce pricing and charging policies.
Due to the fact that packet loss is possible in packet-based networks - i.e. due to the failure of network nodes, traffic congestion, etc - the M2M protocol stack must support reliable message delivery mechanisms, to be used by applications that require it. An example of this kind of application would be a control application needing verification for the delivery of the control message it has sent to a remote machine, using the remote machine's M2M API. Finally, as there may be an application running on a machine that needs to communicate with multiple applications running in remote machines, support for multicasting is required by the underlying network. Such a capability, would allow IoT endpoints to join/leave multicast groups in which messages of interest to these endpoints would be transmitted.
The aforementioned provide a starting point for the gathering of requirements for the IoT-A protocol stack, from the perspective of M2M communications. Specifically, the IoT-A protocol stack must provide the mechanisms necessary to support

- device discovery and device capabilities discovery,
- reliable message transport and delivery,
- data integrity,
- data confidentiality,
- Quality of Service,
- multicasting/broadcasting.

5 Conclusions

With M2M communications being an integral part of the Internet of Things, it is necessary to analyze the overall requirements imposed on M2M systems. Towards this end, the present document provided an overview of the state of the art in M2M communications, both in the research, as well as in the commercial domain. Additionally, commercial and open-source platforms used to develop M2M applications were presented, providing technical details related to the adopted technologies and the relative architecture.
Such an analysis was necessary, in order to identify current trends and gaps in M2M communications and use them as the basis for pointing out requirements for the M2M API to be used in the IoT-A project. Furthermore, M2M communications impose requirements on the protocol stack to be used for realizing the actual communication between machines. Requirements that need to be addressed when designing the narrow-waist IoT protocol stack.
On top of the requirements extracted from the state-of-the-art on M2M communications, the unified requirements derived by WP1 and WP6, along with M2M service requirements published by the ETSI were elaborated. The overall result of this process was a solid, well-defined set of the functionality the IoT-A M2M API must expose to support M2M communication. The elaborated functionality set will be the cornerstone for realizing the public interface each device must export to properly communicate with other devices in the Internet of Things.
Having identified the basic set of M2M API requirements, future work will be delving into the specifics of the implementation of the API, translating the set into specific method calls. An important part of this process will also be to define the format used to represent the information exchanged between devices. Additionally, the actual technologies to be used for realizing M2M communication will be decided; however, the overall approach will remain technology-indepent, thus allowing the implementation of the proposed functionality via alternative technologies. Last but not least, it is crucial to integrate the M2M functionality with the protocol stack of the Internet protocols to ensure maximum compatibility of M2M with underlying networking technologies.

6 Appendix

6.1 WPAN standardization activities

6.1.1 Institute of Electrical and Electronic Engineers

The Institute of Electrical and Electronic Engineers addresses M2M communications via its 802.15 Wireless Personal Area Network (WPAN) Task Group 4 [95]. The outcome of this particular Task Group is the specification of the Physical and Media Access Control layer for low-rate, low power consumption WPANs. IEEE 802.15.4 is operating in an unlicensed international frequency band. Its target application areas include but are not limited to sensors, interactive toys, smart badges, remote controls and home automation. The IEEE 802.15.4 is used as the basis for various specifications, such as the Zigbee, and Wireless Highway Addressable Remote Transducer [96], whereas it can be also used with 6LoWPAN.

6.1.2 Internet Engineering Task Force

M2M communication standardization activities in IETF are centered around embedded devices with limited resources, such as sensors. The main representatives of M2M communication standardization efforts within IETF are the Routing Over Low power and Lossy networks (ROLL) working group [97], as well as the IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) protocol [98]. The ROLL working group is focused on routing issues concerning Low power and Lossy Networks (LLNs), such as

- specification of routing metrics used in path calculation,
- providing an architectural framework for routing and path selection at Layer 3,
- producing a routing security framework for routing in low power and lossy networks.

As there is a wide scope of application areas for LLNs, i.e. industrial monitoring, urban sensor networks, asset tracking, building automation, connected homes, environmental monitoring, healthcare, etc, the working group focuses on routing solutions for the subset of these that routing requirements have been specified, meanly industrial, connected home, building and urban sensor networks.The final outcome of the ROLL working group will be the definition of LLNs routing requirements, which will result in either the specification of a new protocol or the extension of an existing routing protocol. Moreover, if the LLNs requirements are not met with a single protocol, the working group may specify and/or extend more than one protocols.
The 6LoWPAN protocol defines the adaptation mechanisms necessary that allow IPv6 packets to be sent/received over IEEE 802.15.4 networks. Practically, the scope of the 6LoWPAN protocol is to enable low power, WPANs have a network layer based on IPv6. Specifically, the 6LoWPAN protocol addresses issues such as

- adapting packet sizes between IP and 802.15.4 networks,
- address mapping between the two networks,

- defining an adaptation layer, allowing the transmission of IPv6 datagrams over IEEE 802.15.4 networks,
- routing of packets between the IPv6 domain and the PAN domain,
- device and service discovery.

References

[1] http://www.etsi.org/Website/Technologies/M2M.aspx

[2] ETSI TR 102 732 V0.3.1, Machine to Machine Communications (M2M); Use cases of M2M applications for eHealth

[3] ETSI TR 102 857 V0.3.0, Machine to Machine Communications (M2M); Use cases of M2M applications for Connected Consumer

[4] ETSI TR 102 897 V0.1.1, Machine to Machine Communications (M2M); Use cases of M2M applications for City Automation

[5] ETSI TR 102 898 V0.4.0, Machine to Machine Communications (M2M); Use cases of Automotive Applications in M2M capable networks

[6] ETSI TR 102 691 V1.1.1, Machine-to-Machine communications (M2M); Smart Metering Use Cases

[7] http://www.3gpp.org/-SA-

[8] 3GPP TS 22.368 V11.0.1, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Service requirements for Machine-Type Communications (MTC); Stage 1, (Release 11)

[9] 3GPP TR 22.868 V8.0.0 (2007-03), 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on Facilitating Machine to Machine Communication in 3GPP Systems; (Release 8)

[10] 3GPP TR 23.888 V1.0.0 (2010-07), 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; System Improvements for Machine-Type Communications; (Release 10)

[11] http://www.tiaonline.org/standards/procedures/manuals/scope.cfm#TR50

[12] http://ccsa.org.cn/english/tc.php?tcid=tc10

[13] http://www.gisfi.org/index.php

[14] Ubiquitous Sensor Networks (USN), ITU-T Technology Watch Briefing Report Series, No. 4 (February 2008), available at http://www.itu.int/dms_pub/itu-t/oth/23/01/T23010000040001PDFE.pdf

[15] http://www.openmobilealliance.org/

[16] http://docbox.etsi.org/Workshop/2010/201010_M2MWORKSHOP/06_M2MGlobalCollaboration/L

[17] http://www.gsc.etsi.org/

[18] http://www.flexware.at/

[19] http://www.sirena-itea.org/

[20] http://www.soda-itea.org/

[21] https://usenet.erve.vtt.fi/

[22] http://www.ict-exalted.eu/

[23] http://www.ict-aim.eu/

[24] http://www.ist-e-sense.org/

[25] http://openwsn.berkeley.edu/

[26] http://www.openadr.org/

[27] http://sensor.network.com/

[28] http://smote.cs.berkeley.edu:8000/tracenv/wiki

[29] http://www.intel-iris.net/

[30] IrisNet: An Architecture for a World-Wide Sensor Web, Phillip B. Gibbons, Brad Karp, Yan Ke, Suman Nath, Srinivasan Seshan IEEE Pervasive Computing, Volume 2, Number 4 (October-December 2003)

[31] http://local.cs.berkeley.edu/wiki/index.php

[32] An Architecture for Local Energy Generation, Distribution, and Sharing, Mike M. He, Evan M. Reutzel, Xiaofan Jiang, Randy H. Katz, Seth R. Sanders, David E. Culler, Ken Lutz, IEEE Energy2030 Conference Proceedings, Nov 17-18, 2008

[33] http://autoid.mit.edu/CS/

[34] http://www.autoidlabs.org.uk/

[35] http://www.kri.sfc.keio.ac.jp/en/lab/AutoID.html

[36] http://autoidlab.eleceng.adelaide.edu.au/

[37] http://www.item.unisg.ch/en/Chairs/Operations+Mgmt/Research/Auto-ID.aspx

[38] http://www.autoid.or.kr/

[39] http://www.autoid.or.kr/

[40] http://www.modbus.org/docs/PI_MBUS_300.pdf

[41] http://www.modbus.org/specs.php

[42] http://www.scadalink.com/support/cellular-hosted-scada.html

[43] http://www.yokogawa.com/iab/appnotes/iab-app-gprs-en.htm

[44] http://tools.ietf.org/html/draft-ietf-core-coap-04

[45] http://www.upnp.org/resources/upnpresources.zip

[46] http://tools.ietf.org/html/draft-cheshire-nat-pmp-03

[47] http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01

[48] http://www.openhealth.org/RDDL/20040118/rddl-20040118.html

[49] UPnP Device Architecture 1.1 - Document Revision Date 15 October 2008

[50] http://www.bitxml.org/

[51] http://m2mxml.sourceforge.net/

[52] http://mango.serotoninsoftware.com/

[53] http://www.vscp.org/

[54] https://www.alljoyn.org/

[55] http://openscada.org/

[56] http://www.proview.se/

[57] http://pvbrowser.de/pvbrowser/index.php

[58] http://www.osgi.org/Main/HomePage

[59] http://felix.apache.org/site/index.html

[60] http://felix.apache.org/site/index.html

[61] http://www.eclipse.org/equinox/

[62] http://www.inilex.com/Apprize/Apprize.html#

[63] http://www.sensorlogic.com/platform.php

[64] http://www.seecontrol.com/

[65] http://www.ilstechnology.com/devicewise-listing
M2M Applications bibliography goes here

[66] http://www.siemens.com/desigo

[67] http://qees.eu/en

[68] http://www.homeautomationeurope.com/

[69] http://www.energyhub.com/

[70] http://www.openpeak.com/

[71] http://www.prosyst.com/

[72] http://www.shaspa.com/

[73] http://enterprise.alcatel-lucent.com/?solution=Healthcare

[74] http://code.google.com/api/health/

[75] http://www.fifthplay.com/

[76] http://www.opcfoundation.org/

[77] http://www.seecontrol.com/m2m-market-segmentation-software-tools/

[78] http://www.sedonastrategies.com/gov.html

[79] http://mesh-systems.com/wireless-m2m-networks-software-hardware/environmental

[80] InterDigital M2M Prototype Demonstration slideset, available at http://docbox.etsi.org/Workshop/2010/201010_M2MWORKSHOP/09_M2MDEMO_INTERDIGITAL/I September 2010.

[81] Orange Labs Digital Home, http://www.orange-innovation.tv/webtv/digital-home/, accessed March 2011.

[82] http://www.association-aristote.fr/doku.php/public/seminaires/seminaire-2010-10-14, accessed March 2011.

[83] Internet of Things,a Beachhead in the New Wave of Competition among Chinese Cities, http://en.ccidconsulting.com/en/io/mr/mr/si/webinfo/2010/09/1283908065446365.htm, accessed March 2011.

[84] Trangram, http://www.trangram.cc/about.php, accessed March 2011.

[85] http://www.live-pr.com/en/european-consortium-implements-emmon-the-r1048768049.htm, accessed March 2011.

[86] http://www.hpl.hp.com/news/2009/oct-dec/cense.html, accessed March 2011.

[87] http://www.ict-aim.eu/

M2M API Requirements bibliography goes here

[88] Boswarthick, David. Machine 2 Machine: When the machines start talking. MCW 2010. Barcelona, Spain. http://docbox.etsi.org/M2M/Open/Information/M2M_presentation.pdf

[89] Dohler, Mischa, Thomas Watteyne, and Jes?s Alonso-Z?rate. Machine-to-Machine: An Emerging Communication Paradigm. PIMRC 2010, Tutorial. Istambul, Turkey.

[90] ETSI TS 102 689 V1.1.1, Machine to Machine Communications (M2M); M2M service requirements

[91] IoT-A Deliverable D6.1 - Requirements List

[92] IoT-A Deliverable D1.1 - SOTA report on existing integration frameworks/architectures for WSN, RFID and other emerging IoT related technologies.

[93] IoT-A Deliverable D1.2 - Initial architectural reference model for the IoT.

[94] IoT-A Document - Unified Requirements. Work in progress - snapshot from 2011-03-30. http://www.iot-a.eu/internal/workspace/wp6-requirement-validation/t6.1/unified-requirements/Unified%20Requirements%20List_2011_03_30-MB.xlsx/view

[95] http://www.ieee802.org/15/pub/TG4.html

[96] http://www.hartcomm.org/protocol/wihart/wireless_technology.html

[97] https://datatracker.ietf.org/wg/roll/charter/

[98] http://datatracker.ietf.org/wg/6lowpan/charter/